LUMEN

The complete traveller’s handbook

A free journey through Vim.

Polski

LUMEN · Podręcznik strażnika światła

English

LUMEN · The light keeper’s handbook

Español

LUMEN · Manual del guardián de la luz

简体中文

LUMEN · 守光人学习手册

한국어

LUMEN · 빛의 수호자 학습서

日本語

LUMEN · 光の守り手の学習ガイド

24 islands. 72 missions. Six languages.
Use the PDF bookmarks to choose a language or jump to a chapter.

Edition 0.3.0 / vim.promyk.tech

LUMEN

Podręcznik strażnika światła

24 krainy, 72 zadania. Praktyczny przewodnik po ruchach i języku edycji Vima.

24 krainy / 72 zadania / całość bez opłat

Polski / Wydanie 0.3.0

Podręcznik opisuje edukacyjny podzbiór obsługiwany przez grę. Wirtualne pliki pozostają w przeglądarce; gra nie udostępnia systemu plików urządzenia.

Najpierw spróbuj samodzielnie. Rozwiązania są przykładami; inne poprawne sekwencje także mogą spełnić cel.

Spis treści

Zanim wyruszysz 3

Atlas poleceń 5

Przewodnik po krainach 10

01 / Port Latarni . . . 11
02 / Gaj Słów . . . 13
03 / Mosty Wersów . . . 14
04 / Obserwatorium Echa . . . 16
05 / Kuźnia Żaru . . . 17
06 / Biblioteka Luster . . . 19
07 / Pracownia Świtu . . . 21
08 / Wieża Poranka . . . 23
09 / Klify Wiatru . . . 25
10 / Przełęcz Tropicieli . . . 26
11 / Kamienne Kręgi . . . 28
12 / Wieża Horyzontów . . . 30
13 / Warsztat Tkaczy . . . 32
14 / Odlewnia Gramatyki . . . 34
15 / Sanktuarium Słów . . . 36
16 / Skarbiec Ograniczników . . . 38
17 / Kaplica Szeptów . . . 40
18 / Obóz Kartografów . . . 42
19 / Gabinet Ech . . . 44
20 / Komnata Dziewięciu . . . 47
21 / Port Skrybów . . . 50
22 / Mechaniczny Sad . . . 52
23 / Próby Tkaczy . . . 54
24 / Szczyt Początków . . . 56

Dalsza nauka 58

Zanim wyruszysz

Cel podróży

Celem podróży jest świadome operowanie tekstem: od pojedynczego ruchu po obiekty tekstowe, rejestry i makra. Każda kraina wprowadza małą rodzinę poleceń, a zadania wymagają użycia wskazanych narzędzi, nie tylko przypadkowego uzyskania właściwego tekstu.

Ukończenie gry oznacza przećwiczenie jej programu, nie znajomość wszystkich funkcji pełnego edytora Vim. Wracaj do ukończonych zadań i szukaj własnych, czytelnych rozwiązań.

Jak grać

Wybierz język, nadaj imię postaci i zaakceptuj regulamin. Kliknij planszę, aby skierować do niej klawiaturę. W trybie normalnym litery wykonują polecenia; po wejściu do wstawiania stają się tekstem. Esc pozwala wrócić do trybu normalnego. Na urządzeniu dotykowym korzystaj z klawiszy pod planszą i pól wpisywania.

Przed ruchem przeczytaj cel, porównaj zapis z tekstem docelowym i sprawdź wymagane polecenia. Wskazówki są stopniowane. Restart dotyczy bieżącego zadania, a mapa pozwala wracać do już odblokowanych krain. Przycisk Pomoc otwiera objaśnienia i podręcznik w dowolnym momencie. Esc podczas wstawiania wraca do trybu normalnego; w trybie normalnym, gdy żadne polecenie nie czeka na dokończenie, otwiera pauzę. Dostępny jest też przycisk pauzy. Ostatnie trzy zadania można rozgrywać jako opcjonalne próby czasowe. Czas orientacyjny jest dodatkowym wyzwaniem, nie warunkiem ukończenia programu. Najpierw ćwicz poprawność, dopiero potem tempo.

Zapis i prywatność

Imię, zgoda na regulamin i postęp są zapisywane w localStorage tej przeglądarki. Zapis nie jest kontem ani kopią w chmurze. Tryb prywatny, wyczyszczenie danych strony lub ograniczenia pamięci mogą go usunąć; pobieraj kopię zapisu, jeśli chcesz ją zachować lub przenieść na inne urządzenie.

Opcjonalna diagnostyka jest domyślnie wyłączona i ma osobny przełącznik w ustawieniach. Nie jest warunkiem rozpoczęcia gry. Obejmuje ograniczone dane techniczne o błędach, a nie imię, treść ćwiczeń czy pełny zapis podróży. Szczegóły przetwarzania danych opisuje regulamin.

Jak czytać klawisze

Wielkość liter ma znaczenie: w i W, p i P, i i I to różne polecenia. Zapis 3w oznacza cyfrę 3, a potem w; Ctrl-r oznacza przytrzymanie Control i naciśnięcie r. Znaki +, -, |, $, %, średnik oraz przecinek są rzeczywistymi klawiszami poleceń.

Polecenie takie jak ciw składa się z operatora c i obiektu tekstowego iw. Nazwy i teksty docelowe widoczne na tablicach należy wpisywać dokładnie, nawet gdy instrukcja jest w innym języku. Po poleceniach zaczynających się od : oraz wyszukiwaniu / lub ? naciśnij Enter. Sekwencje wpisuj kolejno, bez strzałek rozdzielających kroki. Oznaczenia <Esc>, <Enter>, <C-r> i <Space> znaczą odpowiednio klawisz Escape, Enter, Control+r i spację; nawiasów ostrych nie wpisuje się.

Gra a prawdziwy Vim

LUMEN jest edukacyjnym symulatorem wybranego zakresu Vima. Plansza reprezentuje bufor tekstowy, a jej widoczny fragment jest oknem. Polecenia przewijania zmieniają okno, nie treść. Pliki są wirtualne: gra nie odczytuje katalogów komputera i nie uruchamia poleceń systemowych.

Znaczenie komend odpowiada ćwiczonym operacjom, lecz gra nie zastępuje pełnego Vima z jego opcjami, wtyczkami i wszystkimi wyjątkami składni. Pomoc w grze oraz ten podręcznik opisują obsługiwany zakres.

Wiersz poleceń

Dwukropek otwiera wiersz poleceń. :w zapisuje wirtualny dokument, :e otwiera inny, :ls pokazuje bufory, a :b przełącza bufor. :marks wyświetla znaczniki; :delmarks i skrót :delm je usuwają, a :delmarks! czyści znaczniki lokalne. :reg pokazuje zawartość rejestrów.

:set nu i :set nonu włączają i wyłączają bezwzględne numery wierszy; :set nu! je przełącza. Odpowiednio :set rnu, :set nornu i :set rnu! sterują numerami względnymi. Polecenie zatwierdza Enter, a Esc je anuluje.

Rejestry

Rejestr jest schowkiem Vima. Zwykłe kopiowanie i usuwanie korzystają z rejestru nienazwanego; prefiks cudzysłowu wybiera inny rejestr. Małe litery a-z oznaczają rejestry nazwane, a zapis do A-Z dopisuje do odpowiadającego rejestru małej litery.

Rejestr 0 zachowuje ostatnią kopię, rejestry 1-9 historię większych usunięć, a rejestr - niewielkie usunięcie. Rejestr _ to czarna dziura: usuwa bez zastępowania pozostałych schowków. :reg pomaga sprawdzić, co rzeczywiście zostało zapamiętane.

Makra

Makro zapamiętuje serię działań, także ruchy kursora. Aby nagrać makro w rejestrze a, naciśnij qa, wykonaj potrzebne operacje i zakończ nagrywanie klawiszem q. @a odtwarza rejestr a, a @@ powtarza ostatnio odtworzone makro. Liczba przed odtworzeniem określa liczbę powtórzeń.

Przed wielokrotnym odtworzeniem sprawdź jedno wykonanie: dobra automatyzacja powinna kończyć się w przewidywalnej pozycji, gotowej na następny fragment tekstu.

Wirtualne pliki i bufory

Bufor jest otwartym dokumentem, a nazwa pliku identyfikuje jego wirtualny zapis. :w [nazwa] zachowuje tekst, :e nazwa otwiera dokument, :ls pokazuje listę, a :b nazwa lub numer przełącza bufor. Przed zmianą dokumentu zapisuj zamierzone poprawki.

Wszystko dzieje się wewnątrz gry i jej lokalnego zapisu. Eksport podróży służy do kopii zapasowej, nie zapisuje prawdziwego pliku źródłowego na dysku. Nie wpisuj do ćwiczeń haseł ani innych poufnych informacji.

<Esc> = Escape; <Enter> = Enter; <C-r> = Ctrl+r; <Space> = spacja. Strzałka oddziela fragmenty i nie jest klawiszem. Zachowuj wielkość liter.

Atlas poleceń

Polecenie	Działanie
`0`	Przejdź do pierwszej kolumny bieżącego wersu.
`hjkl`	h w lewo, j w dół, k w górę, l w prawo. Podstawowe ruchy kursora.
`h`	Jedno pole lub znak w lewo.
`j`	Jeden wers lub pole w dół.
`k`	Jeden wers lub pole w górę.
`l`	Jedno pole lub znak w prawo.
`w`	Przejdź do początku następnego słowa.
`b`	Wróć do początku słowa.
`e`	Przejdź na koniec słowa.
`^`	Przejdź do pierwszego znaku wersu, który nie jest spacją.
`$`	Przejdź na ostatni znak bieżącego wersu.
`gg`	Przejdź do pierwszego wersu.
`G`	Przejdź do ostatniego wersu.
`count`	Liczba powtarza następny ruch, np. 3w lub 2j.
`f`	Naciśnij f i znak, aby znaleźć jego następne wystąpienie w bieżącym wersie.
`;`	Powtórz ostatnie wyszukiwanie znaku przez f.
`/`	Wyszukaj tekst naprzód i zatwierdź Enter.
`n`	Powtórz wyszukiwanie w jego dotychczasowym kierunku.
`N`	Powtórz wyszukiwanie w kierunku przeciwnym do ostatniego / lub ?.
`x`	Usuń znak pod kursorem.
`r`	Zastąp znak pod kursorem następnym wpisanym znakiem.
`dd`	Wytnij bieżący wers i zachowaj go w rejestrze.
`dw`	Usuń od kursora do początku następnego słowa.
`u`	Cofnij ostatnią zmianę tekstu.
`Ctrl-r`	Ponów cofniętą zmianę.
`yy`	Skopiuj cały bieżący wers do rejestru.

Polecenie	Działanie
`yw`	Skopiuj tekst do początku następnego słowa.
`p`	Wklej tekst za kursorem, a całe wersy poniżej bieżącego.
`i`	Rozpocznij wstawianie przed kursorem.
`a`	Rozpocznij wstawianie za kursorem.
`A`	Rozpocznij wstawianie na końcu wersu.
`o`	Otwórz nowy wers poniżej i rozpocznij wstawianie.
`Esc`	Wróć do trybu normalnego lub anuluj oczekujące polecenie.
`:w`	Zapisz wirtualny dokument; opcjonalna nazwa wskazuje plik w grze.
`1 ... 9`	Liczba przed operatorem lub ruchem określa powtórzenia; 2d3w łączy oba liczniki.
`W`	Do początku następnego ciągu bez spacji; interpunkcja pozostaje częścią WORD.
`E`	Na koniec WORD, czyli ciągu znaków bez spacji.
`B`	Wstecz do początku WORD; znaki interpunkcyjne nie rozbijają go na słowa.
`ge`	Wstecz na koniec poprzedniego słowa.
`gE`	Wstecz na koniec poprzedniego WORD.
`+`	Do pierwszego znaku innego niż spacja w następnym wersie; liczba powtarza ruch.
`-`	Do pierwszego znaku innego niż spacja w poprzednim wersie.
`|`	Do kolumny wskazanej liczbą, np. 5|; bez liczby do kolumny 1.
`F`	Naciśnij F i znak, aby znaleźć go wstecz w bieżącym wersie.
`t`	Przed następnym wystąpieniem znaku w wersie: t i szukany znak.
`T`	Wstecz, zatrzymując się tuż za znalezionym znakiem.
`,`	Powtórz ostatnie f, F, t lub T w przeciwnym kierunku.
`*`	Wyszukaj naprzód całe słowo pod kursorem.
`#`	Wyszukaj wstecz całe słowo pod kursorem.
`?`	Otwórz wyszukiwanie wstecz; tekst zatwierdź Enter.
`%`	Przejdź do pasującego nawiasu (), [] lub {}.
`[{`	Wstecz do otwierającego nawiasu klamrowego otaczającego bloku.
`]}`	Naprzód do zamykającego nawiasu klamrowego otaczającego bloku.
`[(`	Wstecz do otaczającego otwierającego nawiasu okrągłego.

Polecenie	Działanie
`])`	Naprzód do otaczającego zamykającego nawiasu okrągłego.
`{`	Wstecz do granicy akapitu; puste wersy rozdzielają akapity.
`}`	Naprzód do granicy następnego akapitu.
`(`	Wstecz do początku zdania.
`)`	Naprzód do początku następnego zdania.
`zt`	Przewiń widok tak, aby bieżący wers znalazł się na górze.
`zz`	Wyśrodkuj bieżący wers w widoku, nie zmieniając tekstu.
`zb`	Przewiń bieżący wers na dół widoku.
`H`	Do pierwszego widocznego wersu okna.
`M`	Do środkowego widocznego wersu okna.
`L`	Do ostatniego widocznego wersu okna.
`nu / :set nu`	Włącz bezwzględne numery wierszy.
`nonu / :set nonu`	Wyłącz bezwzględne numery wierszy.
`nu! / :set nu!`	Przełącz widoczność bezwzględnych numerów wierszy.
`rnu / :set rnu`	Włącz numery względne, wskazujące odległość od kursora.
`nornu / :set nornu`	Wyłącz względne numery wierszy.
`rnu! / :set rnu!`	Przełącz względne numery wierszy.
`X`	Usuń znak przed kursorem.
`D`	Usuń od kursora do końca wersu.
`~`	Zmień wielkość litery pod kursorem.
`I`	Wstawiaj przy pierwszym znaku wersu innym niż spacja.
`O`	Otwórz nowy wers nad bieżącym i rozpocznij wstawianie.
`s`	Usuń znak pod kursorem i rozpocznij wstawianie.
`S`	Zastąp cały bieżący wers, przechodząc do wstawiania.
`d`	Operator usuwania: dodaj ruch lub obiekt, np. d$, diw lub d2w.
`y`	Operator kopiowania: połącz z ruchem lub obiektem tekstowym.
`c`	Operator zmiany: usuń wskazany obszar i rozpocznij wstawianie.
`cc`	Zmień cały bieżący wers i rozpocznij wstawianie.

Polecenie	Działanie
`C`	Zmień od kursora do końca wersu i rozpocznij wstawianie.
`Y`	Skopiuj cały bieżący wers; odpowiednik yy w tej grze.
`.`	Powtórz ostatnią zmianę tekstu, łącznie z wpisaną treścią.
`iw`	Wnętrze bieżącego słowa bez otaczających spacji; użyj z d, y lub c.
`aw`	Całe słowo wraz z przyległą spacją; obiekt dla operatora.
`iW`	Wnętrze WORD: ciąg bez białych znaków, także z interpunkcją.
`aW`	WORD wraz z otaczającą białą przestrzenią.
`is`	Wnętrze zdania, bez oddzielającej je białej przestrzeni.
`as`	Zdanie wraz z oddzielającą białą przestrzenią.
`ip`	Wnętrze akapitu bez otaczających pustych wersów.
`ap`	Akapit razem z oddzielającymi pustymi wersami.
`i[ / i]`	Wnętrze pasującej pary [] bez nawiasów; oba aliasy znaczą to samo.
`a[ / a]`	Cała para [] razem z nawiasami.
`i( / i) / ib`	Wnętrze pasującej pary (), bez nawiasów; ib jest aliasem.
`a( / a) / ab`	Cała para () z nawiasami; ab jest aliasem.
`i{ / i} / iB`	Wnętrze pasującej pary {}, bez klamer; iB jest aliasem.
`a{ / a} / aB`	Cała para {} razem z klamrami; aB jest aliasem.
`i< / i>`	Wnętrze pasującej pary <>, bez nawiasów ostrych.
`a< / a>`	Cała para <> wraz z nawiasami ostrymi.
`i"`	Tekst wewnątrz cudzysłowu podwójnego, bez znaków cudzysłowu.
`a"`	Tekst razem z otaczającymi cudzysłowami podwójnymi.
`i'`	Tekst wewnątrz pojedynczych apostrofów, bez ograniczników.
`a'`	Tekst wraz z otaczającymi pojedynczymi apostrofami.
`` i` ``	Tekst wewnątrz znaków `, bez tych ograniczników.
`` a` ``	Tekst wraz z otaczającymi znakami `.
`it`	Wnętrze pasującej pary znaczników HTML, bez samych znaczników.
`at`	Cały element wraz z otwierającym i zamykającym znacznikiem.
`m`	Ustaw znacznik: m i litera, np. ma, zapamiętują bieżącą pozycję.

Polecenie	Działanie
`'`	Apostrof i litera znacznika prowadzą do pierwszego znaku jego wersu.
`` ` ``	Znak `` ` `` i litera znacznika prowadzą do dokładnie zapamiętanej kolumny.
`:marks`	Pokaż zapamiętane znaczniki.
`:delmarks / :delm`	Usuń wskazane znaczniki, np. :delmarks a; :delm jest skrótem.
`:delmarks!`	Usuń wszystkie lokalne znaczniki.
`"`	Prefiks wyboru rejestru: cudzysłów, nazwa rejestru, potem operator lub p.
`""`	Rejestr nienazwany przechowuje ostatnio kopiowany lub usuwany tekst.
`"-`	Rejestr małego usunięcia: fragment krótszy niż cały wers.
`"_`	Czarna dziura: usuń bez nadpisywania innych rejestrów.
`"0`	Rejestr 0 zachowuje ostatnie kopiowanie, także po późniejszym usunięciu.
`:reg / :registers`	Pokaż rejestry; można wskazać ich nazwy, np. :reg a0.
`:e`	Otwórz wirtualny plik pod wskazaną nazwą.
`:ls`	Pokaż otwarte bufory i ich numery.
`:b`	Przełącz wirtualny bufor według nazwy lub numeru.
`q`	q i litera rozpoczynają nagrywanie makra; kolejne q je kończy.
`@`	Odtwórz makro z wybranego rejestru, np. @a; liczba daje powtórzenia.
`@@`	Ponownie odtwórz ostatnio użyte makro.
`"1 ... "9`	Historia większych usunięć: 1 to najnowsze, 9 najstarsze z zachowanych.
`"a ... "z`	Rejestr nazwany a-z: zachowaj tekst przez wskazanie litery przed operatorem.
`"A ... "Z`	Wielka litera A-Z dopisuje do odpowiedniego rejestru a-z zamiast go zastępować.
`P`	Wklej tekst przed kursorem lub cały wers nad bieżącym.
`:set`	Zmień opcje widoku, np. :set nu albo :set rnu!.
`:`	Otwórz wiersz poleceń; wpisz nazwę polecenia i zatwierdź Enter.
`z`	Prefiks przewijania: dokończ przez t, z lub b.
`:undo`	Cofnij ostatnią zmianę; wariant wiersza poleceń klawisza u.
`:redo`	Ponów cofniętą zmianę; wariant wiersza poleceń Ctrl-r.
`:delm!`	Skrót :delmarks!: usuń wszystkie lokalne znaczniki.

Przewodnik po krainach

Najpierw spróbuj samodzielnie. Rozwiązania są przykładami; inne poprawne sekwencje także mogą spełnić cel.

<Esc> = Escape; <Enter> = Enter; <C-r> = Ctrl+r; <Space> = spacja. Strzałka oddziela fragmenty i nie jest klawiszem. Zachowuj wielkość liter.

01 / Port Latarni

Każda podróż zaczyna się od ruchu

Morze przyniosło Cię do portu, w którym zgasły wszystkie latarnie. Neri znalazła zagubione iskry między kamieniami nabrzeża. Podnieś je, a zarys pierwszej ścieżki wróci do świata.

Zadanie 01 / Pierwsza iskra

Przesuń się w prawo przez l, zbierz iskrę, a potem wejdź wyżej przez k i dotrzyj do bramy.

Początek: wiersz 4, kolumna 3

Zbierz światła: (4, 6). Cel kursora: (2, 6).

```
1     #########
2     #.......#
3     #.......#
4     #.......#
5     #########
```

Przećwicz polecenia: l, k

Wskazówki: l przesuwa w prawo, k w górę. W tym zadaniu użyj obu tych klawiszy. Po zebraniu iskry droga jeszcze się nie kończy: dotrzyj do rozświetlonej bramy.

Przykładowa droga

```
lllkk
```

Zadanie 02 / Schody przypływu

Użyj h i j: zejdź z górnej ścieżki, zabierz iskrę i wróć do bramy po lewej.

Początek: wiersz 2, kolumna 8

Zbierz światła: (4, 6). Cel kursora: (4, 3).

```
1     #########
2     #.......#
3     #.......#
4     #.......#
5     #########
```

Przećwicz polecenia: h, j

Wskazówki: h prowadzi w lewo, j w dół. Potrzebujesz obu ruchów. Najpierw zbliż się do iskry, potem zejdź do niej przez j. Zebrane światło zabierz do bramy.

Przykładowa droga

```
hhjjhhh
```

Zadanie 03 / Rozświetl nabrzeże

Zbierz wszystkie pozostałe iskry. Gdy wróci światło, port pokaże drogę ku następnej wyspie.

Początek: wiersz 2, kolumna 2

Zbierz światła: (2, 8), (6, 8), (6, 2). Cel kursora: (4, 4).

```
1     #########
2     #.......#
3     #.#####.#
4     #...#...#
5     #.#####.#
6     #.......#
7     #########
```

Wskazówki: Przed pierwszym ruchem spójrz na całą ścieżkę i położenie świateł. Jeśli droga się kończy, wróć kilka pól. Możesz spokojnie próbować; pośpiech nie jest częścią zadania.

Przykładowa droga

`lllllljjjjhhhhhhkkll`

02 / Gaj Słów

Myśl słowami, nie pojedynczymi krokami

W tym gaju drzewa wyrastają ze słów. Cisza rozsunęła ich korzenie, a Ivo nie potrafi już przejść między zdaniami. Pokaże Ci ruchy, które odnajdują początek i koniec całych słów.

Zadanie 04 / Od korzenia do korzenia

Dotrzyj do wskazanego miejsca, przeskakując między początkami słów za pomocą w.

Początek: wiersz 1, kolumna 1

Cel kursora: wiersz 1, kolumna 10.

```
1     dew moss lantern river
```

Przećwicz polecenia: w

Wskazówki: w przesuwa do początku następnego słowa. Spacje oddzielają wyspy słów. w pokonuje taki odstęp jednym ruchem.

Przykładowa droga

ww

Zadanie 05 / Ślad za plecami

Odszukaj wcześniejsze słowo. Użyj b, aby wracać po ścieżce zapisów.

Początek: wiersz 1, kolumna 10

Cel kursora: wiersz 1, kolumna 5.

```
1     dew moss lantern river
```

Przećwicz polecenia: b

Wskazówki: b cofa do początku słowa. Jeśli jesteś w środku słowa, pierwsze b wróci do jego początku. Następne zabierze Cię dalej wstecz.

Przykładowa droga

b

Zadanie 06 / Na końcu gałęzi

Zatrzymaj się na ostatniej literze właściwego słowa. Wypróbuj e.

Początek: wiersz 1, kolumna 1

Cel kursora: wiersz 1, kolumna 11.

```
1     soft silver light
```

Przećwicz polecenia: e

Wskazówki: e przesuwa na koniec słowa. Spójrz na podświetlony cel. Początek i koniec słowa to różne miejsca: wybierz ruch, który trafia dokładnie tam.

Przykładowa droga

we

03 / Mosty Wersów

Od jednej krawędzi do drugiej

Kamienne mosty pamiętają tylko początki i końce. Sela nauczy Cię odnajdywać ich krawędzie i łączyć ruch z liczbą. Dalekie przejście nie musi składać się z wielu osobnych naciśnięć.

Zadanie 07 / Pierwszy kamień

Odwiedź początek wersu przez 0, a następnie pierwszą literę przez ^.

Początek: wiersz 1, kolumna 10

Cel kursora: wiersz 1, kolumna 4.

```
1        stone bridge
```

Przećwicz polecenia: 0, ^

Wskazówki: 0 prowadzi do pierwszej kolumny, nawet jeśli jest tam spacja. ^ pomija początkowe spacje. Naciśnij 0, a potem ^. Celem jest pierwsza litera, lecz w tym zadaniu poznajesz oba początki.

Przykładowa droga

`0^`

Zadanie 08 / Drugi brzeg

Skocz przez G do ostatniego wersu i dotrzyj na jego koniec przez $.

Początek: wiersz 1, kolumna 1

Cel kursora: wiersz 3, kolumna 16.

```
1     a quiet river
2     under the bridge
3     the last lantern
```

Przećwicz polecenia: G, $

Wskazówki: G prowadzi do ostatniego wersu całego zapisu. $ do ostatniego znaku bieżącego wersu. Połącz oba ruchy: G, potem $. Nie musisz schodzić wers po wersie.

Przykładowa droga

`G$`

Zadanie 09 / Długi krok

Wróć na górę przez gg, przesuń się o trzy wersy przez 3j i wybierz pierwszą kolumnę przez 0.

Początek: wiersz 5, kolumna 5

Cel kursora: wiersz 4, kolumna 1.

```
1     north gate
2     mist
3     old steps
4     star bridge
5     last lantern
```

Przećwicz polecenia: gg, count, 0

Wskazówki: gg zabiera na pierwszy wers. Liczba 3 przed j powtarza ten ruch trzy razy. Użyj gg, potem 3j, a na końcu 0. Liczba jest częścią polecenia, a nie tekstem do wpisania.

Przykładowa droga

```
gg3j0
```

04 / Obserwatorium Echa

Znajdź to, czego nie widać od razu

Nad wyspą wciąż świecą gwiazdy, lecz ich nazwy rozsypały się po tablicach. Orin słyszy powracające echa. Naucz się wyszukiwać znaki i słowa, a odczytasz z nich drogę.

Zadanie 10 / Jedna wybrana gwiazda

Znajdź literę l w bieżącym wersie za pomocą f.

Początek: wiersz 1, kolumna 1

Cel kursora: wiersz 1, kolumna 8.

```
1     moss | lantern | stars
```

Przećwicz polecenia: f

Wskazówki: f czeka na szukany znak. W tym zadaniu jest nim l. Naciśnij f, potem l. Zatrzymasz się na następnym l bez przechodzenia przez każdą literę.

Przykładowa droga

`fl`

Zadanie 11 / Echo znaku

Znajdź następne b przez fb, a potem powtórz to wyszukanie średnikiem ;.

Początek: wiersz 1, kolumna 1

Cel kursora: wiersz 1, kolumna 11.

```
1     bell bell bell
```

Przećwicz polecenia: f, ;

Wskazówki: Po f naciśnij b. ; powtarza ostatnie wyszukiwanie pojedynczego znaku. Użyj fb, potem ;, aby dotrzeć do kolejnego b w tym samym wersie.

Przykładowa droga

`fb;`

Zadanie 12 / Powracające echo

Wyszukaj light przez /light i Enter. Użyj n, N i znów n, aby odnaleźć echo na końcowej tablicy.

Początek: wiersz 1, kolumna 1

Cel kursora: wiersz 3, kolumna 10.

```
1     light sleeps
2     follow light
3     wake the light
```

Przećwicz polecenia: /, n, N

Wskazówki: n szuka kolejnego wystąpienia w tym samym kierunku, N wraca w kierunku przeciwnym. Wpisz /light, potwierdź Enter, a następnie użyj n, N, n. W tym zadaniu przećwicz oba kierunki.

Przykładowa droga

`/light → <Enter> → nNn`

05 / Kuźnia Żaru

Naprawiaj śmiało. Zawsze możesz cofnąć zmianę

W palenisku zostało dość żaru, by wykuć nowe litery. Asha nie wymaga bezbłędnych ruchów: dobry rzemieślnik potrafi również cofnąć nietrafioną zmianę. Oczyść uszkodzone zapisy i rozpal piec.

Zadanie 13 / Odłamek w zapisie

Usuń zbędny znak za pomocą x i przywróć właściwy tekst.

Początek: wiersz 1, kolumna 6

Stan początkowy

```
1     sparkx
```

Docelowy tekst

```
1     spark
```

Przećwicz polecenia: x

Wskazówki: x usuwa znak pod kursorem. Najpierw ustaw kursor na zbędnej literze. Po usunięciu spójrz na cały tekst, nie tylko na miejsce kursora.

Przykładowa droga

x

Zadanie 14 / Pamięć metalu

Zamień pierwsze c na g przez rg, cofnij zmianę przez u, a następnie ponów ją przez Ctrl-r.

Początek: wiersz 1, kolumna 1

Stan początkowy

```
1     cold ember
2     cold ember
```

Docelowy tekst

```
1     gold ember
2     cold ember
```

Przećwicz polecenia: r, u, Ctrl-r

Wskazówki: r czeka na znak zastępujący literę pod kursorem. rg zmienia cold na gold. Po rg użyj u, a potem Ctrl-r. Cel to gold ember w pierwszym wersie i cold ember w drugim.

Przykładowa droga

`rg → u → <C-r>`

Zadanie 15 / Nowy kształt

Usuń pierwsze broken przez dw, a potem wytnij pozostały uszkodzony wers przez dd. Zostaw tylko dawn.

Początek: wiersz 1, kolumna 1

Stan początkowy

```
1  broken broken
2  dawn
```

Docelowy tekst

```
1  dawn
```

Przećwicz polecenia: dw, dd

Wskazówki: dw usuwa fragment do następnego słowa. dd usuwa cały bieżący wers. Najpierw dw, następnie dd. Dwa różne rozmiary uszkodzeń, dwa narzędzia.

Przykładowa droga

`dwdd`

06 / Biblioteka Luster

Dobra myśl zasługuje na powtórzenie

Na półkach stoją księgi, których strony odbijają się w wodzie. Mira potrafi zachować słowo, zanim zniknie. Dzięki rejestrowi skopiujesz światło, a nawet przeniesiesz cały wers w nowe miejsce.

Zadanie 16 / Bliźniaczy wers

Skopiuj wers za pomocą yy i umieść jego odbicie poniżej przez p.

Początek: wiersz 1, kolumna 1

Stan początkowy

```
1     light
2     sleep
```

Docelowy tekst

```
1     light
2     light
3     sleep
```

Przećwicz polecenia: yy, p

Wskazówki: yy kopiuje cały bieżący wers, nie usuwając oryginału. Po skopiowaniu p wkleja wers pod bieżącym. Najpierw wybierz miejsce, pod którym ma znaleźć się kopia.

Przykładowa droga

`yyp`

Zadanie 17 / Słowo w lustrze

Zachowaj słowo przez yw i wklej je w odpowiednie miejsce poleceniem p.

Początek: wiersz 1, kolumna 1

Stan początkowy

```
1     glow moss
```

Docelowy tekst

```
1     glow glow moss
```

Przećwicz polecenia: yw, p

Wskazówki: yw kopiuje tekst do początku następnego słowa. Przy kopiowaniu fragmentu p wkleja go za kursorem. To inne zachowanie niż wklejanie całego wersu.

Przykładowa droga

`ywwhp`

Zadanie 18 / Nowa kolejność

Wytnij wers przez dd i przenieś go za pomocą p, aby odtworzyć właściwą kolejność.

Początek: wiersz 1, kolumna 1

Stan początkowy

```
1  mist
2  sun
3  sky
```

Docelowy tekst

```
1  sun
2  mist
3  sky
```

Przećwicz polecenia: dd, p

Wskazówki: dd odkłada usunięty wers do tego samego rejestru, z którego korzysta p. Ustaw kursor w miejscu przeznaczenia po wycięciu. p umieści zapamiętany wers poniżej.

Przykładowa droga

`ddp`

07 / Pracownia Świtu

Pozostaw po sobie nowy zapis

Eno odnalazł puste miejsca w kronice. Tych słów nie da się odzyskać samym ruchem: trzeba je napisać. W pracowni odkryjesz rytm przechodzenia między działaniem a wstawianiem tekstu.

Zadanie 19 / Słowo, którego brak

Ustaw kursor, wejdź do wstawiania przez i i dopisz brakujący tekst.

Początek: wiersz 1, kolumna 5

Stan początkowy

```
1     the returns
```

Docelowy tekst

```
1     the dawn returns
```

Przećwicz polecenia: i, Esc

Wskazówki: i rozpoczyna wpisywanie przed kursorem. W trybie wstawiania litery stają się tekstem. Esc przywraca tryb normalny i polecenia ruchu.

Przykładowa droga

`i → dawn<Space> → <Esc>`

Zadanie 20 / Miejsce na więcej

Dopisz s przez a, wróć przez Esc, a potem dopisz spację i shine na końcu wersu przez A.

Początek: wiersz 1, kolumna 4

Stan początkowy

```
1     star
```

Docelowy tekst

```
1     stars shine
```

Przećwicz polecenia: a, A, Esc

Wskazówki: a rozpoczyna wpisywanie za kursorem. A rozpoczyna je na końcu całego wersu. Zamień star na stars przez a, s, Esc. Następnie A, spacja, shine, Esc dają stars shine.

Przykładowa droga

`as → <Esc> → A<Space>shine → <Esc>`

Zadanie 21 / Pieczęć kroniki

Otwórz nowy wers przez o, wpisz returns, wróć przez Esc i zapisz :w oraz Enter.

Początek: wiersz 1, kolumna 1

Stan początkowy

```
1      dawn
```

Docelowy tekst

```
1      dawn
2      returns
```

Przećwicz polecenia: o, Esc, :w

Wskazówki: o otwiera nowy wers poniżej i od razu włącza wstawianie. Pod dawn wpisz returns. Potem Esc, :w i Enter zatwierdzają ukończoną tablicę.

Przykładowa droga

`oreturns → <Esc> → :w → <Enter>`

08 / Wieża Poranka

Wszystkie ścieżki prowadzą ku światłu

Na szczycie czeka soczewka złożona z odłamków pierwszych wysp. Połącz poznane umiejętności, by przywołać pierwszy świt. Jego światło odsłoni dalsze krainy i drogę do mistrzostwa.

Zadanie 22 / Mapa w pamięci

Wytnij uszkodzony pierwszy wers przez dd, skopiuj dobry przez yy, wklej go przez p i zatwierdź :w.

Początek: wiersz 1, kolumna 1

Stan początkowy

```
1     lost lost
2     keep the light
```

Docelowy tekst

```
1     keep the light
2     keep the light
```

Przećwicz polecenia: dd, yy, p, :w

Wskazówki: Po dd pozostaje keep the light. yy zachowuje cały ten wers, a p tworzy jego kopię poniżej. Użyj dd, yy, p, następnie :w i Enter. Na tablicy mają zostać dwa identyczne wersy keep the light.

Przykładowa droga

`ddyyp:w → <Enter>`

Zadanie 23 / Ostatnia naprawa

Znajdź sleep przez /sleep i Enter. Usuń je przez dw, dopisz sing na końcu przez A i zapisz :w.

Początek: wiersz 1, kolumna 1

Stan początkowy

```
1     moss covers dawn
2     the bells sleep
```

Docelowy tekst

```
1     moss covers dawn
2     the bells sing
```

Przećwicz polecenia: /, dw, A, :w

Wskazówki: Wyszukiwanie ustawi Cię na sleep. dw usunie to słowo, a A pozwoli dokończyć wers. Zostaw pierwszy wers bez zmian. W drugim ma być the bells sing. Zakończ wstawianie przez Esc, potem :w i Enter.

Przykładowa droga

`/sleep → <Enter> → dwAsing → <Esc> → :w → <Enter>`

Zadanie 24 / Niech wróci światło

Usuń pierwszy wers, skopiuj pozostały, a w nowym wersie dopisz the dawn returns. Zapisz odtworzoną kronikę.

Początek: wiersz 1, kolumna 1

Stan początkowy

```
1     the night stays
2     one small spark
```

Docelowy tekst

```
1     one small spark
2     one small spark
3     the dawn returns
```

Przećwicz polecenia: dd, yy, p, o, :w

Wskazówki: dd usuwa uszkodzony wers, yy i p tworzą drugie one small spark. o otwiera miejsce na ostatni wers. Po dd, yy, p użyj o i wpisz the dawn returns. Następnie Esc, :w i Enter przywołają pierwszy świt i otworzą dalszą drogę.

Przykładowa droga

`ddyypothe<Space>dawn<Space>returns → <Esc> → :w → <Enter>`

09 / Klify Wiatru

Słowa większe niż słowa

Wiatr niesie porwane wstęgi zapisu. Tavi pokazuje, jak przeskoczyć cały zapisany na wstędze fragment i wrócić po śladach bez liczenia liter.

Zadanie 25 / Wstęga bez przerw

Przeskocz całe one-two przez W i stań na początku three.four. Wymagane polecenia: W.

Początek: wiersz 1, kolumna 1

Cel kursora: wiersz 1, kolumna 9.

```
1     one-two three.four five
```

Przećwicz polecenia: W

Wskazówki: Do początku następnego ciągu bez spacji; interpunkcja pozostaje częścią WORD. Najpierw wykonaj jeden świadomy krok i sprawdź kursor lub tekst. Powtórzenia dodawaj dopiero, gdy rozumiesz pojedynczą operację.

Przykładowa droga

`W`

Zadanie 26 / Powrót granią

Porównaj E i B z ge oraz gE, a na końcu wróć do pierwszej kolumny. Wymagane polecenia: E · B · ge · gE.

Początek: wiersz 1, kolumna 23

Cel kursora: wiersz 1, kolumna 1.

```
1     one-two three.four five
```

Przećwicz polecenia: E, B, ge, gE

Wskazówki: Na koniec WORD, czyli ciągu znaków bez spacji. Wstecz do początku WORD; znaki interpunkcyjne nie rozbijają go na słowa.

Przykładowa droga

`BEgegE0`

Zadanie 27 / Kroki odmierzone

Przećwicz + i -, dodaj licznik do ruchu i dotrzyj do litery t w third przez wybór kolumny. Wymagane polecenia: + · - · | · licznik.

Początek: wiersz 1, kolumna 4

Cel kursora: wiersz 3, kolumna 2.

```
1      first
2        second
3      third
```

Przećwicz polecenia: +, -, |, count

Wskazówki: Do pierwszego znaku innego niż spacja w następnym wersie; liczba powtarza ruch. Do pierwszego znaku innego niż spacja w poprzednim wersie.

Przykładowa droga

`+-2+2|`

10 / Przełęcz Tropicieli

Szukaj również za sobą

Na przełęczy drogę znaczą pojedyncze nacięcia. Ryn uczy patrzeć w obie strony i zatrzymywać się dokładnie tam, gdzie zaczyna się bezpieczny kamień.

Zadanie 28 / Pod prąd śladu

Znajdź a wstecz, wypróbuj t oraz T i odwróć kierunek przecinkiem, kończąc tuż przed ostatnim a. Wymagane polecenia: F · t · T · ,.

Początek: wiersz 1, kolumna 13

Cel kursora: wiersz 1, kolumna 12.

```
1     a x a y a z a
```

Przećwicz polecenia: F, t, T, ,

Wskazówki: Naciśnij F i znak, aby znaleźć go wstecz w bieżącym wersie. Przed następnym wystąpieniem znaku w wersie: t i szukany znak.

Przykładowa droga

`FataTa,`

Zadanie 29 / Słowo przywołuje

Prześledź całe słowo glow w obie strony przez * i #. Zakończ na drugim glow pierwszego wersu. Wymagane polecenia: * · #.

Początek: wiersz 1, kolumna 1

Cel kursora: wiersz 1, kolumna 11.

```
1     glow moss glow
2     glow
```

Przećwicz polecenia: *, #

Wskazówki: Wyszukaj naprzód całe słowo pod kursorem. Wyszukaj wstecz całe słowo pod kursorem.

Przykładowa droga

#

Zadanie 30 / Droga wstecz

Szukaj star wstecz przez ?, przećwicz n i N, a potem zatrzymaj się na środkowym wystąpieniu. Wymagane polecenia: ? · n · N.

Początek: wiersz 3, kolumna 1

Cel kursora: wiersz 2, kolumna 1.

```
1     star
2     star
3     star
```

Przećwicz polecenia: ?, n, N

Wskazówki: Otwórz wyszukiwanie wstecz; tekst zatwierdź Enter. Powtórz wyszukiwanie w jego dotychczasowym kierunku.

Przykładowa droga

`?star → <Enter> → nN`

11 / Kamienne Kręgi

Każde otwarcie ma domknięcie

Uma strzeże kręgów ukrytych jeden w drugim. Odczytaj nawiasy, granice zdań i oddechy akapitów, a kamienie przestaną wyglądać jak labirynt.

Zadanie 31 / Para kamieni

Zbadaj zagnieżdżone nawiasy przez % i ruchy granic bloków. Wróć do otwierającej klamry zewnętrznej. Wymagane polecenia: % · [(·]) · [{ ·]}.

Początek: wiersz 2, kolumna 5

Cel kursora: wiersz 1, kolumna 1.

```
1     {
2       [a(b)c]
3     }
```

Przećwicz polecenia: %, [(,]), [{,]}

Wskazówki: Przejdź do pasującego nawiasu (), [] lub {}. Wstecz do otaczającego otwierającego nawiasu okrągłego.

Przykładowa droga

`% → [( → ]) → [{ → ]} → %`

Zadanie 32 / Między akapitami

Przejdź naprzód i wstecz między akapitami. Zatrzymaj się na pustym wersie oddzielającym north od south. Wymagane polecenia: { · }.

Początek: wiersz 1, kolumna 1

Cel kursora: wiersz 3, kolumna 1.

```
1     north
2     path
3     [pusty wiersz]
4     south
5     gate
6     [pusty wiersz]
7     dawn
```

Przećwicz polecenia: {, }

Wskazówki: Wstecz do granicy akapitu; puste wersy rozdzielają akapity. Naprzód do granicy następnego akapitu.

Przykładowa droga

`}}{`

Zadanie 33 / Miejsce na oddech

Przejdź przez początki zdań w obu kierunkach i wróć do początku Second. Wymagane polecenia: (·).

Początek: wiersz 1, kolumna 1

Cel kursora: wiersz 1, kolumna 14.

```
1     First light. Second light! Third light?
```

Przećwicz polecenia: (,)

Wskazówki: Wstecz do początku zdania. Naprzód do początku następnego zdania.

Przykładowa droga

```
)))(
```

12 / Wieża Horyzontów

Zmień perspektywę, nie treść

Okna wieży przesuwają się wzdłuż zwoju. Noa pokaże Ci różnicę między miejscem kursora a tym, co akurat widać przez okno.

Zadanie 34 / Wędrujące okno

Wypróbuj trzy ustawienia okna. Na końcu pokaż bieżący, piąty wers przy górnej krawędzi. Wymagane polecenia: zt · zz · zb.

Początek: wiersz 5, kolumna 1

Cel kursora: wiersz 5, kolumna 1.

```
1     line one
2     line two
3     line three
4     line four
5     line five
6     line six
7     line seven
8     line eight
9     line nine
```

Górny widoczny wiersz: 5

Przećwicz polecenia: zt, zz, zb

Wskazówki: Przewiń widok tak, aby bieżący wers znalazł się na górze. Wyśrodkuj bieżący wers w widoku, nie zmieniając tekstu.

Przykładowa droga

```
ztzzzbzt
```

Zadanie 35 / Stały horyzont

Wyśrodkuj piąty wers, odwiedź górę i dół okna, a następnie wróć do jego środka. Wymagane polecenia: H · M · L.

Początek: wiersz 5, kolumna 1

Cel kursora: wiersz 5, kolumna 1.

```
1     line one
2     line two
3     line three
4     line four
5     line five
6     line six
7     line seven
8     line eight
9     line nine
```

Górny widoczny wiersz: 3

Przećwicz polecenia: H, M, L

Wskazówki: Do pierwszego widocznego wersu okna. Do środkowego widocznego wersu okna.

Przykładowa droga

`zzHLM`

Zadanie 36 / Liczby przy drodze

Przećwicz włączanie, wyłączanie i przełączanie obu rodzajów numeracji. Zostaw numery bezwzględne włączone, względne wyłączone. Wymagane polecenia: nu · nonu · nu! · rnu · nornu · rnu!.

Początek: wiersz 1, kolumna 1

Cel kursora: wiersz 1, kolumna 1.

```
1     north
2     south
```

Numery bezwzględne: włączone

Numery względne: wyłączone

Przećwicz polecenia: nu, nonu, nu!, rnu, nornu, rnu!

Wskazówki: Włącz bezwzględne numery wierszy. Wyłącz bezwzględne numery wierszy.

Przykładowa droga

`:set<Space>nonu → <Enter> → :set<Space>nu → <Enter> → :set<Space>nu! → <Enter> → :set<Space>nu! → <Enter> → :set<Space>rnu → <Enter> → :set<Space>nornu → <Enter> → :set<Space>rnu! → <Enter> → :set<Space>rnu! → <Enter>`

13 / Warsztat Tkaczy

Nacinaj, dopisuj, zszywaj

Ada naprawia płótna utkane z liter. Czasem trzeba sięgnąć przed kursor, czasem ponad wers, a czasem rozpocząć cały ścieg od nowa.

Zadanie 37 / Przed nitką

Usuń x sprzed kursora, zmień DAWN na małe litery i odetnij junk. Zachowaj spację po dawn. Wymagane polecenia: X · ~ · D.

Początek: wiersz 1, kolumna 2

Stan początkowy

```
1     xDAWN junk
```

Docelowy tekst

```
1     dawn
```

Przećwicz polecenia: X, ~, D

Wskazówki: Usuń znak przed kursorem. Zmień wielkość litery pod kursorem.

Przykładowa droga

`X4~wD`

Zadanie 38 / Nacięcie i ścieg

Przez I dopisz the przed glow, zachowując wcięcie. Następnie przez O dodaj header w nowym wersie powyżej. Wymagane polecenia: I · O.

Początek: wiersz 1, kolumna 7

Stan początkowy

```
1        glow
```

Docelowy tekst

```
1     header
2        the glow
```

Przećwicz polecenia: I, O

Wskazówki: Wstawiaj przy pierwszym znaku wersu innym niż spacja. Otwórz nowy wers nad bieżącym i rozpocznij wstawianie.

Przykładowa droga

`Ithe<Space> → <Esc> → Oheader → <Esc>`

Zadanie 39 / Nowa warstwa

Zastąp pojedyncze x tekstem sun przez s, a cały drugi wers tekstem moon przez S. Wymagane polecenia: s · S.

Początek: wiersz 1, kolumna 1

Stan początkowy

```
1     x
2     bad line
```

Docelowy tekst

```
1     sun
2     moon
```

Przećwicz polecenia: s, S

Wskazówki: Usuń znak pod kursorem i rozpocznij wstawianie. Zastąp cały bieżący wers, przechodząc do wstawiania.

Przykładowa droga

`ssun → <Esc> → jSmoon → <Esc>`

14 / Odlewnia Gramatyki

Czynność i jej zasięg

Keris odlewa polecenia z dwóch części: co zrobić i na czym. Poznaj operator, dobierz ruch i odkryj, ile sensu mieści krótka składnia.

Zadanie 40 / Czasownik i droga

Porównaj zmianę słowa przez c, całego wersu przez cc i końcówki przez C. Otrzymaj warm sun oraz fixed. Wymagane polecenia: c · cc · C.

Początek: wiersz 1, kolumna 1

Stan początkowy

```
1  cold moss
2  broken
```

Docelowy tekst

```
1  warm sun
2  fixed
```

Przećwicz polecenia: c, cc, C

Wskazówki: Operator zmiany: usuń wskazany obszar i rozpocznij wstawianie. Zmień cały bieżący wers i rozpocznij wstawianie.

Przykładowa droga

`cewarm → <Esc> → jccfixed → <Esc> → ggwCsun → <Esc>`

Zadanie 41 / Zachowaj i wstaw

Skopiuj słowo operatorem y i wstaw je przed kursorem przez P. Wypróbuj też Y oraz usunięcie, zachowując north. Wymagane polecenia: y · Y · P · d.

Początek: wiersz 1, kolumna 1

Stan początkowy

```
1  red blue
2  north
```

Docelowy tekst

```
1  red red blue
2  north
```

Przećwicz polecenia: y, Y, P, d

Wskazówki: Operator kopiowania: połącz z ruchem lub obiektem tekstowym. Skopiuj cały bieżący wers; odpowiednik yy w tej grze.

Przykładowa droga

`ywwPYpdd`

Zadanie 42 / Jeszcze raz

Usuń pierwsze bad przez dw, a dwa następne przez powtórzenie zmiany. Pozostaw tylko light. Wymagane polecenia: dw · ..

Początek: wiersz 1, kolumna 1

Stan początkowy

```
1     bad bad bad light
```

Docelowy tekst

```
1     light
```

Przećwicz polecenia: dw, .

Wskazówki: Usuń od kursora do początku następnego słowa. Powtórz ostatnią zmianę tekstu, łącznie z wpisaną treścią.

Przykładowa droga

`dw..`

15 / Sanktuarium Słów

Wybieraj znaczenie, nie odległość

Eli nie mierzy tekstu liczbą znaków. W sanktuarium słowo, zdanie i akapit stają się gotowymi naczyniami dla Twojej intencji.

Zadanie 43 / Serce słowa

Porównaj iw i aw: usuń wnętrze moss, a potem całe blue z przyległą przestrzenią. Pozostaw red. Wymagane polecenia: iw · aw.

Początek: wiersz 1, kolumna 6

Stan początkowy

```
1     red moss blue
```

Docelowy tekst

```
1     red
```

Przećwicz polecenia: iw, aw

Wskazówki: Wnętrze bieżącego słowa bez otaczających spacji; użyj z d, y lub c. Całe słowo wraz z przyległą spacją; obiekt dla operatora.

Przykładowa droga

`diwwdaw`

Zadanie 44 / Duże słowo

Potraktuj moss.green jako jeden WORD. Przećwicz iW oraz aW i pozostaw tylko red. Wymagane polecenia: iW · aW.

Początek: wiersz 1, kolumna 7

Stan początkowy

```
1     red moss.green blue
```

Docelowy tekst

```
1     red
```

Przećwicz polecenia: iW, aW

Wskazówki: Wnętrze WORD: ciąg bez białych znaków, także z interpunkcją. WORD wraz z otaczającą białą przestrzenią.

Przykładowa droga

`diWwdaW`

Zadanie 45 / Zdania i akapity

Skopiuj wnętrze i całość zdania, potem wnętrze akapitu. Usuń akapit alpha/beta wraz z odstępem, zachowując zdania i keep. Wymagane polecenia: is · as · ip · ap.

Początek: wiersz 1, kolumna 1

Stan początkowy

```
1     One. Two.
2     [pusty wiersz]
3     alpha
4     beta
5     [pusty wiersz]
6     keep
```

Docelowy tekst

```
1     One. Two.
2     [pusty wiersz]
3     keep
```

Przećwicz polecenia: is, as, ip, ap

Wskazówki: Wnętrze zdania, bez oddzielającej je białej przestrzeni. Zdanie wraz z oddzielającą białą przestrzenią.

Przykładowa droga

```
yisyasjjyipdap
```

16 / Skarbiec Ograniczników

Wnętrze i otaczająca je rama

Sen pilnuje szkatułek, które mieszczą się jedna w drugiej. Wybierz samą zawartość albo całą ramę, a różne nazwy kluczy otworzą te same zamki.

Zadanie 46 / Kwadrat i okrąg

Skopiuj zawartość i ramy [] oraz () każdym podanym aliasem. Na końcu rejestr nienazwany ma zawierać całe (glow). Wymagane polecenia: i[· a[· i] · a] · i(· a(· i) · a) · ib · ab.

Początek: wiersz 1, kolumna 3

Cel kursora: wiersz 1, kolumna 8.

```
1     [moss] (glow)
```

Rejestr ""

```
1     (glow)
```

Przećwicz polecenia: i[, a[, i], a], i(, a(, i), a), ib, ab

Wskazówki: Wnętrze pasującej pary [] bez nawiasów; oba aliasy znaczą to samo. Cała para [] razem z nawiasami.

Przykładowa droga

`yi[ya[yi]ya]f(yi(ya(yi)ya)yibyab`

Zadanie 47 / Kąty i aliasy

Porównaj wszystkie aliasy {} i <>. Na końcu zachowaj w rejestrze całe <dawn>, razem z ramą. Wymagane polecenia: i{ · a{ · i} · a} · iB · aB · i< · a< · i> · a>.

Początek: wiersz 1, kolumna 3

Cel kursora: wiersz 1, kolumna 9.

```
1     {ember} <dawn>
```

Rejestr ""

```
1     <dawn>
```

Przećwicz polecenia: i{, a{, i}, a}, iB, aB, i<, a<, i>, a>

Wskazówki: Wnętrze pasującej pary {}, bez klamer; iB jest aliasem. Cała para {} razem z klamrami; aB jest aliasem.

Przykładowa droga

`yi{ya{yi}ya}yiByaBf<yi<ya<yi>ya>`

Zadanie 48 / Cała rama

Zmień wnętrze (seed) na bloom bez usuwania nawiasów, a następnie usuń całe [noise]. Wymagane polecenia: i(· a[.

Początek: wiersz 1, kolumna 11

Stan początkowy

```
1     [noise] (seed) {ash} <sky>
```

Docelowy tekst

```
1      (bloom) {ash} <sky>
```

Przećwicz polecenia: i(, a[

Wskazówki: Wnętrze pasującej pary (), bez nawiasów; ib jest aliasem. Cała para [] razem z nawiasami.

Przykładowa droga

```
ci(bloom → <Esc> → 0da[
```

17 / Kaplica Szeptów

Posłuchaj tego, co między znakami

Yuna przechowuje wiadomości w cudzysłowach i znacznikach. Aby usłyszeć treść, naucz się odróżniać głos od jego oprawy.

Zadanie 49 / Dwa ciche głosy

Porównaj wnętrze i całość obu rodzajów cudzysłowów. Na końcu skopiuj całe pojedynczo cytowane glow. Wymagane polecenia: i" · a" · i' · a'.

Początek: wiersz 1, kolumna 3

Cel kursora: wiersz 1, kolumna 8.

```
1     "moss" 'glow'
```

Rejestr ""

```
1     'glow'
```

Przećwicz polecenia: i", a", i', a'

Wskazówki: Tekst wewnątrz cudzysłowu podwójnego, bez znaków cudzysłowu. Tekst razem z otaczającymi cudzysłowami podwójnymi.

Przykładowa droga

`yi"ya"f'yi'ya'`

Zadanie 50 / Szept i znacznik

Skopiuj wnętrze i całość zapisu w znakach `, a potem elementu HTML. Zachowaj całe <b>glow</b>. Wymagane polecenia: i` · a` · it · at.

Początek: wiersz 1, kolumna 3

Cel kursora: wiersz 1, kolumna 8.

```
1     `seed` <b>glow</b>
```

Rejestr ""

```
1     <b>glow</b>
```

Przećwicz polecenia: i`, a`, it, at

Wskazówki: Tekst wewnątrz znaków `, bez tych ograniczników. Tekst wraz z otaczającymi znakami `.

Przykładowa droga

`yi`ya`f<yityat`

Zadanie 51 / Zagnieżdżona wiadomość

Zmień tylko treść elementu p na new, pozostawiając znaczniki. Usuń całe cytowane noise i zachowaj końcową spację. Wymagane polecenia: it · a".

Początek: wiersz 1, kolumna 5

Stan początkowy

```
1      <p>old</p> "noise"
```

Docelowy tekst

```
1      <p>new</p>
```

Przećwicz polecenia: it, a"

Wskazówki: Wnętrze pasującej pary znaczników HTML, bez samych znaczników. Tekst razem z otaczającymi cudzysłowami podwójnymi.

Przykładowa droga

`citnew → <Esc> → f"da"`

18 / Obóz Kartografów

Zostaw punkt, do którego wrócisz

Aro rysuje mapy, których nie rozwieje wiatr. Zapamiętaj miejsce, wróć do wersu lub dokładnej kolumny i usuń znaki, które już nie prowadzą dalej.

Zadanie 52 / Postaw drogowskaz

Ustaw znacznik a w drugim wersie. Porównaj skok do wersu z powrotem do dokładnej kolumny i wróć na zapisane miejsce. Wymagane polecenia: m · ' · `.

Początek: wiersz 2, kolumna 5

Cel kursora: wiersz 2, kolumna 5.

```
1       north
2       south
3       dawn
```

Znacznik a: wiersz 2, kolumna 5

Przećwicz polecenia: m, ', `

Wskazówki: Ustaw znacznik: m i litera, np. ma, zapamiętują bieżącą pozycję. Apostrof i litera znacznika prowadzą do pierwszego znaku jego wersu.

Przykładowa droga

`magg'agg`a`

Zadanie 53 / Odczytaj mapę

Ustaw a i b w sąsiednich kolumnach, pokaż listę znaczników i usuń tylko a skrótem :delm. Wymagane polecenia: :marks · :delm.

Początek: wiersz 1, kolumna 1

Cel kursora: wiersz 1, kolumna 2.

```
1     path
```

Znacznik b: wiersz 1, kolumna 2

Usunięte znaczniki: a

Otwarta lista: Znaczniki

Przećwicz polecenia: :marks, :delm

Wskazówki: Pokaż zapamiętane znaczniki. Usuń wskazane znaczniki, np. :delmarks a; :delm jest skrótem.

Przykładowa droga

`malmb → :marks → <Enter> → :delm<Space>a → <Enter>`

Zadanie 54 / Wymaż dawny szlak

Przećwicz usuwanie zakresu znaczników, pełne :delmarks! i skrót :delm!. Na końcu usuń a, b i c. Wymagane polecenia: :delmarks · :delmarks! · :delm!.

Początek: wiersz 1, kolumna 1

Cel kursora: wiersz 1, kolumna 2.

```
1     path
```

Usunięte znaczniki: a, b, c

Przećwicz polecenia: :delmarks, :delmarks!, :delm!

Wskazówki: Usuń wskazane znaczniki, np. :delmarks a; :delm jest skrótem. Usuń wszystkie lokalne znaczniki.

Przykładowa droga

`malmb → :delmarks<Space>a-b → <Enter> → ma → :delmarks! → <Enter> → mc → :delm! → <Enter>`

19 / Gabinet Ech

Schowek może mieć imię

Esme przechowuje echa w osobnych szufladkach. Jedno zachowasz, do drugiego dopiszesz, a niepotrzebny szmer wypuścisz tak, by niczego nie zagłuszył.

Zadanie 55 / Echo bez imienia

Usuń a, odzyskaj je z rejestru małego usunięcia i skopiuj całe ab przez jawny rejestr nienazwany. Wymagane polecenia: "- · "" · x · yy · P.

Początek: wiersz 1, kolumna 1

Docelowy tekst

```
1      ab
```

Rejestr ""

```
1      ab
```

Rejestr "-

```
1      a
```

Przećwicz polecenia: "-, "", x, yy, P

Wskazówki: Rejestr małego usunięcia: fragment krótszy niż cały wers. Rejestr nienazwany przechowuje ostatnio kopiowany lub usuwany tekst.

Przykładowa droga

`x"-P""yy`

Zadanie 56 / Więcej bez nadpisania

Zachowaj moss w a, dopisz glow przez A, a następnie wklej oba wersy. Sprawdź treść rejestru a. Wymagane polecenia: "a · "A · p.

Początek: wiersz 1, kolumna 1

Stan początkowy

```
1     moss
2     glow
```

Docelowy tekst

```
1     moss
2     moss
3     glow
4     glow
```

Rejestr "a

```
1     moss
2     glow
```

Przećwicz polecenia: "a, "A, p

Wskazówki: Rejestr nazwany a-z: zachowaj tekst przez wskazanie litery przed operatorem. Wielka litera A-Z dopisuje do odpowiedniego rejestru a-z zamiast go zastępować.

Przykładowa droga

`"ayyj"Ayygg"ap`

Zadanie 57 / Cisza chroni kopię

Zachowaj keep w a, usuń trash do czarnej dziury i wstaw zachowaną kopię przed bieżącym wersem. Wymagane polecenia: "_ · "a · dd · P.

Początek: wiersz 1, kolumna 1

Stan początkowy

```
1  keep
2  trash
```

Docelowy tekst

```
1  keep
2  keep
```

Rejestr "a

```
1  keep
```

Rejestr ""

```
1  keep
```

Przećwicz polecenia: "_, "a, dd, P

Wskazówki: Czarna dziura: usuń bez nadpisywania innych rejestrów. Rejestr nazwany a-z: zachowaj tekst przez wskazanie litery przed operatorem.

Przykładowa droga

`"ayyj"_dd"aP`

20 / Komnata Dziewięciu

Pamięć ma swoją kolejność

Nona pamięta nie tylko to, co zniknęło ostatnio. Otwórz kolejne szuflady historii i przekonaj się, że usunięcie nie musi być pożegnaniem.

Zadanie 58 / Ostatnia kopia

Skopiuj light, usuń dust, a potem odzyskaj kopię z rejestru 0. Rejestr 1 ma nadal pamiętać dust. Wymagane polecenia: "0 · dd.

Początek: wiersz 1, kolumna 1

Stan początkowy

```
1     light
2     dust
```

Docelowy tekst

```
1     light
2     light
```

Rejestr "0

```
1     light
```

Rejestr "1

```
1     dust
```

Przećwicz polecenia: "0, dd

Wskazówki: Rejestr 0 zachowuje ostatnie kopiowanie, także po późniejszym usunięciu. Wytnij bieżący wers i zachowaj go w rejestrze.

Przykładowa droga

```
yyjdd"0p
```

Zadanie 59 / Łańcuch pożegnań

Usuń dziewięć wersów i sprawdź rejestry historii od 1 do 9. Przywróć najstarszy one nad keep. Wymagane polecenia: "1 · "2 · "3 · "4 · "5 · "6 · "7 · "8 · "9.

Początek: wiersz 1, kolumna 1

Stan początkowy

```
1     one
2     two
3     three
4     four
5     five
6     six
7     seven
8     eight
9     nine
10    keep
```

Docelowy tekst

```
1     one
2     keep
```

Rejestr "1

```
1     nine
```

Rejestr "2

```
1     eight
```

Rejestr "3

```
1     seven
```

Rejestr "4

```
1     six
```

Rejestr "5

```
1     five
```

Rejestr "6

```
1     four
```

Rejestr "7

```
1     three
```

Rejestr "8

```
1     two
```

Rejestr "9

```
1     one
```

Przećwicz polecenia: "1, "2, "3, "4, "5, "6, "7, "8, "9

Wskazówki: Historia większych usunięć: 1 to najnowsze, 9 najstarsze z zachowanych. Historia większych usunięć: 1 to najnowsze, 9 najstarsze z zachowanych.

Przykładowa droga

`dd........ → "1pu → "2pu → "3pu → "4pu → "5pu → "6pu → "7pu → "8pu → "9P`

Zadanie 60 / Odczytaj szuflady

Skopiuj one do a, a następnie otwórz podgląd rejestrów. Sprawdź, że a oraz 0 pamiętają ten sam wers. Wymagane polecenia: :reg.

Początek: wiersz 1, kolumna 1

Cel kursora: wiersz 1, kolumna 1.

```
1      one
2      two
```

Rejestr "0

```
1      one
```

Rejestr "a

```
1      one
```

Otwarta lista: Rejestry

Przećwicz polecenia: :reg

Wskazówki: Pokaż rejestry; można wskazać ich nazwy, np. :reg a0. Najpierw wykonaj jeden świadomy krok i sprawdź kursor lub tekst. Powtórzenia dodawaj dopiero, gdy rozumiesz pojedynczą operację.

Przykładowa droga

`"ayy → :reg → <Enter>`

21 / Port Skrybów

Wiele stron, jedna podróż

Tero przewozi dokumenty między wirtualnymi przystaniami. Zapisuj to, co chcesz zachować, i przełączaj otwarte strony bez gubienia ich nazw.

Zadanie 61 / Zapamiętana strona

Zapisz dawn rises w notes.txt, popraw echo.txt na awake i zapisz go. Wróć do notes.txt bez utraty zmian. Wymagane polecenia: :w · :e · c.

Początek: wiersz 1, kolumna 1

Stan początkowy

```
1  dawn
```

Docelowy tekst

```
1  dawn rises
```

Aktywny bufor: notes.txt

Zapisany wirtualny plik: notes.txt

```
1  dawn rises
```

Zapisany wirtualny plik: echo.txt

```
1  awake
```

Przećwicz polecenia: :w, :e, c

Wskazówki: Zapisz wirtualny dokument; opcjonalna nazwa wskazuje plik w grze. Otwórz wirtualny plik pod wskazaną nazwą.

Przykładowa droga

```
A<Space>rises → <Esc> → :w → <Enter> → :e<Space>echo.txt → <Enter> → ccawake → <Esc> → :w → <Enter> → :e<Space>notes.txt → <Enter>
```

Zadanie 62 / Inna przystań

Otwórz beta.txt, pokaż listę buforów i wróć po nazwie do alpha.txt. Wymagane polecenia: :e · :ls · :b.

Początek: wiersz 1, kolumna 1

Cel kursora: wiersz 1, kolumna 1.

```
1  north
```

Aktywny bufor: alpha.txt

Otwarta lista: Bufory

Przećwicz polecenia: :e, :ls, :b

Wskazówki: Otwórz wirtualny plik pod wskazaną nazwą. Pokaż otwarte bufory i ich numery.

Przykładowa droga

```
:e<Space>beta.txt → <Enter> → :ls → <Enter> → :b<Space>alpha.txt → <Enter>
```

Zadanie 63 / Przeprawa między buforami

Zapisz kopię jako final.txt, otwórz ją i porównaj przełączanie buforów numerami. Zakończ w final.txt. Wymagane polecenia: :w · :e · :b.

Początek: wiersz 1, kolumna 1

Docelowy tekst

```
1     light
```

Aktywny bufor: final.txt

Zapisany wirtualny plik: final.txt

```
1     light
```

Przećwicz polecenia: :w, :e, :b

Wskazówki: Zapisz wirtualny dokument; opcjonalna nazwa wskazuje plik w grze. Otwórz wirtualny plik pod wskazaną nazwą.

Przykładowa droga

```
:w<Space>final.txt → <Enter> → :e<Space>final.txt → <Enter> → :b<Space>1 → <Enter> → :b<Space>2 → <Enter>
```

22 / Mechaniczny Sad

Niech dobry ruch wyda wiele owoców

Omi nastawia delikatne mechanizmy w sadzie. Nagraj jeden niezawodny cykl, sprawdź jego koniec i pozwól mu pracować przy kolejnych gałęziach.

Zadanie 64 / Mała maszyna

Nagraj w a dopisanie ! i przejście do kolejnego wersu. Odtwórz raz, aby zmienić dwa pierwsze dawn. Wymagane polecenia: q · @.

Początek: wiersz 1, kolumna 1

Stan początkowy

```
1	dawn
2	dawn
3	dawn
```

Docelowy tekst

```
1	dawn!
2	dawn!
3	dawn
```

Nagrane makra: a

Przećwicz polecenia: q, @

Wskazówki: q i litera rozpoczynają nagrywanie makra; kolejne q je kończy. Odtwórz makro z wybranego rejestru, np. @a; liczba daje powtórzenia.

Przykładowa droga

`qaA! → <Esc> → jq@a`

Zadanie 65 / Sad się powtarza

Nagraj dodanie + na początku wersu i przejście dalej. Użyj @a i @@, zostawiając czwarty seed bez zmian. Wymagane polecenia: q · @ · @@.

Początek: wiersz 1, kolumna 1

Stan początkowy

```
1  seed
2  seed
3  seed
4  seed
```

Docelowy tekst

```
1  +seed
2  +seed
3  +seed
4  seed
```

Nagrane makra: a

Przećwicz polecenia: q, @, @@

Wskazówki: q i litera rozpoczynają nagrywanie makra; kolejne q je kończy. Odtwórz makro z wybranego rejestru, np. @a; liczba daje powtórzenia.

Przykładowa droga

`qaI+ → <Esc> → jq@a@@`

Zadanie 66 / Jeszcze raz, dokładnie

Nagraj usunięcie jednego x, a potem odtwórz makro trzy razy. Zostaw samo light. Wymagane polecenia: q · @ · licznik.

Początek: wiersz 1, kolumna 1

Stan początkowy

```
1  xxxxlight
```

Docelowy tekst

```
1  light
```

Nagrane makra: a

Przećwicz polecenia: q, @, count

Wskazówki: q i litera rozpoczynają nagrywanie makra; kolejne q je kończy. Odtwórz makro z wybranego rejestru, np. @a; liczba daje powtórzenia.

Przykładowa droga

`qaxq3@a`

23 / Próby Tkaczy

Składaj poznane wzory

Vel rozciąga przed Tobą długi wzór. Liczniki, powtórzenie zmiany i makro są różnymi narzędziami: wybierz takie, które wyraża zamierzoną pracę.

Zadanie 67 / Dwa liczniki

Połącz licznik operatora i ruchu: usuń sześć słów jednym 2d3w i zachowaj keep. Wymagane polecenia: licznik · dw.

Początek: wiersz 1, kolumna 1

Stan początkowy

```
1      one two three four five six keep
```

Docelowy tekst

```
1      keep
```

Przećwicz polecenia: count, dw

Wskazówki: Liczba przed operatorem lub ruchem określa powtórzenia; 2d3w łączy oba liczniki. Usuń od kursora do początku następnego słowa.

Przykładowa droga

`2d3w`

Zadanie 68 / Ten sam splot

Usuń x, cofnij i ponów zmianę poleceniami :undo oraz :redo, a potem powtórz usunięcie kropką. Wymagane polecenia: :undo · :redo · ..

Początek: wiersz 1, kolumna 1

Stan początkowy

```
1      xxdawn
```

Docelowy tekst

```
1      dawn
```

Przećwicz polecenia: :undo, :redo, .

Wskazówki: Cofnij ostatnią zmianę; wariant wiersza poleceń klawisza u. Ponów cofniętą zmianę; wariant wiersza poleceń Ctrl-r.

Przykładowa droga

`x → :undo → <Enter> → :redo → <Enter> → .`

Zadanie 69 / Niezawodny wzór

Nagraj usunięcie ash i przejście w dół, powtórz cykl dwa razy, sprawdź rejestr 0 na keep i cofnij próbną kopię. Wymagane polecenia: q · @ · licznik · "0.

Początek: wiersz 1, kolumna 1

Stan początkowy

```
1  ash dawn
2  ash dawn
3  ash dawn
4  keep
```

Docelowy tekst

```
1  dawn
2  dawn
3  dawn
4  keep
```

Rejestr "0

```
1  keep
```

Nagrane makra: a

Przećwicz polecenia: q, @, count, "0

Wskazówki: q i litera rozpoczynają nagrywanie makra; kolejne q je kończy. Odtwórz makro z wybranego rejestru, np. @a; liczba daje powtórzenia.

Przykładowa droga

`qadwjq2@ayy"0Pu`

24 / Szczyt Początków

Cały archipelag w jednym zapisie

Luma czeka tam, gdzie spotykają się wszystkie ścieżki. Nie przywołuje nowych mocy: prosi, by połączyć pamięć, precyzję i wyobraźnię w ostatnią pracę.

Zadanie 70 / Zapis bez szumu

Zmień wnętrze znacznika na light, usuń całe [noise] oraz pusty wers, a potem zapisz dokument. Wymagane polecenia: it · a[· dd · :w.

Początek: wiersz 1, kolumna 5

Stan początkowy

```
1     <p>old</p>
2     [noise]
3     keep
```

Docelowy tekst

```
1     <p>light</p>
2     keep
```

Przećwicz polecenia: it, a[, dd, :w

Wskazówki: Wnętrze pasującej pary znaczników HTML, bez samych znaczników. Cała para [] razem z nawiasami.

Przykładowa droga

`citlight → <Esc> → jda[dd → :w → <Enter>`

Zadanie 71 / Dokładna droga powrotna

Zapamiętaj dwa miejsca znacznikami A i a. Połącz dokładne skoki z wyszukiwaniem w obu kierunkach i wróć do dawn. Wymagane polecenia: m · ` · / · ? · n.

Początek: wiersz 1, kolumna 6

Cel kursora: wiersz 3, kolumna 6.

```
1     star north
2     star south
3     star dawn
```

Znacznik A: wiersz 1, kolumna 6

Znacznik a: wiersz 3, kolumna 6

Przećwicz polecenia: m, `, /, ?, n

Wskazówki: Ustaw znacznik: m i litera, np. ma, zapamiętują bieżącą pozycję. Znak ` i litera znacznika prowadzą do dokładnie zapamiętanej kolumny.

Przykładowa droga

`mA/dawn → <Enter> → ma`A?star → <Enter> → n`a`

Zadanie 72 / Wszystkie światła, jedno niebo

Nagraj zmianę sleep na wake i wykonaj ją w trzech wersach. Pobierz ostatni wers z message.txt, wstaw do dawn.txt i zapisz. Wymagane polecenia: q · @ · @@ · :w · :e · yy · "0 · p.

Początek: wiersz 1, kolumna 1

Stan początkowy

```
1     sleep
2     sleep
3     sleep
```

Docelowy tekst

```
1     wake
2     wake
3     wake
4     the dawn returns
```

Rejestr "0

```
1     the dawn returns
```

Aktywny bufor: dawn.txt

Zapisany wirtualny plik: dawn.txt

```
1     wake
2     wake
3     wake
4     the dawn returns
```

Zapisany wirtualny plik: message.txt

```
1     the dawn returns
```

Nagrane makra: a

Przećwicz polecenia: q, @, @@, :w, :e, yy, "0, p

Wskazówki: q i litera rozpoczynają nagrywanie makra; kolejne q je kończy. Odtwórz makro z wybranego rejestru, np. @a; liczba daje powtórzenia.

Przykładowa droga

`qaccwake → <Esc> → jq@a@@ → :w → <Enter> → :e<Space>message.txt → <Enter> → yy → :e<Space>dawn.txt → <Enter> → "0p → :w → <Enter>`

Dalsza nauka

Podręcznik opisuje edukacyjny podzbiór obsługiwany przez grę. Wirtualne pliki pozostają w przeglądarce; gra nie udostępnia systemu plików urządzenia.

Vim documentation

Motions and text objects
https://vimhelp.org/motion.txt.html

Editing and registers
https://vimhelp.org/change.txt.html

Macros
https://vimhelp.org/usr_10.txt.html

Files and buffers
https://vimhelp.org/usr_22.txt.html

LUMEN

The light keeper’s handbook

24 realms, 72 quests. A practical guide to Vim movements and its language of editing.

24 islands / 72 missions / entirely free

English / Edition 0.3.0

This handbook describes the educational subset supported by the game. Virtual files stay in the browser; the game does not expose your device file system.

Try it yourself first. Solutions are examples; other correct sequences may also satisfy the goal.

Contents

Before your journey 3

Command atlas 5

The island guide 10

01 / Lantern Harbor . . . 11
02 / Word Grove . . . 13
03 / Bridges of Lines . . . 14
04 / Echo Observatory . . . 16
05 / Ember Forge . . . 17
06 / Mirror Library . . . 19
07 / Dawn Scriptorium . . . 21
08 / Morning Tower . . . 23
09 / Wind Cliffs . . . 25
10 / Hunters' Pass . . . 26
11 / Stone Circles . . . 27
12 / Horizon Tower . . . 29
13 / Weavers' Workshop . . . 31
14 / Grammar Foundry . . . 33
15 / Word Sanctuary . . . 35
16 / Delimiter Vault . . . 37
17 / Whisper Chapel . . . 39
18 / Cartographers' Camp . . . 41
19 / Cabinet of Echoes . . . 43
20 / Chamber of Nine . . . 46
21 / Scribes' Harbor . . . 49
22 / Clockwork Orchard . . . 51
23 / Weavers' Trials . . . 53
24 / Summit of Beginnings . . . 55

Keep learning 57

Before your journey

Your goal

The journey teaches deliberate text editing, from a single movement to text objects, registers, and macros. Each realm introduces a small family of commands. Quests require the specified tools, not just an accidental match with the target text.

Completing the game means practising its curriculum, not mastering every feature of the full Vim editor. Revisit completed quests and look for your own clear, repeatable solutions.

How to play

Choose a language, name your character, and accept the terms. Click the board to give it keyboard focus. In Normal mode, letters perform commands; in Insert mode, they become text. Esc returns to Normal mode. On touch devices, use the keys beneath the board and the text entry fields.

Before moving, read the objective, compare the buffer with the target, and check the required commands. Hints become more specific in stages. Restart affects the current quest; the map lets you return to unlocked realms. The Help button opens explanations and the manual at any time. Esc in Insert mode returns to Normal; in Normal mode, when no command is pending, it opens pause. A pause button is also available. The final three quests can be played as optional timed trials. Their suggested times are an extra challenge, not a condition for completing the curriculum. Practise correctness before speed.

Saving and privacy

Your name, terms acceptance, and progress are stored in this browser's localStorage. This is not an account or a cloud backup. Private browsing, clearing site data, or storage restrictions can remove it. Download a save backup if you want to preserve it or move to another device.

Optional diagnostics are off by default and have a separate switch in settings. They are not required to play. Diagnostics contain limited technical error data, not your name, exercise contents, or full save. The terms explain the data handling in more detail.

Reading the keys

Case matters: w and W, p and P, i and I are different commands. 3w means press 3 and then w; Ctrl-r means hold Control and press r. Symbols including +, -, |, $, %, semicolon, and comma are actual command keys.

A command such as ciw combines operator c with text object iw. Enter the names and target text shown on the tablets exactly, even when the instructions use another language. Press Enter after : commands and after searches beginning with / or ?. Type sequences in order without the arrows that separate steps. <Esc>, <Enter>, <C-r>, and <Space> mean Escape, Enter, Control+r, and the space key; do not type the angle brackets.

The game and real Vim

LUMEN is an educational simulator of a selected Vim curriculum. The board represents a text buffer and its visible part is a viewport. Scrolling commands change the viewport, not the text. Files are virtual: the game does not read your computer's directories or execute operating-system commands.

Commands follow the editing operations being taught, but the game does not replace full Vim with all its options, plugins, and syntax exceptions. In-game help and this handbook describe the supported scope.

The command line

A colon opens the command line. :w saves a virtual document, :e opens another, :ls lists buffers, and :b switches buffers. :marks lists marks; :delmarks and its abbreviation :delm remove them, while :delmarks! clears local marks. :reg shows register contents.

:set nu and :set nonu enable and disable absolute line numbers; :set nu! toggles them. :set rnu, :set nornu, and :set rnu! do the same for relative numbers. Enter confirms a command; Esc cancels it.

Registers

A register is Vim's clipboard. Ordinary copying and deleting use the unnamed register; a double-quote prefix selects another. Lowercase a-z identify named registers, while writing to A-Z appends to the corresponding lowercase register.

Register 0 preserves the last yank, registers 1-9 retain a history of larger deletions, and register - stores a small deletion. Register _ is the black hole: delete into it without replacing other clipboard contents. :reg helps you inspect what was actually stored.

Macros

A macro records a sequence of actions, including cursor movements. To record in register a, press qa, perform the actions, and stop recording with q. @a replays register a; @@ repeats the last replayed macro. A count before replay sets the number of repetitions.

Test a single run before replaying many times. A useful macro ends in a predictable position, ready to process the next piece of text.

Virtual files and buffers

A buffer is an open document; a filename identifies its virtual saved copy. :w [name] saves text, :e name opens a document, :ls lists buffers, and :b name or number switches buffers. Save intentional changes before leaving a document.

Everything happens inside the game and its local save. Exporting your journey creates a backup, not a real source file in your computer's filesystem. Do not type passwords or other confidential information into exercises.

<Esc> = Escape; <Enter> = Enter; <C-r> = Ctrl+r; <Space> = space. An arrow separates fragments and is not a key. Letter case matters.

Command atlas

Command	What it does
`0`	Move to the first column of the current line.
`hjkl`	h left, j down, k up, l right: the basic cursor movements.
`h`	Move one tile or character left.
`j`	Move one line or tile down.
`k`	Move one line or tile up.
`l`	Move one tile or character right.
`w`	Move to the beginning of the next word.
`b`	Move backward to a word's beginning.
`e`	Move to the end of a word.
`^`	Move to the first non-blank character of the line.
`$`	Move to the last character of the current line.
`gg`	Move to the first line.
`G`	Move to the last line.
`count`	A number repeats the next movement, such as 3w or 2j.
`f`	Press f and a character to find its next occurrence on this line.
`;`	Repeat the last character find made with f.
`/`	Search forward for text and confirm with Enter.
`n`	Repeat the search in its existing direction.
`N`	Repeat opposite to the last / or ? search direction.
`x`	Delete the character under the cursor.
`r`	Replace the character under the cursor with the next character typed.
`dd`	Cut the current line and keep it in a register.
`dw`	Delete from the cursor toward the next word's beginning.
`u`	Undo the last text change.
`Ctrl-r`	Redo an undone change.
`yy`	Yank the whole current line into the register.

Command	What it does
`yw`	Copy text toward the next word’s beginning.
`p`	Put text after the cursor, or complete lines below the current line.
`i`	Begin inserting before the cursor.
`a`	Begin inserting after the cursor.
`A`	Begin inserting at the end of the line.
`o`	Open a new line below and begin inserting.
`Esc`	Return to Normal mode or cancel a pending command.
`:w`	Save the virtual document; an optional name selects an in-game file.
`1 ... 9`	A count before an operator or motion sets repetitions; 2d3w combines both counts.
`W`	Move to the next whitespace-delimited WORD; punctuation stays part of it.
`E`	Move to the end of a whitespace-delimited WORD.
`B`	Move backward to a WORD’s beginning; punctuation does not split it.
`ge`	Move backward to the end of the previous word.
`gE`	Move backward to the end of the previous WORD.
`+`	Move to the first non-blank of the next line; a count repeats it.
`-`	Move to the first non-blank of the previous line.
`|`	Go to the numbered column, such as 5|; without a count, column 1.
`F`	Press F and a character to find it backward on the current line.
`t`	Move just before the next occurrence of a character on this line.
`T`	Find a character backward, stopping just after it.
`,`	Repeat the last f, F, t, or T in the opposite direction.
`*`	Search forward for the whole word under the cursor.
`#`	Search backward for the whole word under the cursor.
`?`	Start a backward text search; confirm with Enter.
`%`	Jump to the matching (), [], or {} delimiter.
`[{`	Move backward to the enclosing unmatched opening brace.
`]}`	Move forward to the enclosing unmatched closing brace.
`[(`	Move backward to an enclosing unmatched opening parenthesis.

Command	What it does
`])`	Move forward to an enclosing unmatched closing parenthesis.
`{`	Move backward to a paragraph boundary; blank lines separate paragraphs.
`}`	Move forward to the next paragraph boundary.
`(`	Move backward to a sentence beginning.
`)`	Move forward to the next sentence beginning.
`zt`	Scroll the current line to the top of the viewport.
`zz`	Center the current line in the viewport without changing text.
`zb`	Scroll the current line to the bottom of the viewport.
`H`	Move to the first visible line of the viewport.
`M`	Move to the middle visible line of the viewport.
`L`	Move to the last visible line of the viewport.
`nu / :set nu`	Enable absolute line numbers.
`nonu / :set nonu`	Disable absolute line numbers.
`nu! / :set nu!`	Toggle absolute line numbers.
`rnu / :set rnu`	Enable relative line numbers showing distance from the cursor.
`nornu / :set nornu`	Disable relative line numbers.
`rnu! / :set rnu!`	Toggle relative line numbers.
`X`	Delete the character before the cursor.
`D`	Delete from the cursor to the end of the line.
`~`	Toggle the case of the character under the cursor.
`I`	Begin inserting at the first non-blank of the line.
`O`	Open a line above the current one and enter Insert mode.
`s`	Delete the character under the cursor and begin inserting.
`S`	Replace the current line by entering Insert mode.
`d`	Delete operator: add a motion or object, such as d$, diw, or d2w.
`y`	Yank operator: combine it with a motion or text object.
`c`	Change operator: remove the selected range and begin inserting.
`cc`	Change the whole current line and begin inserting.

Command	What it does
`C`	Change from the cursor to line end and begin inserting.
`Y`	Yank the whole current line; equivalent to yy in this game.
`.`	Repeat the last text change, including inserted text.
`iw`	The inner word without surrounding spaces; combine with d, y, or c.
`aw`	A word including adjacent whitespace; an operator's text object.
`iW`	The inner WORD: non-whitespace text, including punctuation.
`aW`	A WORD including surrounding whitespace.
`is`	The inner sentence without its separating whitespace.
`as`	A sentence including separating whitespace.
`ip`	The inner paragraph without surrounding blank lines.
`ap`	A paragraph including its separating blank lines.
`i[ / i]`	Inside a matching [] pair, excluding brackets; both aliases are equivalent.
`a[ / a]`	A complete [] pair, including brackets.
`i( / i) / ib`	Inside a matching () pair, excluding parentheses; ib is an alias.
`a( / a) / ab`	A complete () pair, including parentheses; ab is an alias.
`i{ / i} / iB`	Inside a matching {} pair, excluding braces; iB is an alias.
`a{ / a} / aB`	A complete {} pair, including braces; aB is an alias.
`i< / i>`	Inside a matching <> pair, excluding angle brackets.
`a< / a>`	A complete <> pair, including angle brackets.
`i"`	Text inside double quotes, excluding the quotes.
`a"`	Text including its surrounding double quotes.
`i'`	Text inside single quotes, excluding delimiters.
`a'`	Text including surrounding single quotes.
`` i` ``	Text inside backticks, excluding the delimiters.
`` a` ``	Text including surrounding backticks.
`it`	The inside of a matching HTML tag pair, excluding the tags.
`at`	A whole element, including its opening and closing tags.
`m`	Set a mark: m and a letter, such as ma, remember the current position.

Command	What it does
'	A quote and mark letter jump to the first non-blank of its line.
`	A backtick and mark letter jump to the exact saved position.
:marks	List saved marks.
:delmarks / :delm	Remove specified marks, such as :delmarks a; :delm is an abbreviation.
:delmarks!	Remove all local marks.
"	Register prefix: double quote, register name, then an operator or p.
""	The unnamed register holds the latest yanked or deleted text.
"-	Small-delete register: deleted text smaller than a whole line.
"_	Black hole: delete without overwriting other registers.
"0	Register 0 keeps the latest yank, even after a subsequent deletion.
:reg / :registers	Show registers; optionally specify names, such as :reg a0.
:e	Open a virtual file with the given name.
:ls	List open buffers and their numbers.
:b	Switch virtual buffers by name or number.
q	q and a letter start recording a macro; another q stops it.
@	Replay a register's macro, such as @a; a count repeats it.
@@	Replay the last used macro again.
"1 ... "9	Larger-deletion history: 1 is newest, 9 the oldest retained.
"a ... "z	Named a-z register: select its letter before an operator to keep text.
"A ... "Z	Uppercase A-Z appends to the corresponding a-z register instead of replacing it.
P	Put text before the cursor, or a whole line above the current line.
:set	Change display options, such as :set nu or :set rnu!.
:	Open the command line; type a command and confirm with Enter.
z	Scroll prefix: finish with t, z, or b.
:undo	Undo the last edit; the command-line form of u.
:redo	Redo an undone edit; the command-line form of Ctrl-r.
:delm!	Abbreviation of :delmarks!: remove all local marks.

The island guide

Try it yourself first. Solutions are examples; other correct sequences may also satisfy the goal.

<Esc> = Escape; <Enter> = Enter; <C-r> = Ctrl+r; <Space> = space. An arrow separates fragments and is not a key. Letter case matters.

01 / Lantern Harbor

Every journey begins with a movement

The sea brings you to a harbor where every lantern has gone dark. Neri has found scattered sparks among the quay stones. Gather them, and the outline of the first path will return to the world.

Mission 01 / The first spark

Move right with l, gather the spark, then climb with k and reach the gate.

Start: row 4, column 3

Collect the lights: (4, 6). Cursor destination: (2, 6).

```
1     #########
2     #.......#
3     #.......#
4     #.......#
5     #########
```

Practise these commands: l, k

Hints: l moves right; k moves up. Use both keys in this quest. The journey is not over when you collect the spark: carry its light to the glowing gate.

One possible route

`lllkk`

Mission 02 / Steps of the tide

Use h and j: leave the upper path, gather the spark, and reach the gate to the left.

Start: row 2, column 8

Collect the lights: (4, 6). Cursor destination: (4, 3).

```
1     #########
2     #.......#
3     #.......#
4     #.......#
5     #########
```

Practise these commands: h, j

Hints: h moves left; j moves down. You need both movements. Approach the spark, then move down to it with j. Bring the recovered light to the gate.

One possible route

`hhjjhhh`

Mission 03 / Light the quay

Gather every remaining spark. With the light restored, the harbor will reveal the next island.

Start: row 2, column 2

Collect the lights: (2, 8), (6, 8), (6, 2). Cursor destination: (4, 4).

```
#########
#.......#
#.#####.#
#...#...#
#.#####.#
#.......#
#########
```

Hints: Before moving, look at the whole path and the positions of the lights. When a path ends, retrace a few steps. Take your time: this quest does not reward rushing.

One possible route

`lllllljjjjhhhhhhkkll`

02 / Word Grove

Think in words, not single steps

The trees in this grove grow from words. The Quiet Tide scattered their roots, and Ivo can no longer cross between sentences. He will show you movements that find the beginning and end of whole words.

Mission 04 / Root to root

Reach the marked position by moving between word beginnings with w.

Start: row 1, column 1

Cursor destination: row 1, column 10.

```
1     dew moss lantern river
```

Practise these commands: w

Hints: w moves to the beginning of the next word. Spaces separate the word islands. w crosses such a gap in a single movement.

One possible route

`ww`

Mission 05 / The trail behind

Find an earlier word. Use b to travel backward along the written trail.

Start: row 1, column 10

Cursor destination: row 1, column 5.

```
1     dew moss lantern river
```

Practise these commands: b

Hints: b moves backward to a word's beginning. From inside a word, the first b returns to its beginning. Another b carries you farther back.

One possible route

`b`

Mission 06 / The branch's end

Stop on the last letter of the right word. Try e.

Start: row 1, column 1

Cursor destination: row 1, column 11.

```
1     soft silver light
```

Practise these commands: e

Hints: e moves to the end of a word. Look at the glowing target. A word's beginning and end are different places: choose the movement that lands precisely.

One possible route

`we`

03 / Bridges of Lines

From one edge to the other

The stone bridges remember only beginnings and endings. Sela will teach you to find their edges and give a movement a count. A long crossing need not require many separate keystrokes.

Mission 07 / The first stone

Visit the line's beginning with 0, then reach its first letter with ^.

Start: row 1, column 10

Cursor destination: row 1, column 4.

```
1         stone bridge
```

Practise these commands: 0, ^

Hints: 0 reaches the first column, even when it contains a space. ^ skips leading spaces. Press 0, then ^. The goal is the first letter, but this quest teaches both beginnings.

One possible route

`0^`

Mission 08 / The far bank

Jump to the last line with G, then reach its end with $.

Start: row 1, column 1

Cursor destination: row 3, column 16.

```
1     a quiet river
2     under the bridge
3     the last lantern
```

Practise these commands: G, $

Hints: G reaches the last line of the record. $ reaches the last character of the current line. Combine the two movements: G, then $. You do not need to travel one line at a time.

One possible route

`G$`

Mission 09 / The long stride

Return to the top with gg, move three lines down with 3j, then choose the first column with 0.

Start: row 5, column 5

Cursor destination: row 4, column 1.

```
1     north gate
2     mist
3     old steps
4     star bridge
5     last lantern
```

Practise these commands: gg, count, 0

Hints: gg reaches the first line. The count 3 before j repeats the movement three times. Use gg, then 3j, then 0. The number is part of the command, not text to insert.

One possible route

```
gg3j0
```

04 / Echo Observatory

Find what is not obvious at first glance

The stars still shine above this island, but their names lie scattered across its tablets. Orin hears their returning echoes. Learn to find characters and words, and read a path among them.

Mission 10 / One chosen star

Find the letter l on the current line with f.

Start: row 1, column 1

Cursor destination: row 1, column 8.

```
1     moss | lantern | stars
```

Practise these commands: f

Hints: f waits for the character to find. In this quest, that character is l. Press f, then l. You reach the next l without passing through every letter.

One possible route

`fl`

Mission 11 / A character's echo

Find the next b with fb, then repeat the find with the semicolon ;.

Start: row 1, column 1

Cursor destination: row 1, column 11.

```
1     bell bell bell
```

Practise these commands: f, ;

Hints: Press b after f. ; repeats the last single-character find. Use fb, then ; to reach another b on the same line.

One possible route

`fb;`

Mission 12 / The returning echo

Search for light with /light and Enter. Use n, N, then n again to reach the echo on the final tablet.

Start: row 1, column 1

Cursor destination: row 3, column 10.

```
1     light sleeps
2     follow light
3     wake the light
```

Practise these commands: /, n, N

Hints: n finds another match in the same direction; N searches in the opposite direction. Enter /light, press Enter, then use n, N, n. Practise both search directions in this quest.

One possible route

`/light → <Enter> → nNn`

05 / Ember Forge

Repair boldly. A change can be undone

The hearth still holds enough warmth to forge new letters. Asha does not demand perfect movements: a good craftsperson also knows how to undo a mistake. Clean the damaged records and rekindle the fire.

Mission 13 / A splinter in the record

Remove an unwanted character with x and restore the text.

Start: row 1, column 6

Starting text

```
1      sparkx
```

Target text

```
1      spark
```

Practise these commands: x

Hints: x deletes the character under the cursor. Place the cursor on the unwanted letter first. After deleting, examine the whole text, not only the cursor's position.

One possible route

x

Mission 14 / The metal remembers

Replace the first c with g using rg, undo it with u, then redo it with Ctrl-r.

Start: row 1, column 1

Starting text

```
1      cold ember
2      cold ember
```

Target text

```
1      gold ember
2      cold ember
```

Practise these commands: r, u, Ctrl-r

Hints: r waits for a replacement character. rg changes cold to gold. After rg, use u, then Ctrl-r. The goal is gold ember on the first line and cold ember on the second.

One possible route

`rg → u → <C-r>`

Mission 15 / A new shape

Delete the first broken with dw, then cut the remaining damaged line with dd. Leave only dawn.

Start: row 1, column 1

Starting text

```
1      broken broken
2      dawn
```

Target text

```
1      dawn
```

Practise these commands: dw, dd

Hints: dw deletes text toward the next word. dd deletes the whole current line. First dw, then dd: two sizes of damage, two different tools.

One possible route

`dwdd`

06 / Mirror Library

A good thought deserves an echo

Books on these shelves reflect their pages in the water. Mira can preserve a word before it fades. With a register, you can copy a little light or move a whole line somewhere new.

Mission 16 / A twin line

Copy a line with yy and place its reflection below with p.

Start: row 1, column 1

Starting text

```
1     light
2     sleep
```

Target text

```
1     light
2     light
3     sleep
```

Practise these commands: yy, p

Hints: yy copies the whole current line without removing the original. After copying a line, p puts it below the current one. Choose where the copy should go first.

One possible route

`yyp`

Mission 17 / A word in the mirror

Preserve a word with yw and put it in the right place with p.

Start: row 1, column 1

Starting text

```
1     glow moss
```

Target text

```
1     glow glow moss
```

Practise these commands: yw, p

Hints: yw copies text toward the beginning of the next word. For a text fragment, p puts it after the cursor. This differs from putting a whole line.

One possible route

`ywwhp`

Mission 18 / A new order

Cut a line with dd and move it with p to restore the right order.

Start: row 1, column 1

Starting text

```
1     mist
2     sun
3     sky
```

Target text

```
1     sun
2     mist
3     sky
```

Practise these commands: dd, p

Hints: dd stores the deleted line in the same register used by p. After cutting, move to the destination. p puts the remembered line below the current one.

One possible route

```
ddp
```

07 / Dawn Scriptorium

Leave a new record behind

Eno has found gaps in the chronicle. Movement alone cannot recover these words: someone must write them. Here, you will learn the rhythm of switching between acting on text and inserting it.

Mission 19 / The missing word

Position the cursor, enter Insert mode with i, and type the missing text.

Start: row 1, column 5

Starting text

```
1     the returns
```

Target text

```
1     the dawn returns
```

Practise these commands: i, Esc

Hints: i begins inserting before the cursor. In Insert mode, letters become text. Esc restores Normal mode and its movement commands.

One possible route

`i → dawn<Space> → <Esc>`

Mission 20 / Room for more

Add s with a, return with Esc, then append a space and shine at the line's end with A.

Start: row 1, column 4

Starting text

```
1     star
```

Target text

```
1     stars shine
```

Practise these commands: a, A, Esc

Hints: a begins inserting after the cursor. A begins inserting at the end of the whole line. Turn star into stars with a, s, Esc. Then A, a space, shine, Esc produce stars shine.

One possible route

`as → <Esc> → A<Space>shine → <Esc>`

Mission 21 / Seal the chronicle

Open a new line with o, type returns, leave Insert mode with Esc, and save with :w and Enter.

Start: row 1, column 1

Starting text

```
1      dawn
```

Target text

```
1      dawn
2      returns
```

Practise these commands: o, Esc, :w

Hints: o opens a line below and immediately enters Insert mode. Write returns beneath dawn. Then Esc, :w, and Enter seal the completed tablet.

One possible route

`oreturns → <Esc> → :w → <Enter>`

08 / Morning Tower

Every path leads toward the light

A lens made from the first islands' fragments waits at the summit. Bring your skills together to summon the first dawn. Its light will reveal further realms and a path toward mastery.

Mission 22 / A map remembered

Cut the damaged first line with dd, copy the good line with yy, put it with p, and confirm with :w.

Start: row 1, column 1

Starting text

```
1     lost lost
2     keep the light
```

Target text

```
1     keep the light
2     keep the light
```

Practise these commands: dd, yy, p, :w

Hints: After dd, keep the light remains. yy remembers this whole line; p makes a copy below. Use dd, yy, p, then :w and Enter. The tablet should contain two identical keep the light lines.

One possible route

`ddyyp:w → <Enter>`

Mission 23 / The final repair

Find sleep with /sleep and Enter. Remove it with dw, append sing with A, and save with :w.

Start: row 1, column 1

Starting text

```
1     moss covers dawn
2     the bells sleep
```

Target text

```
1     moss covers dawn
2     the bells sing
```

Practise these commands: /, dw, A, :w

Hints: The search places you on sleep. dw removes that word; A lets you finish the line. Keep the first line unchanged. The second should read the bells sing. Finish inserting with Esc, then use :w and Enter.

One possible route

`/sleep → <Enter> → dwAsing → <Esc> → :w → <Enter>`

Mission 24 / Let there be light

Remove the first line, copy the remaining one, and add the dawn returns on a new line. Save the restored chronicle.

Start: row 1, column 1

Starting text

```
1     the night stays
2     one small spark
```

Target text

```
1     one small spark
2     one small spark
3     the dawn returns
```

Practise these commands: dd, yy, p, o, :w

Hints: dd removes the damaged line; yy and p create a second one small spark. o opens space for the final line. After dd, yy, p, use o and type the dawn returns. Then Esc, :w, and Enter summon the first dawn and open the path onward.

One possible route

`ddyypothe<Space>dawn<Space>returns → <Esc> → :w → <Enter>`

09 / Wind Cliffs

When a word is more than a word

Wind carries torn ribbons of writing. Tavi shows you how to cross an entire unbroken fragment, then retrace its outline without counting letters.

Mission 25 / An unbroken ribbon

Jump across the whole one-two with W and land at the start of three.four. Required commands: W.

Start: row 1, column 1

Cursor destination: row 1, column 9.

```
1     one-two three.four five
```

Practise these commands: W

Hints: Move to the next whitespace-delimited WORD; punctuation stays part of it. Make one deliberate step first and inspect the cursor or text. Add repetitions only when you understand a single operation.

One possible route

```
W
```

Mission 26 / Back along the ridge

Compare E and B with ge and gE, then return to the first column. Required commands: E · B · ge · gE.

Start: row 1, column 23

Cursor destination: row 1, column 1.

```
1     one-two three.four five
```

Practise these commands: E, B, ge, gE

Hints: Move to the end of a whitespace-delimited WORD. Move backward to a WORD's beginning; punctuation does not split it.

One possible route

```
BEgegE0
```

Mission 27 / Measured steps

Practise + and -, give a movement a count, and use a column jump to reach the t in third. Required commands: + · - · | · count.

Start: row 1, column 4

Cursor destination: row 3, column 2.

```
1      first
2        second
3      third
```

Practise these commands: +, -, |, count

Hints: Move to the first non-blank of the next line; a count repeats it. Move to the first non-blank of the previous line.

One possible route

```
+-2+2|
```

10 / Hunters' Pass

Look behind you, too

Single notches mark the pass. Ryn teaches you to search in both directions and stop exactly where the next safe stone begins.

Mission 28 / Against the trail

Find a backward, compare t and T, then reverse with comma and finish just before the last a. Required commands: F · t · T · ,.

Start: row 1, column 13

Cursor destination: row 1, column 12.

```
1     a x a y a z a
```

Practise these commands: F, t, T, ,

Hints: Press F and a character to find it backward on the current line. Move just before the next occurrence of a character on this line.

One possible route

`FataTa,`

Mission 29 / A word calls back

Follow the whole word glow both ways with * and #. Finish on the second glow on the first line. Required commands: * · #.

Start: row 1, column 1

Cursor destination: row 1, column 11.

```
1     glow moss glow
2     glow
```

Practise these commands: *, #

Hints: Search forward for the whole word under the cursor. Search backward for the whole word under the cursor.

One possible route

`*#*`

Mission 30 / The reverse path

Search backward for star with ?, practise n and N, and finish on the middle occurrence. Required commands: ? · n · N.

Start: row 3, column 1

Cursor destination: row 2, column 1.

```
1     star
2     star
3     star
```

Practise these commands: ?, n, N

Hints: Start a backward text search; confirm with Enter. Repeat the search in its existing direction.

One possible route

`?star → <Enter> → nN`

11 / Stone Circles

Every opening has a closing

Uma guards circles hidden within circles. Read brackets, sentence boundaries, and the breathing space of paragraphs until the stones cease to look like a maze.

Mission 31 / Paired stones

Explore nested brackets with % and block-boundary movements. Return to the outer opening brace. Required commands: % · [(·]) · [{ ·]}.

Start: row 2, column 5

Cursor destination: row 1, column 1.

```
1    {
2      [a(b)c]
3    }
```

Practise these commands: %, [(,]), [{,]}

Hints: Jump to the matching (), [], or {} delimiter. Move backward to an enclosing unmatched opening parenthesis.

One possible route

`% → [( → ]) → [{ → ]} → %`

Mission 32 / Between paragraphs

Travel forward and backward between paragraphs. Stop on the blank line between north and south. Required commands: { · }.

Start: row 1, column 1

Cursor destination: row 3, column 1.

```
1    north
2    path
3    [empty line]
4    south
5    gate
6    [empty line]
7    dawn
```

Practise these commands: {, }

Hints: Move backward to a paragraph boundary; blank lines separate paragraphs. Move forward to the next paragraph boundary.

One possible route

`}}{`

Mission 33 / Room to breathe

Visit sentence beginnings in both directions and return to the start of Second. Required commands: (·).

Start: row 1, column 1

Cursor destination: row 1, column 14.

```
1     First light. Second light! Third light?
```

Practise these commands: (,)

Hints: Move backward to a sentence beginning. Move forward to the next sentence beginning.

One possible route

```
))(
```

12 / Horizon Tower

Change your view, not your text

The tower's windows move along a scroll. Noa shows you the difference between where the cursor stands and what the window happens to reveal.

Mission 34 / The window rises

Try all three window positions. Finish with the current, fifth line at the top edge. Required commands: zt · zz · zb.

Start: row 5, column 1

Cursor destination: row 5, column 1.

```
1     line one
2     line two
3     line three
4     line four
5     line five
6     line six
7     line seven
8     line eight
9     line nine
```

First visible line: 5

Practise these commands: zt, zz, zb

Hints: Scroll the current line to the top of the viewport. Center the current line in the viewport without changing text.

One possible route

```
ztzzzbzt
```

Mission 35 / A stable horizon

Center the fifth line, visit the window’s top and bottom, then return to its middle. Required commands: H · M · L.

Start: row 5, column 1

Cursor destination: row 5, column 1.

```
1     line one
2     line two
3     line three
4     line four
5     line five
6     line six
7     line seven
8     line eight
9     line nine
```

First visible line: 3

Practise these commands: H, M, L

Hints: Move to the first visible line of the viewport. Move to the middle visible line of the viewport.

One possible route

`zzHLM`

Mission 36 / Numbers at your side

Enable, disable, and toggle both numbering styles. Leave absolute numbers on and relative numbers off. Required commands: nu · nonu · nu! · rnu · nornu · rnu!.

Start: row 1, column 1

Cursor destination: row 1, column 1.

```
1     north
2     south
```

Absolute numbers: on

Relative numbers: off

Practise these commands: nu, nonu, nu!, rnu, nornu, rnu!

Hints: Enable absolute line numbers. Disable absolute line numbers.

One possible route

`:set<Space>nonu → <Enter> → :set<Space>nu → <Enter> → :set<Space>nu! → <Enter> → :set<Space>nu! → <Enter> → :set<Space>rnu → <Enter> → :set<Space>nornu → <Enter> → :set<Space>rnu! → <Enter> → :set<Space>rnu! → <Enter>`

13 / Weavers' Workshop

Cut, add, and stitch

Ada repairs cloth woven from letters. Sometimes you must reach before the cursor, sometimes above the line, and sometimes begin the whole stitch again.

Mission 37 / Before the thread

Delete x before the cursor, change DAWN to lowercase, and trim junk. Keep the space after dawn. Required commands: X · ~ · D.

Start: row 1, column 2

Starting text

```
1      xDAWN junk
```

Target text

```
1      dawn
```

Practise these commands: X, ~, D

Hints: Delete the character before the cursor. Toggle the case of the character under the cursor.

One possible route

`X4~wD`

Mission 38 / Cut and stitch

Use I to insert the before glow while keeping the indentation. Use O to add header on a new line above. Required commands: I · O.

Start: row 1, column 7

Starting text

```
1        glow
```

Target text

```
1      header
2        the glow
```

Practise these commands: I, O

Hints: Begin inserting at the first non-blank of the line. Open a line above the current one and enter Insert mode.

One possible route

`Ithe<Space> → <Esc> → Oheader → <Esc>`

Mission 39 / A new layer

Replace the single x with sun using s, then replace the whole second line with moon using S. Required commands: s · S.

Start: row 1, column 1

Starting text

```
1     x
2     bad line
```

Target text

```
1     sun
2     moon
```

Practise these commands: s, S

Hints: Delete the character under the cursor and begin inserting. Replace the current line by entering Insert mode.

One possible route

`ssun → <Esc> → jSmoon → <Esc>`

14 / Grammar Foundry

An action and its reach

Keris casts commands in two parts: what to do, and what to do it to. Choose an operator and a movement, and discover the meaning inside a short expression.

Mission 40 / A verb and a path

Compare a word change with c, a whole-line change with cc, and a suffix change with C. Produce warm sun and fixed. Required commands: c · cc · C.

Start: row 1, column 1

Starting text

```
1      cold moss
2      broken
```

Target text

```
1      warm sun
2      fixed
```

Practise these commands: c, cc, C

Hints: Change operator: remove the selected range and begin inserting. Change the whole current line and begin inserting.

One possible route

`cewarm → <Esc> → jccfixed → <Esc> → ggwCsun → <Esc>`

Mission 41 / Preserve and put

Yank a word with operator y and put it before the cursor with P. Also practise Y and deletion while preserving north. Required commands: y · Y · P · d.

Start: row 1, column 1

Starting text

```
1      red blue
2      north
```

Target text

```
1      red red blue
2      north
```

Practise these commands: y, Y, P, d

Hints: Yank operator: combine it with a motion or text object. Yank the whole current line; equivalent to yy in this game.

One possible route

`ywwPYpdd`

Mission 42 / One more time

Delete the first bad with dw and repeat the edit for the next two. Leave only light. Required commands: dw · ..

Start: row 1, column 1

Starting text

```
1     bad bad bad light
```

Target text

```
1     light
```

Practise these commands: dw, .

Hints: Delete from the cursor toward the next word’s beginning. Repeat the last text change, including inserted text.

One possible route

`dw..`

15 / Word Sanctuary

Choose meaning, not distance

Eli does not measure text by counting characters. Here, words, sentences, and paragraphs become ready-made vessels for your intention.

Mission 43 / A word's heart

Compare iw and aw: remove the inside of moss, then blue and its adjacent space. Leave red. Required commands: iw · aw.

Start: row 1, column 6

Starting text

```
1     red moss blue
```

Target text

```
1     red
```

Practise these commands: iw, aw

Hints: The inner word without surrounding spaces; combine with d, y, or c. A word including adjacent whitespace; an operator's text object.

One possible route

`diwwdaw`

Mission 44 / A larger word

Treat moss.green as one WORD. Practise iW and aW, leaving only red. Required commands: iW · aW.

Start: row 1, column 7

Starting text

```
1     red moss.green blue
```

Target text

```
1     red
```

Practise these commands: iW, aW

Hints: The inner WORD: non-whitespace text, including punctuation. A WORD including surrounding whitespace.

One possible route

`diWwdaW`

Mission 45 / Sentences and paragraphs

Yank an inner and whole sentence, then an inner paragraph. Remove the alpha/beta paragraph and its spacing, preserving the sentences and keep. Required commands: is · as · ip · ap.

Start: row 1, column 1

Starting text

```
1     One. Two.
2     [empty line]
3     alpha
4     beta
5     [empty line]
6     keep
```

Target text

```
1     One. Two.
2     [empty line]
3     keep
```

Practise these commands: is, as, ip, ap

Hints: The inner sentence without its separating whitespace. A sentence including separating whitespace.

One possible route

```
yisyasjjyipdap
```

16 / Delimiter Vault

The inside and its frame

Sen guards boxes nested within boxes. Choose only the contents or the entire frame; different names for a key can open the same lock.

Mission 46 / Square and circle

Yank the insides and frames of [] and () with every listed alias. Finish with the complete (glow) in the unnamed register. Required commands: i[· a[· i] · a] · i(· a(· i) · a) · ib · ab.

Start: row 1, column 3

Cursor destination: row 1, column 8.

```
1     [moss] (glow)
```

Register ""

```
1     (glow)
```

Practise these commands: i[, a[, i], a], i(, a(, i), a), ib, ab

Hints: Inside a matching [] pair, excluding brackets; both aliases are equivalent. A complete [] pair, including brackets.

One possible route

```
yi[ya[yi]ya]f(yi(ya(yi)ya)yibyab
```

Mission 47 / Angles and aliases

Compare all {} and <> aliases. Finish with the complete <dawn>, including its frame, in the register. Required commands: i{ · a{ · i} · a} · iB · aB · i< · a< · i> · a>.

Start: row 1, column 3

Cursor destination: row 1, column 9.

```
1     {ember} <dawn>
```

Register ""

```
1     <dawn>
```

Practise these commands: i{, a{, i}, a}, iB, aB, i<, a<, i>, a>

Hints: Inside a matching {} pair, excluding braces; iB is an alias. A complete {} pair, including braces; aB is an alias.

One possible route

```
yi{ya{yi}ya}yiByaBf<yi<ya<yi>ya>
```

Mission 48 / The whole frame

Change only the inside of (seed) to bloom, keeping parentheses, then remove the whole [noise]. Required commands: i(· a[.

Start: row 1, column 11

Starting text

```
1     [noise] (seed) {ash} <sky>
```

Target text

```
1      (bloom) {ash} <sky>
```

Practise these commands: i(, a[

Hints: Inside a matching () pair, excluding parentheses; ib is an alias. A complete [] pair, including brackets.

One possible route

`ci(bloom → <Esc> → 0da[`

17 / Whisper Chapel

Listen between the symbols

Yuna preserves messages inside quotes and tags. To hear the message, learn to distinguish its voice from the frame that carries it.

Mission 49 / Two quiet voices

Compare inner and complete double- and single-quoted objects. Finish by yanking the whole single-quoted glow. Required commands: i" · a" · i' · a'.

Start: row 1, column 3

Cursor destination: row 1, column 8.

```
1     "moss" 'glow'
```

Register ""

```
1     'glow'
```

Practise these commands: i", a", i', a'

Hints: Text inside double quotes, excluding the quotes. Text including its surrounding double quotes.

One possible route

`yi"ya"f'yi'ya'`

Mission 50 / A whisper and a tag

Yank inside and around backticks, then an HTML element. Preserve the complete glow. Required commands: i` · a` · it · at.

Start: row 1, column 3

Cursor destination: row 1, column 8.

```
1     `seed` <b>glow</b>
```

Register ""

```
1     <b>glow</b>
```

Practise these commands: i`, a`, it, at

Hints: Text inside backticks, excluding the delimiters. Text including surrounding backticks.

One possible route

`yi`ya`f<yityat`

Mission 51 / A nested message

Change only the p element’s contents to new, keeping its tags. Remove the whole quoted noise and keep the final space. Required commands: it · a".

Start: row 1, column 5

Starting text

```
1     <p>old</p> "noise"
```

Target text

```
1     <p>new</p>
```

Practise these commands: it, a"

Hints: The inside of a matching HTML tag pair, excluding the tags. Text including its surrounding double quotes.

One possible route

`citnew → <Esc> → f"da"`

18 / Cartographers' Camp

Leave a point you can return to

Aro draws maps that wind cannot scatter. Remember a position, return to its line or exact column, and remove landmarks that no longer guide you.

Mission 52 / Set a landmark

Set mark a on the second line. Compare a line jump with an exact-column return, then return to the saved position. Required commands: m · ' · `.

Start: row 2, column 5

Cursor destination: row 2, column 5.

```
1       north
2       south
3       dawn
```

Mark a: row 2, column 5

Practise these commands: m, ', `

Hints: Set a mark: m and a letter, such as ma, remember the current position. A quote and mark letter jump to the first non-blank of its line.

One possible route

`magg'agg`a`

Mission 53 / Read the map

Set a and b in adjacent columns, open the marks list, and remove only a with the :delm abbreviation. Required commands: :marks · :delm.

Start: row 1, column 1

Cursor destination: row 1, column 2.

```
1     path
```

Mark b: row 1, column 2

Removed marks: a

Open list: Marks

Practise these commands: :marks, :delm

Hints: List saved marks. Remove specified marks, such as :delmarks a; :delm is an abbreviation.

One possible route

`malmb → :marks → <Enter> → :delm<Space>a → <Enter>`

Mission 54 / Erase old routes

Practise deleting a mark range, :delmarks!, and :delm!. Finish with a, b, and c removed. Required commands: :delmarks · :delmarks! · :delm!.

Start: row 1, column 1

Cursor destination: row 1, column 2.

```
1     path
```

Removed marks: a, b, c

Practise these commands: :delmarks, :delmarks!, :delm!

Hints: Remove specified marks, such as :delmarks a; :delm is an abbreviation. Remove all local marks.

One possible route

`malmb → :delmarks<Space>a-b → <Enter> → ma → :delmarks! → <Enter> → mc → :delm! → <Enter>`

19 / Cabinet of Echoes

A clipboard can have a name

Esme keeps echoes in separate drawers. Preserve one, add to another, and let unwanted noise vanish without drowning out what matters.

Mission 55 / An unnamed echo

Delete a, restore it from the small-delete register, then yank all of ab through the explicit unnamed register. Required commands: "- · "" · x · yy · P.

Start: row 1, column 1

Target text

```
1      ab
```

Register ""

```
1      ab
```

Register "-

```
1      a
```

Practise these commands: "-, "", x, yy, P

Hints: Small-delete register: deleted text smaller than a whole line. The unnamed register holds the latest yanked or deleted text.

One possible route

`x"-P""yy`

Mission 56 / More without overwriting

Keep moss in a, append glow through A, then put both lines. Inspect the contents of register a. Required commands: "a · "A · p.

Start: row 1, column 1

Starting text

```
1     moss
2     glow
```

Target text

```
1     moss
2     moss
3     glow
4     glow
```

Register "a

```
1     moss
2     glow
```

Practise these commands: "a, "A, p

Hints: Named a-z register: select its letter before an operator to keep text. Uppercase A-Z appends to the corresponding a-z register instead of replacing it.

One possible route

`"ayyj"Ayygg"ap`

Mission 57 / Silence keeps the copy

Keep keep in a, delete trash into the black hole, then put the preserved copy above the current line. Required commands: "_ · "a · dd · P.

Start: row 1, column 1

Starting text

```
1  keep
2  trash
```

Target text

```
1  keep
2  keep
```

Register "a

```
1  keep
```

Register ""

```
1  keep
```

Practise these commands: "_, "a, dd, P

Hints: Black hole: delete without overwriting other registers. Named a-z register: select its letter before an operator to keep text.

One possible route

```
"ayyj"_dd"aP
```

20 / Chamber of Nine

Memory has an order

Nona remembers more than the most recent disappearance. Open the drawers of history and discover that deleting need not mean saying goodbye.

Mission 58 / The last copy

Yank light, delete dust, then recover the copy from register 0. Register 1 should still remember dust. Required commands: "0 · dd.

Start: row 1, column 1

Starting text

```
1     light
2     dust
```

Target text

```
1     light
2     light
```

Register "0

```
1     light
```

Register "1

```
1     dust
```

Practise these commands: "0, dd

Hints: Register 0 keeps the latest yank, even after a subsequent deletion. Cut the current line and keep it in a register.

One possible route

`yyjdd"0p`

Mission 59 / A chain of departures

Delete nine lines and inspect history registers 1 through 9. Restore the oldest one above keep. Required commands: "1 · "2 · "3 · "4 · "5 · "6 · "7 · "8 · "9.

Start: row 1, column 1

Starting text

```
1     one
2     two
3     three
4     four
5     five
6     six
7     seven
8     eight
9     nine
10    keep
```

Target text

```
1     one
2     keep
```

Register "1

```
1     nine
```

Register "2

```
1     eight
```

Register "3

```
1     seven
```

Register "4

```
1     six
```

Register "5

```
1     five
```

Register "6

```
1     four
```

Register "7

```
1     three
```

Register "8

```
1     two
```

Register "9

```
1     one
```

Practise these commands: "1, "2, "3, "4, "5, "6, "7, "8, "9

Hints: Larger-deletion history: 1 is newest, 9 the oldest retained. Larger-deletion history: 1 is newest, 9 the oldest retained.

One possible route

`dd........ → "1pu → "2pu → "3pu → "4pu → "5pu → "6pu → "7pu → "8pu → "9P`

Mission 60 / Read the drawers

Yank one into a, then open the register inspector. Check that a and 0 retain the same line. Required commands: :reg.

Start: row 1, column 1

Cursor destination: row 1, column 1.

```
1     one
2     two
```

Register "0

```
1     one
```

Register "a

```
1     one
```

Open list: Registers

Practise these commands: :reg

Hints: Show registers; optionally specify names, such as :reg a0. Make one deliberate step first and inspect the cursor or text. Add repetitions only when you understand a single operation.

One possible route

`"ayy → :reg → <Enter>`

21 / Scribes' Harbor

Many pages, one journey

Tero ferries documents between virtual harbors. Save what you wish to keep, and switch between open pages without losing track of their names.

Mission 61 / A page remembered

Save dawn rises in notes.txt, change echo.txt to awake, and save it. Return to notes.txt without losing either change. Required commands: :w · :e · c.

Start: row 1, column 1

Starting text

```
1  dawn
```

Target text

```
1  dawn rises
```

Active buffer: notes.txt

Saved virtual file: notes.txt

```
1  dawn rises
```

Saved virtual file: echo.txt

```
1  awake
```

Practise these commands: :w, :e, c

Hints: Save the virtual document; an optional name selects an in-game file. Open a virtual file with the given name.

One possible route

```
A<Space>rises → <Esc> → :w → <Enter> → :e<Space>echo.txt → <Enter> → ccawake → <Esc> → :w → <Enter> → :e<Space>notes.txt → <Enter>
```

Mission 62 / Another harbor

Open beta.txt, list the buffers, and return by name to alpha.txt. Required commands: :e · :ls · :b.

Start: row 1, column 1

Cursor destination: row 1, column 1.

```
1  north
```

Active buffer: alpha.txt

Open list: Buffers

Practise these commands: :e, :ls, :b

Hints: Open a virtual file with the given name. List open buffers and their numbers.

One possible route

```
:e<Space>beta.txt → <Enter> → :ls → <Enter> → :b<Space>alpha.txt → <Enter>
```

Mission 63 / The buffer crossing

Save a copy as final.txt, open it, and compare switching buffers by number. Finish in final.txt. Required commands: :w · :e · :b.

Start: row 1, column 1

Target text

```
1     light
```

Active buffer: final.txt

Saved virtual file: final.txt

```
1     light
```

Practise these commands: :w, :e, :b

Hints: Save the virtual document; an optional name selects an in-game file. Open a virtual file with the given name.

One possible route

`:w<Space>final.txt → <Enter> → :e<Space>final.txt → <Enter> → :b<Space>1 → <Enter> → :b<Space>2 → <Enter>`

22 / Clockwork Orchard

Let one good movement bear many fruits

Omi tends the orchard's delicate mechanisms. Record one reliable cycle, check where it ends, and let it work along the next branches.

Mission 64 / A small machine

Record appending ! and moving down in a. Replay once to change the first two dawn lines. Required commands: q · @.

Start: row 1, column 1

Starting text

```
1  dawn
2  dawn
3  dawn
```

Target text

```
1  dawn!
2  dawn!
3  dawn
```

Recorded macros: a

Practise these commands: q, @

Hints: q and a letter start recording a macro; another q stops it. Replay a register's macro, such as @a; a count repeats it.

One possible route

`qaA! → <Esc> → jq@a`

Mission 65 / The orchard repeats

Record adding + at line start and moving down. Use @a and @@, leaving the fourth seed unchanged. Required commands: q · @ · @@.

Start: row 1, column 1

Starting text

```
1  seed
2  seed
3  seed
4  seed
```

Target text

```
1  +seed
2  +seed
3  +seed
4  seed
```

Recorded macros: a

Practise these commands: q, @, @@

Hints: q and a letter start recording a macro; another q stops it. Replay a register's macro, such as @a; a count repeats it.

One possible route

`qaI+ → <Esc> → jq@a@@`

Mission 66 / Once more, precisely

Record deleting one x, then replay the macro three times. Leave only light. Required commands: q · @ · count.

Start: row 1, column 1

Starting text

```
1  xxxxlight
```

Target text

```
1  light
```

Recorded macros: a

Practise these commands: q, @, count

Hints: q and a letter start recording a macro; another q stops it. Replay a register's macro, such as @a; a count repeats it.

One possible route

`qaxq3@a`

23 / Weavers' Trials

Combine the patterns you know

Vel stretches a long pattern before you. Counts, repeated edits, and macros are different tools: choose the one that expresses the work you intend.

Mission 67 / Two counts

Combine operator and movement counts: delete six words with a single 2d3w and preserve keep. Required commands: count · dw.

Start: row 1, column 1

Starting text

```
1     one two three four five six keep
```

Target text

```
1     keep
```

Practise these commands: count, dw

Hints: A count before an operator or motion sets repetitions; 2d3w combines both counts. Delete from the cursor toward the next word's beginning.

One possible route

`2d3w`

Mission 68 / The same weave

Delete x, undo and redo with :undo and :redo, then repeat the deletion with dot. Required commands: :undo · :redo · .

Start: row 1, column 1

Starting text

```
1     xxdawn
```

Target text

```
1     dawn
```

Practise these commands: :undo, :redo, .

Hints: Undo the last edit; the command-line form of u. Redo an undone edit; the command-line form of Ctrl-r.

One possible route

`x → :undo → <Enter> → :redo → <Enter> → .`

Mission 69 / A reliable pattern

Record deleting ash and moving down, repeat twice, test register 0 on keep, and undo the trial copy. Required commands: q · @ · count · "0.

Start: row 1, column 1

Starting text

```
1      ash dawn
2      ash dawn
3      ash dawn
4      keep
```

Target text

```
1      dawn
2      dawn
3      dawn
4      keep
```

Register "0

```
1      keep
```

Recorded macros: a

Practise these commands: q, @, count, "0

Hints: q and a letter start recording a macro; another q stops it. Replay a register's macro, such as @a; a count repeats it.

One possible route

```
qadwjq2@ayy"0Pu
```

24 / Summit of Beginnings

The whole archipelago in one record

Luma waits where all the paths meet. She offers no new powers: only a final task that asks you to join memory, precision, and imagination.

Mission 70 / A record without noise

Change the tag's contents to light, remove the whole [noise] and its empty line, then save. Required commands: it · a[· dd · :w.

Start: row 1, column 5

Starting text

```
1     <p>old</p>
2     [noise]
3     keep
```

Target text

```
1     <p>light</p>
2     keep
```

Practise these commands: it, a[, dd, :w

Hints: The inside of a matching HTML tag pair, excluding the tags. A complete [] pair, including brackets.

One possible route

`citlight → <Esc> → jda[dd → :w → <Enter>`

Mission 71 / The precise way back

Remember two positions with A and a. Combine exact jumps with searches in both directions, then return to dawn. Required commands: m · ` · / · ? · n.

Start: row 1, column 6

Cursor destination: row 3, column 6.

```
1     star north
2     star south
3     star dawn
```

Mark A: row 1, column 6

Mark a: row 3, column 6

Practise these commands: m, `, /, ?, n

Hints: Set a mark: m and a letter, such as ma, remember the current position. A backtick and mark letter jump to the exact saved position.

One possible route

`mA/dawn → <Enter> → ma`A?star → <Enter> → n`a`

Mission 72 / Every light, one sky

Record changing sleep to wake and apply it on three lines. Fetch the final line from message.txt, put it in dawn.txt, and save. Required commands: q · @ · @@ · :w · :e · yy · "0 · p.

Start: row 1, column 1

Starting text

```
1     sleep
2     sleep
3     sleep
```

Target text

```
1     wake
2     wake
3     wake
4     the dawn returns
```

Register "0

```
1     the dawn returns
```

Active buffer: dawn.txt

Saved virtual file: dawn.txt

```
1     wake
2     wake
3     wake
4     the dawn returns
```

Saved virtual file: message.txt

```
1     the dawn returns
```

Recorded macros: a

Practise these commands: q, @, @@, :w, :e, yy, "0, p

Hints: q and a letter start recording a macro; another q stops it. Replay a register's macro, such as @a; a count repeats it.

One possible route

```
qaccwake → <Esc> → jq@a@@ → :w → <Enter> → :e<Space>message.txt → <Enter> → yy → :e<Space>dawn.txt → <Enter> → "0p → :w → <Enter>
```

Keep learning

This handbook describes the educational subset supported by the game. Virtual files stay in the browser; the game does not expose your device file system.

Vim documentation

Motions and text objects
https://vimhelp.org/motion.txt.html

Editing and registers
https://vimhelp.org/change.txt.html

Macros
https://vimhelp.org/usr_10.txt.html

Files and buffers
https://vimhelp.org/usr_22.txt.html

LUMEN

Manual del guardián de la luz

24 regiones, 72 misiones. Guía práctica de los movimientos y el lenguaje de edición de Vim.

24 islas / 72 misiones / todo gratis

Español / Edición 0.3.0

Este manual describe el subconjunto educativo que admite el juego. Los archivos virtuales permanecen en el navegador; el juego no accede al sistema de archivos de tu dispositivo.

Inténtalo primero. Las soluciones son ejemplos; otras secuencias correctas también pueden cumplir el objetivo.

Índice

Antes del viaje 3

Atlas de comandos 5

Guía de las islas 10

01 / Puerto de los Faroles . . . 11
02 / Arboleda de Palabras . . . 13
03 / Puentes de Líneas . . . 14
04 / Observatorio del Eco . . . 16
05 / Forja de las Brasas . . . 17
06 / Biblioteca de los Espejos . . . 19
07 / Escritorio del Alba . . . 21
08 / Torre de la Mañana . . . 23
09 / Acantilados del Viento . . . 25
10 / Paso de los Rastreadores . . . 26
11 / Círculos de Piedra . . . 27
12 / Torre del Horizonte . . . 29
13 / Taller de Tejedores . . . 31
14 / Fundición de Gramática . . . 33
15 / Santuario de Palabras . . . 35
16 / Bóveda de Delimitadores . . . 37
17 / Capilla de los Susurros . . . 39
18 / Campamento de Cartógrafos . . . 41
19 / Gabinete de Ecos . . . 43
20 / Cámara de los Nueve . . . 45
21 / Puerto de Escribas . . . 48
22 / Huerto Mecánico . . . 50
23 / Pruebas de Tejedores . . . 52
24 / Cumbre de los Comienzos . . . 54

Sigue aprendiendo 56

Antes del viaje

Tu objetivo

El viaje enseña a editar texto de forma consciente, desde un movimiento hasta objetos de texto, registros y macros. Cada región introduce una familia pequeña de órdenes. Las misiones requieren utilizar las herramientas indicadas, no obtener el resultado por casualidad.

Completar el juego significa practicar su programa, no dominar todas las funciones del editor Vim completo. Vuelve a las misiones terminadas y busca soluciones claras que puedas repetir.

Cómo jugar

Elige el idioma, da nombre a tu personaje y acepta las condiciones. Haz clic en el tablero para dirigirle el teclado. En modo Normal, las letras ejecutan órdenes; en modo de inserción, escriben texto. Esc vuelve al modo Normal. En pantallas táctiles usa las teclas bajo el tablero y los campos de texto.

Antes de actuar, lee el objetivo, compara el búfer con el texto esperado y consulta las órdenes necesarias. Las pistas son progresivas. Reiniciar afecta a la misión actual; el mapa permite volver a regiones desbloqueadas. El botón Ayuda abre las explicaciones y el manual en cualquier momento. Esc sale de inserción al modo Normal; en modo Normal, sin órdenes pendientes, abre la pausa. También hay un botón de pausa. Las tres últimas misiones pueden jugarse como pruebas de tiempo opcionales. El tiempo orientativo es un reto adicional, no una condición para completar el programa. Practica primero la precisión y luego la rapidez.

Guardado y privacidad

El nombre, la aceptación de las condiciones y el progreso se guardan en el localStorage de este navegador. No es una cuenta ni una copia en la nube. La navegación privada, el borrado de datos o las restricciones de almacenamiento pueden eliminarlo. Descarga una copia para conservar tu partida o llevarla a otro dispositivo.

Los diagnósticos opcionales están desactivados inicialmente y disponen de un interruptor separado en ajustes. No son necesarios para jugar. Incluyen datos técnicos limitados sobre errores, no tu nombre, el contenido de ejercicios ni toda la partida. Las condiciones detallan el tratamiento de los datos.

Cómo leer las teclas

Importan las mayúsculas: w y W, p y P, i e I son órdenes distintas. 3w significa pulsar 3 y luego w; Ctrl-r significa mantener Control y pulsar r. +, -, |, $, %, punto y coma y coma son teclas reales de órdenes.

Una orden como ciw combina el operador c con el objeto de texto iw. Escribe exactamente los nombres y textos objetivo de las tablillas, aunque las instrucciones estén en otro idioma. Confirma con Enter las órdenes con : y las búsquedas con / o ?. Escribe las secuencias en orden, sin las flechas que separan pasos. <Esc>, <Enter>, <C-r> y <Space> significan Escape, Enter, Control+r y espacio; no escribas los corchetes angulares.

El juego y Vim real

LUMEN es un simulador educativo de un conjunto seleccionado de habilidades de Vim. El tablero representa un búfer de texto y su parte visible es una ventana. Las órdenes de desplazamiento cambian la vista, no el contenido. Los archivos son virtuales: el juego no lee carpetas del equipo ni ejecuta órdenes del sistema.

Las órdenes siguen las operaciones enseñadas, pero el juego no sustituye al Vim completo con todas sus opciones, extensiones y excepciones de sintaxis. La ayuda y este manual describen el ámbito compatible.

Línea de comandos

Los dos puntos abren la línea de órdenes. :w guarda un documento virtual, :e abre otro, :ls lista los búferes y :b cambia de búfer. :marks muestra las marcas; :delmarks y su abreviatura :delm las eliminan; :delmarks! borra las marcas locales. :reg muestra los registros.

:set nu y :set nonu activan y desactivan los números absolutos; :set nu! los alterna. :set rnu, :set nornu y :set rnu! hacen lo mismo con los relativos. Enter confirma y Esc cancela.

Registros

Un registro es un portapapeles de Vim. La copia y el borrado normales utilizan el registro sin nombre; el prefijo de comillas dobles selecciona otro. a-z son registros con nombre y escribir en A-Z añade al correspondiente registro en minúscula.

El registro 0 conserva la última copia, 1-9 mantienen el historial de borrados grandes y - guarda un borrado pequeño. _ es el agujero negro: permite borrar sin reemplazar los demás portapapeles. :reg muestra lo que realmente quedó guardado.

Macros

Una macro graba una secuencia de acciones, incluidos los movimientos. Para grabar en el registro a, pulsa qa, realiza las acciones y termina con q. @a reproduce ese registro; @@ repite la última macro reproducida. Un número delante indica cuántas veces repetir.

Prueba una ejecución antes de repetir muchas. Una macro útil termina en una posición previsible, preparada para procesar el siguiente fragmento.

Archivos y búferes virtuales

Un búfer es un documento abierto; el nombre de archivo identifica su copia virtual guardada. :w [nombre] guarda, :e nombre abre, :ls enumera y :b nombre o número cambia de búfer. Guarda los cambios que quieras conservar antes de salir.

Todo ocurre dentro del juego y su partida local. Exportar el viaje crea una copia de seguridad, no un archivo fuente real en el sistema del equipo. No escribas contraseñas ni información confidencial en los ejercicios.

<Esc> = Escape; <Enter> = Intro; <C-r> = Ctrl+r; <Space> = espacio. La flecha separa fragmentos y no es una tecla. Las mayúsculas importan.

Atlas de comandos

Comando	Qué hace
`0`	Ve a la primera columna de la línea actual.
`hjkl`	h a la izquierda, j abajo, k arriba, l a la derecha: los movimientos básicos.
`h`	Mueve una casilla o carácter a la izquierda.
`j`	Mueve una línea o casilla hacia abajo.
`k`	Mueve una línea o casilla hacia arriba.
`l`	Mueve una casilla o carácter a la derecha.
`w`	Ve al principio de la siguiente palabra.
`b`	Retrocede al principio de una palabra.
`e`	Ve al final de una palabra.
`^`	Ve al primer carácter de la línea que no sea un espacio.
`$`	Ve al último carácter de la línea actual.
`gg`	Ve a la primera línea.
`G`	Ve a la última línea.
`count`	Un número repite el siguiente movimiento, como 3w o 2j.
`f`	Pulsa f y un carácter para encontrar su siguiente aparición en esta línea.
`;`	Repite la última búsqueda de un carácter hecha con f.
`/`	Busca texto hacia delante y confirma con Enter.
`n`	Repite la búsqueda en la dirección que ya tenía.
`N`	Repite en dirección contraria a la última búsqueda / o ?.
`x`	Borra el carácter bajo el cursor.
`r`	Sustituye el carácter bajo el cursor por el siguiente que escribas.
`dd`	Corta la línea actual y consérvala en un registro.
`dw`	Borra desde el cursor hasta el comienzo de la palabra siguiente.
`u`	Deshaz el último cambio del texto.
`Ctrl-r`	Rehaz un cambio deshecho.
`yy`	Copia toda la línea actual en el registro.

Comando	Qué hace
`yw`	Copia texto hacia el principio de la siguiente palabra.
`p`	Pega texto después del cursor o líneas enteras debajo de la actual.
`i`	Empieza a insertar antes del cursor.
`a`	Empieza a insertar después del cursor.
`A`	Empieza a insertar al final de la línea.
`o`	Abre una línea nueva debajo y empieza a insertar.
`Esc`	Vuelve al modo Normal o cancela una orden pendiente.
`:w`	Guarda el documento virtual; un nombre opcional indica el archivo del juego.
`1 ... 9`	Un número ante operador o movimiento indica repeticiones; 2d3w combina ambos.
`W`	Ve al siguiente WORD separado por espacios; la puntuación sigue dentro.
`E`	Ve al final de un WORD separado por espacios.
`B`	Retrocede al principio de un WORD; la puntuación no lo divide.
`ge`	Retrocede al final de la palabra anterior.
`gE`	Retrocede al final del WORD anterior.
`+`	Ve al primer carácter no vacío de la línea siguiente; admite un número.
`-`	Ve al primer carácter no vacío de la línea anterior.
`|`	Ve a la columna indicada, por ejemplo 5|; sin número, a la primera.
`F`	Pulsa F y un carácter para buscarlo hacia atrás en esta línea.
`t`	Avanza hasta justo antes de la siguiente aparición de un carácter.
`T`	Busca un carácter hacia atrás y detente justo después.
`,`	Repite el último f, F, t o T en la dirección contraria.
`*`	Busca hacia delante la palabra completa bajo el cursor.
`#`	Busca hacia atrás la palabra completa bajo el cursor.
`?`	Abre una búsqueda hacia atrás y confirma con Enter.
`%`	Salta al delimitador correspondiente (), [] o {}.
`[{`	Retrocede a la llave de apertura del bloque envolvente.
`]}`	Avanza a la llave de cierre del bloque envolvente.
`[(`	Retrocede al paréntesis de apertura envolvente.

Comando	Qué hace
`])`	Avanza al paréntesis de cierre envolvente.
`{`	Retrocede al límite de un párrafo; las líneas vacías los separan.
`}`	Avanza al siguiente límite de párrafo.
`(`	Retrocede al principio de una oración.
`)`	Avanza al principio de la siguiente oración.
`zt`	Desplaza la vista para dejar la línea actual arriba.
`zz`	Centra la línea actual sin modificar el texto.
`zb`	Desplaza la línea actual a la parte inferior de la vista.
`H`	Ve a la primera línea visible de la ventana.
`M`	Ve a la línea visible del centro de la ventana.
`L`	Ve a la última línea visible de la ventana.
`nu / :set nu`	Activa los números de línea absolutos.
`nonu / :set nonu`	Desactiva los números de línea absolutos.
`nu! / :set nu!`	Alterna los números de línea absolutos.
`rnu / :set rnu`	Activa números relativos que muestran la distancia al cursor.
`nornu / :set nornu`	Desactiva los números de línea relativos.
`rnu! / :set rnu!`	Alterna los números de línea relativos.
`X`	Borra el carácter anterior al cursor.
`D`	Borra desde el cursor hasta el final de la línea.
`~`	Invierte la mayúscula o minúscula bajo el cursor.
`I`	Inserta en el primer carácter no vacío de la línea.
`O`	Abre una línea encima y entra en inserción.
`s`	Borra el carácter bajo el cursor y empieza a insertar.
`S`	Sustituye la línea actual entrando en inserción.
`d`	Operador de borrado: añade un movimiento u objeto, como d$, diw o d2w.
`y`	Operador de copia: combínalo con un movimiento u objeto.
`c`	Operador de cambio: elimina el rango e inicia la inserción.
`cc`	Cambia toda la línea actual e inicia la inserción.

Comando	Qué hace
`C`	Cambia desde el cursor hasta el final e inicia la inserción.
`Y`	Copia toda la línea actual; equivale a yy en este juego.
`.`	Repite el último cambio, incluido el texto insertado.
`iw`	La palabra interior sin espacios alrededor; combina con d, y o c.
`aw`	Una palabra con espacio adyacente; objeto para un operador.
`iW`	El interior de un WORD: texto sin espacios, incluida puntuación.
`aW`	Un WORD con el espacio en blanco adyacente.
`is`	La oración interior sin el espacio que la separa.
`as`	Una oración con el espacio que la separa.
`ip`	El párrafo interior sin líneas vacías alrededor.
`ap`	Un párrafo con las líneas vacías que lo separan.
`i[ / i]`	Interior de un par [], sin corchetes; ambos alias equivalen.
`a[ / a]`	Un par [] completo, incluidos los corchetes.
`i( / i) / ib`	Interior de un par (), sin paréntesis; ib es un alias.
`a( / a) / ab`	Un par () completo, incluidos los paréntesis; ab es un alias.
`i{ / i} / iB`	Interior de un par {}, sin llaves; iB es un alias.
`a{ / a} / aB`	Un par {} completo, incluidas las llaves; aB es un alias.
`i< / i>`	Interior de un par <>, sin los delimitadores.
`a< / a>`	Un par <> completo, incluidos los delimitadores.
`i"`	Texto dentro de comillas dobles, sin las comillas.
`a"`	Texto con las comillas dobles que lo rodean.
`i'`	Texto dentro de comillas simples, sin delimitadores.
`a'`	Texto con las comillas simples que lo rodean.
`` i` ``	Texto dentro de acentos graves, sin delimitadores.
`` a` ``	Texto con los acentos graves que lo rodean.
`it`	Interior de una pareja de etiquetas HTML, sin las etiquetas.
`at`	Elemento completo con sus etiquetas de apertura y cierre.
`m`	Crea una marca: m y una letra, como ma, recuerdan la posición.

Comando	Qué hace
`'`	Comilla simple y letra de marca saltan al primer carácter de su línea.
`` ` ``	Acento grave y letra de marca saltan a la posición exacta.
`:marks`	Muestra las marcas guardadas.
`:delmarks / :delm`	Elimina marcas indicadas, como :delmarks a; :delm es la abreviatura.
`:delmarks!`	Elimina todas las marcas locales.
`"`	Prefijo de registro: comilla doble, nombre y después operador o p.
`""`	El registro sin nombre guarda el último texto copiado o borrado.
`"-`	Registro de borrado pequeño: texto menor que una línea completa.
`"_`	Agujero negro: borra sin sobrescribir otros registros.
`"0`	El registro 0 conserva la última copia incluso tras un borrado.
`:reg / :registers`	Muestra registros; puedes indicar nombres, como :reg a0.
`:e`	Abre un archivo virtual con el nombre indicado.
`:ls`	Muestra los búferes abiertos y sus números.
`:b`	Cambia de búfer virtual por nombre o número.
`q`	q y una letra inician una macro; otra q termina la grabación.
`@`	Reproduce la macro de un registro, como @a; admite repeticiones.
`@@`	Reproduce de nuevo la última macro utilizada.
`"1 ... "9`	Historial de borrados grandes: 1 es el más reciente y 9 el más antiguo conservado.
`"a ... "z`	Registro a-z: elige su letra antes del operador para conservar texto.
`"A ... "Z`	A-Z añade al registro a-z correspondiente en lugar de sustituirlo.
`P`	Pega texto antes del cursor o una línea entera encima de la actual.
`:set`	Cambia opciones de vista, como :set nu o :set rnu!.
`:`	Abre la línea de órdenes; escribe una orden y confirma con Enter.
`z`	Prefijo de desplazamiento: completa con t, z o b.
`:undo`	Deshaz el último cambio; forma de u en la línea de órdenes.
`:redo`	Rehaz un cambio; forma de Ctrl-r en la línea de órdenes.
`:delm!`	Abreviatura de :delmarks!: elimina todas las marcas locales.

Guía de las islas

Inténtalo primero. Las soluciones son ejemplos; otras secuencias correctas también pueden cumplir el objetivo.

<Esc> = Escape; <Enter> = Intro; <C-r> = Ctrl+r; <Space> = espacio. La flecha separa fragmentos y no es una tecla. Las mayúsculas importan.

01 / Puerto de los Faroles

Todo viaje empieza con un movimiento

El mar te lleva a un puerto donde todos los faroles se han apagado. Neri ha encontrado unas chispas perdidas entre las piedras del muelle. Recógelas y el primer camino volverá a dibujarse en el mundo.

Misión 01 / La primera chispa

Avanza a la derecha con l, recoge la chispa y después sube con k hasta la puerta.

Inicio: línea 4, columna 3

Recoge las luces: (4, 6). Destino del cursor: (2, 6).

```
1     #########
2     #.......#
3     #.......#
4     #.......#
5     #########
```

Practica estos comandos: l, k

Pistas: l mueve a la derecha y k hacia arriba. Utiliza ambas teclas en esta misión. El viaje no termina al recoger la chispa: lleva su luz hasta la puerta iluminada.

Una ruta posible

`lllkk`

Misión 02 / Peldaños de la marea

Utiliza h y j: sal del camino superior, recoge la chispa y llega a la puerta de la izquierda.

Inicio: línea 2, columna 8

Recoge las luces: (4, 6). Destino del cursor: (4, 3).

```
1     #########
2     #.......#
3     #.......#
4     #.......#
5     #########
```

Practica estos comandos: h, j

Pistas: h mueve a la izquierda y j hacia abajo. Necesitas ambos movimientos. Acércate a la chispa y baja hasta ella con j. Lleva después la luz recuperada a la puerta.

Una ruta posible

`hhjjhhh`

Misión 03 / Iluminar el muelle

Recoge todas las chispas que quedan. Con la luz recuperada, el puerto revelará la siguiente isla.

Inicio: línea 2, columna 2

Recoge las luces: (2, 8), (6, 8), (6, 2). Destino del cursor: (4, 4).

```
1     #########
2     #.......#
3     #.#####.#
4     #...#...#
5     #.#####.#
6     #.......#
7     #########
```

Pistas: Antes de moverte, observa el camino entero y la posición de las luces. Si un camino termina, retrocede unos pasos. Puedes tomarte tu tiempo: aquí no se premia la prisa.

Una ruta posible

```
lllllljjjjhhhhhhkkll
```

02 / Arboleda de Palabras

Piensa en palabras, no en pasos sueltos

Los árboles de esta arboleda crecen de las palabras. La Marea Silenciosa dispersó sus raíces e Ivo ya no puede cruzar entre frases. Te enseñará movimientos que encuentran el principio y el final de palabras enteras.

Misión 04 / De raíz en raíz

Llega a la posición indicada saltando entre principios de palabras con w.

Inicio: línea 1, columna 1

Destino del cursor: línea 1, columna 10.

```
1     dew moss lantern river
```

Practica estos comandos: w

Pistas: w lleva al principio de la siguiente palabra. Los espacios separan las islas de palabras. w cruza ese hueco con un solo movimiento.

Una ruta posible

`ww`

Misión 05 / El rastro que dejamos

Encuentra una palabra anterior. Usa b para retroceder por el sendero escrito.

Inicio: línea 1, columna 10

Destino del cursor: línea 1, columna 5.

```
1     dew moss lantern river
```

Practica estos comandos: b

Pistas: b retrocede al principio de una palabra. Desde el interior de una palabra, la primera b vuelve a su principio. Otra b te lleva más atrás.

Una ruta posible

`b`

Misión 06 / El final de la rama

Detente en la última letra de la palabra correcta. Prueba e.

Inicio: línea 1, columna 1

Destino del cursor: línea 1, columna 11.

```
1     soft silver light
```

Practica estos comandos: e

Pistas: e lleva al final de una palabra. Mira el objetivo iluminado. El principio y el final de una palabra son lugares diferentes: elige el movimiento preciso.

Una ruta posible

`we`

03 / Puentes de Líneas

De un extremo al otro

Los puentes de piedra solo recuerdan comienzos y finales. Sela te enseñará a encontrar sus bordes y a dar una cantidad a cada movimiento. Un cruce largo no tiene por qué exigir muchas pulsaciones separadas.

Misión 07 / La primera piedra

Visita el principio de la línea con 0 y después llega a su primera letra con ^.

Inicio: línea 1, columna 10

Destino del cursor: línea 1, columna 4.

```
1        stone bridge
```

Practica estos comandos: 0, ^

Pistas: 0 llega a la primera columna aunque contenga un espacio. ^ salta los espacios iniciales. Pulsa 0 y después ^. El objetivo es la primera letra, pero aquí aprenderás ambos comienzos.

Una ruta posible

`0^`

Misión 08 / La otra orilla

Salta a la última línea con G y llega a su final con $.

Inicio: línea 1, columna 1

Destino del cursor: línea 3, columna 16.

```
1     a quiet river
2     under the bridge
3     the last lantern
```

Practica estos comandos: G, $

Pistas: G lleva a la última línea del registro. $ lleva al último carácter de la línea actual. Combina los dos movimientos: G y después $. No hace falta bajar línea a línea.

Una ruta posible

`G$`

Misión 09 / Una zancada larga

Vuelve arriba con gg, baja tres líneas con 3j y elige la primera columna con 0.

Inicio: línea 5, columna 5

Destino del cursor: línea 4, columna 1.

```
1     north gate
2     mist
3     old steps
4     star bridge
5     last lantern
```

Practica estos comandos: gg, count, 0

Pistas: gg lleva a la primera línea. El número 3 antes de j repite el movimiento tres veces. Utiliza gg, después 3j y, por último, 0. El número forma parte de la orden, no del texto.

Una ruta posible

```
gg3j0
```

04 / Observatorio del Eco

Encuentra lo que no se ve a primera vista

Las estrellas siguen brillando sobre la isla, pero sus nombres están dispersos por las tablillas. Orin escucha cómo vuelven sus ecos. Aprende a buscar caracteres y palabras para leer el camino que esconden.

Misión 10 / Una estrella elegida

Encuentra la letra l en la línea actual con f.

Inicio: línea 1, columna 1

Destino del cursor: línea 1, columna 8.

```
1     moss | lantern | stars
```

Practica estos comandos: f

Pistas: f espera el carácter que quieres buscar. En esta misión es l. Pulsa f y después l. Llegarás a la siguiente l sin pasar por cada letra.

Una ruta posible

```
fl
```

Misión 11 / El eco de un carácter

Encuentra la siguiente b con fb y repite la búsqueda con el punto y coma ;.

Inicio: línea 1, columna 1

Destino del cursor: línea 1, columna 11.

```
1     bell bell bell
```

Practica estos comandos: f, ;

Pistas: Pulsa b después de f. ; repite la última búsqueda de un solo carácter. Usa fb y después ; para llegar a otra b de la misma línea.

Una ruta posible

```
fb;
```

Misión 12 / El eco que vuelve

Busca light con /light y Enter. Usa n, N y de nuevo n para llegar al eco de la última tablilla.

Inicio: línea 1, columna 1

Destino del cursor: línea 3, columna 10.

```
1     light sleeps
2     follow light
3     wake the light
```

Practica estos comandos: /, n, N

Pistas: n busca otro resultado en la misma dirección; N busca en la contraria. Escribe /light, pulsa Enter y después usa n, N, n. En esta misión practica ambas direcciones.

Una ruta posible

```
/light → <Enter> → nNn
```

05 / Forja de las Brasas

Repara sin miedo. Los cambios se pueden deshacer

En el hogar queda calor suficiente para forjar nuevas letras. Asha no exige movimientos perfectos: quien conoce su oficio también sabe deshacer un error. Limpia los registros dañados y reaviva el fuego.

Misión 13 / Una astilla en el registro

Elimina un carácter sobrante con x y restaura el texto.

Inicio: línea 1, columna 6

Texto inicial

```
1     sparkx
```

Texto objetivo

```
1     spark
```

Practica estos comandos: x

Pistas: x borra el carácter bajo el cursor. Coloca primero el cursor sobre la letra que sobra. Después de borrarla, examina el texto completo.

Una ruta posible

x

Misión 14 / La memoria del metal

Sustituye la primera c por g con rg, deshaz el cambio con u y rehazlo con Ctrl-r.

Inicio: línea 1, columna 1

Texto inicial

```
1     cold ember
2     cold ember
```

Texto objetivo

```
1     gold ember
2     cold ember
```

Practica estos comandos: r, u, Ctrl-r

Pistas: r espera un carácter de sustitución. rg convierte cold en gold. Después de rg, usa u y Ctrl-r. El objetivo es gold ember en la primera línea y cold ember en la segunda.

Una ruta posible

```
rg → u → <C-r>
```

Misión 15 / Una forma nueva

Borra el primer broken con dw y corta la línea dañada que queda con dd. Deja solo dawn.

Inicio: línea 1, columna 1

Texto inicial

```
1     broken broken
2     dawn
```

Texto objetivo

```
1     dawn
```

Practica estos comandos: dw, dd

Pistas: dw borra hacia la siguiente palabra. dd borra toda la línea actual. Primero dw y después dd: dos tamaños de daño, dos herramientas diferentes.

Una ruta posible

`dwdd`

06 / Biblioteca de los Espejos

Una buena idea merece un eco

Los libros de estas estanterías reflejan sus páginas en el agua. Mira sabe conservar una palabra antes de que se desvanezca. Con un registro podrás copiar un poco de luz o trasladar una línea completa.

Misión 16 / Una línea gemela

Copia una línea con yy y coloca su reflejo debajo con p.

Inicio: línea 1, columna 1

Texto inicial

```
1     light
2     sleep
```

Texto objetivo

```
1     light
2     light
3     sleep
```

Practica estos comandos: yy, p

Pistas: yy copia toda la línea actual sin quitar el original. Después de copiar una línea, p la pega debajo de la actual. Primero elige dónde debe aparecer la copia.

Una ruta posible

`yyp`

Misión 17 / Una palabra en el espejo

Conserva una palabra con yw y colócala en el lugar correcto con p.

Inicio: línea 1, columna 1

Texto inicial

```
1     glow moss
```

Texto objetivo

```
1     glow glow moss
```

Practica estos comandos: yw, p

Pistas: yw copia el texto hacia el principio de la palabra siguiente. Si has copiado un fragmento, p lo pega después del cursor. No se comporta igual que al pegar una línea entera.

Una ruta posible

`ywwhp`

Misión 18 / Un orden nuevo

Corta una línea con dd y muévela con p para recuperar el orden correcto.

Inicio: línea 1, columna 1

Texto inicial

```
1   mist
2   sun
3   sky
```

Texto objetivo

```
1   sun
2   mist
3   sky
```

Practica estos comandos: dd, p

Pistas: dd guarda la línea eliminada en el mismo registro que usa p. Tras cortar, ve al destino. p coloca la línea recordada debajo de la actual.

Una ruta posible

`ddp`

07 / Escritorio del Alba

Deja un nuevo registro a tu paso

Eno ha encontrado huecos en la crónica. Esas palabras no volverán solo con movimientos: alguien tiene que escribirlas. Aquí aprenderás el ritmo de alternar entre actuar sobre el texto e introducirlo.

Misión 19 / La palabra que falta

Coloca el cursor, entra en el modo de inserción con i y escribe el texto que falta.

Inicio: línea 1, columna 5

Texto inicial

```
1     the returns
```

Texto objetivo

```
1     the dawn returns
```

Practica estos comandos: i, Esc

Pistas: i empieza a insertar antes del cursor. En el modo de inserción, las letras se convierten en texto. Esc devuelve el modo Normal y sus movimientos.

Una ruta posible

`i → dawn<Space> → <Esc>`

Misión 20 / Espacio para más

Añade s con a, vuelve con Esc y después agrega un espacio y shine al final de la línea con A.

Inicio: línea 1, columna 4

Texto inicial

```
1     star
```

Texto objetivo

```
1     stars shine
```

Practica estos comandos: a, A, Esc

Pistas: a empieza a insertar después del cursor. A lo hace al final de toda la línea. Convierte star en stars con a, s, Esc. Después A, espacio, shine, Esc producen stars shine.

Una ruta posible

`as → <Esc> → A<Space>shine → <Esc>`

Misión 21 / El sello de la crónica

Abre una línea nueva con o, escribe returns, sal con Esc y guarda con :w y Enter.

Inicio: línea 1, columna 1

Texto inicial

```
1  dawn
```

Texto objetivo

```
1  dawn
2  returns
```

Practica estos comandos: o, Esc, :w

Pistas: o abre una línea debajo y entra directamente en el modo de inserción. Escribe returns debajo de dawn. Después Esc, :w y Enter sellan la tablilla completa.

Una ruta posible

`oreturns → <Esc> → :w → <Enter>`

08 / Torre de la Mañana

Todos los caminos conducen a la luz

En la cumbre espera una lente hecha con fragmentos de las primeras islas. Reúne tus habilidades para llamar al primer amanecer. Su luz revelará más regiones y el camino hacia la maestría.

Misión 22 / Un mapa en la memoria

Corta la primera línea dañada con dd, copia la buena con yy, pégala con p y confirma con :w.

Inicio: línea 1, columna 1

Texto inicial

```
1     lost lost
2     keep the light
```

Texto objetivo

```
1     keep the light
2     keep the light
```

Practica estos comandos: dd, yy, p, :w

Pistas: Después de dd queda keep the light. yy conserva toda esa línea y p crea una copia debajo. Usa dd, yy, p, después :w y Enter. Deben quedar dos líneas idénticas de keep the light.

Una ruta posible

`ddyyp:w → <Enter>`

Misión 23 / La última reparación

Busca sleep con /sleep y Enter. Bórralo con dw, añade sing con A y guarda con :w.

Inicio: línea 1, columna 1

Texto inicial

```
1     moss covers dawn
2     the bells sleep
```

Texto objetivo

```
1     moss covers dawn
2     the bells sing
```

Practica estos comandos: /, dw, A, :w

Pistas: La búsqueda te coloca en sleep. dw elimina esa palabra y A permite terminar la línea. No cambies la primera línea. La segunda debe decir the bells sing. Sal con Esc y termina con :w y Enter.

Una ruta posible

`/sleep → <Enter> → dwAsing → <Esc> → :w → <Enter>`

Misión 24 / Que vuelva la luz

Elimina la primera línea, copia la que queda y añade the dawn returns en una línea nueva. Guarda la crónica restaurada.

Inicio: línea 1, columna 1

Texto inicial

```
1     the night stays
2     one small spark
```

Texto objetivo

```
1     one small spark
2     one small spark
3     the dawn returns
```

Practica estos comandos: dd, yy, p, o, :w

Pistas: dd elimina la línea dañada; yy y p crean otro one small spark. o abre espacio para la última línea. Después de dd, yy, p, usa o y escribe the dawn returns. Esc, :w y Enter llamarán al primer amanecer y abrirán el camino siguiente.

Una ruta posible

`ddyypothe<Space>dawn<Space>returns → <Esc> → :w → <Enter>`

09 / Acantilados del Viento

Cuando una palabra es algo más

El viento transporta cintas de escritura rotas. Tavi te enseña a cruzar fragmentos sin espacios y volver sobre ellos sin contar letras.

Misión 25 / Una cinta continua

Salta todo one-two con W y llega al principio de three.four. Órdenes necesarias: W.

Inicio: línea 1, columna 1

Destino del cursor: línea 1, columna 9.

```
1     one-two three.four five
```

Practica estos comandos: W

Pistas: Ve al siguiente WORD separado por espacios; la puntuación sigue dentro. Haz primero un paso consciente y comprueba el cursor o texto. Añade repeticiones cuando comprendas una operación.

Una ruta posible

```
W
```

Misión 26 / Volver por la cresta

Compara E y B con ge y gE y vuelve después a la primera columna. Órdenes necesarias: E · B · ge · gE.

Inicio: línea 1, columna 23

Destino del cursor: línea 1, columna 1.

```
1     one-two three.four five
```

Practica estos comandos: E, B, ge, gE

Pistas: Ve al final de un WORD separado por espacios. Retrocede al principio de un WORD; la puntuación no lo divide.

Una ruta posible

```
BEgegE0
```

Misión 27 / Pasos medidos

Practica + y -, añade un número al movimiento y usa la columna para llegar a la t de third. Órdenes necesarias: + · - · | · cantidad.

Inicio: línea 1, columna 4

Destino del cursor: línea 3, columna 2.

```
1      first
2        second
3      third
```

Practica estos comandos: +, -, |, count

Pistas: Ve al primer carácter no vacío de la línea siguiente; admite un número. Ve al primer carácter no vacío de la línea anterior.

Una ruta posible

```
+-2+2|
```

10 / Paso de los Rastreadores

Mira también hacia atrás

Unas muescas señalan el paso. Ryn te enseña a buscar en ambos sentidos y detenerte justo donde empieza la siguiente piedra segura.

Misión 28 / Contra el rastro

Busca a hacia atrás, compara t y T e invierte con la coma hasta quedar justo antes de la última a. Órdenes necesarias: F · t · T · ,.

Inicio: línea 1, columna 13

Destino del cursor: línea 1, columna 12.

```
1     a x a y a z a
```

Practica estos comandos: F, t, T, ,

Pistas: Pulsa F y un carácter para buscarlo hacia atrás en esta línea. Avanza hasta justo antes de la siguiente aparición de un carácter.

Una ruta posible

`FataTa,`

Misión 29 / Una palabra responde

Sigue la palabra completa glow en ambos sentidos con * y #. Termina en el segundo glow de la primera línea. Órdenes necesarias: * · #.

Inicio: línea 1, columna 1

Destino del cursor: línea 1, columna 11.

```
1     glow moss glow
2     glow
```

Practica estos comandos: *, #

Pistas: Busca hacia delante la palabra completa bajo el cursor. Busca hacia atrás la palabra completa bajo el cursor.

Una ruta posible

#

Misión 30 / El camino inverso

Busca star hacia atrás con ?, practica n y N y termina en la aparición central. Órdenes necesarias: ? · n · N.

Inicio: línea 3, columna 1

Destino del cursor: línea 2, columna 1.

```
1     star
2     star
3     star
```

Practica estos comandos: ?, n, N

Pistas: Abre una búsqueda hacia atrás y confirma con Enter. Repite la búsqueda en la dirección que ya tenía.

Una ruta posible

`?star → <Enter> → nN`

11 / Círculos de Piedra

Cada apertura tiene un cierre

Uma custodia círculos dentro de otros círculos. Lee los delimitadores, las oraciones y los espacios de los párrafos hasta descubrir el orden del laberinto.

Misión 31 / Piedras emparejadas

Explora los delimitadores anidados con % y los límites de bloque. Vuelve a la llave exterior de apertura. Órdenes necesarias: % · [(·]) · [{ ·]}.

Inicio: línea 2, columna 5

Destino del cursor: línea 1, columna 1.

```
1     {
2       [a(b)c]
3     }
```

Practica estos comandos: %, [(,]), [{,]}

Pistas: Salta al delimitador correspondiente (), [] o {}. Retrocede al paréntesis de apertura envolvente.

Una ruta posible

`% → [( → ]) → [{ → ]} → %`

Misión 32 / Entre párrafos

Avanza y retrocede entre párrafos. Detente en la línea vacía entre north y south. Órdenes necesarias: { · }.

Inicio: línea 1, columna 1

Destino del cursor: línea 3, columna 1.

```
1     north
2     path
3     [línea vacía]
4     south
5     gate
6     [línea vacía]
7     dawn
```

Practica estos comandos: {, }

Pistas: Retrocede al límite de un párrafo; las líneas vacías los separan. Avanza al siguiente límite de párrafo.

Una ruta posible

`}}{`

Misión 33 / Espacio para respirar

Visita principios de oraciones en ambos sentidos y vuelve al principio de Second. Órdenes necesarias: (·).

Inicio: línea 1, columna 1

Destino del cursor: línea 1, columna 14.

```
1     First light. Second light! Third light?
```

Practica estos comandos: (,)

Pistas: Retrocede al principio de una oración. Avanza al principio de la siguiente oración.

Una ruta posible

```
)))(
```

12 / Torre del Horizonte

Cambia la vista, no el texto

Las ventanas de la torre recorren un pergamino. Noa muestra la diferencia entre la posición del cursor y el fragmento que permite ver una ventana.

Misión 34 / La ventana sube

Prueba las tres posiciones de ventana. Termina con la línea actual, la quinta, en el borde superior. Órdenes necesarias: zt · zz · zb.

Inicio: línea 5, columna 1

Destino del cursor: línea 5, columna 1.

```
1     line one
2     line two
3     line three
4     line four
5     line five
6     line six
7     line seven
8     line eight
9     line nine
```

Primera línea visible: 5

Practica estos comandos: zt, zz, zb

Pistas: Desplaza la vista para dejar la línea actual arriba. Centra la línea actual sin modificar el texto.

Una ruta posible

```
ztzzzbzt
```

Misión 35 / Un horizonte estable

Centra la quinta línea, visita los extremos de la ventana y vuelve a su centro. Órdenes necesarias: H · M · L.

Inicio: línea 5, columna 1

Destino del cursor: línea 5, columna 1.

```
1     line one
2     line two
3     line three
4     line four
5     line five
6     line six
7     line seven
8     line eight
9     line nine
```

Primera línea visible: 3

Practica estos comandos: H, M, L

Pistas: Ve a la primera línea visible de la ventana. Ve a la línea visible del centro de la ventana.

Una ruta posible

`zzHLM`

Misión 36 / Números a tu lado

Activa, desactiva y alterna ambos tipos de números. Deja los absolutos activados y los relativos desactivados. Órdenes necesarias: nu · nonu · nu! · rnu · nornu · rnu!.

Inicio: línea 1, columna 1

Destino del cursor: línea 1, columna 1.

```
1     north
2     south
```

Números absolutos: activados

Números relativos: desactivados

Practica estos comandos: nu, nonu, nu!, rnu, nornu, rnu!

Pistas: Activa los números de línea absolutos. Desactiva los números de línea absolutos.

Una ruta posible

`:set<Space>nonu → <Enter> → :set<Space>nu → <Enter> → :set<Space>nu! → <Enter> → :set<Space>nu! → <Enter> → :set<Space>rnu → <Enter> → :set<Space>nornu → <Enter> → :set<Space>rnu! → <Enter> → :set<Space>rnu! → <Enter>`

13 / Taller de Tejedores

Corta, añade y cose

Ada repara telas tejidas con letras. A veces debes alcanzar antes del cursor, otras subir una línea y otras empezar de nuevo toda la puntada.

Misión 37 / Antes del hilo

Borra la x anterior al cursor, convierte DAWN a minúsculas y recorta junk. Conserva el espacio tras dawn. Órdenes necesarias: X · ~ · D.

Inicio: línea 1, columna 2

Texto inicial

```
1     xDAWN junk
```

Texto objetivo

```
1     dawn
```

Practica estos comandos: X, ~, D

Pistas: Borra el carácter anterior al cursor. Invierte la mayúscula o minúscula bajo el cursor.

Una ruta posible

`X4~wD`

Misión 38 / Corte y costura

Usa I para añadir the antes de glow conservando la sangría. Usa O para crear header en una línea superior. Órdenes necesarias: I · O.

Inicio: línea 1, columna 7

Texto inicial

```
1        glow
```

Texto objetivo

```
1     header
2        the glow
```

Practica estos comandos: I, O

Pistas: Inserta en el primer carácter no vacío de la línea. Abre una línea encima y entra en inserción.

Una ruta posible

`Ithe<Space> → <Esc> → Oheader → <Esc>`

Misión 39 / Una capa nueva

Sustituye x por sun con s y toda la segunda línea por moon con S. Órdenes necesarias: s · S.

Inicio: línea 1, columna 1

Texto inicial

```
1     x
2     bad line
```

Texto objetivo

```
1     sun
2     moon
```

Practica estos comandos: s, S

Pistas: Borra el carácter bajo el cursor y empieza a insertar. Sustituye la línea actual entrando en inserción.

Una ruta posible

`ssun → <Esc> → jSmoon → <Esc>`

14 / Fundición de Gramática

Una acción y su alcance

Keris funde órdenes de dos piezas: qué hacer y sobre qué. Combina operador y movimiento para descubrir el sentido de una expresión breve.

Misión 40 / Un verbo y un camino

Compara el cambio de palabra con c, de línea con cc y de final con C. Obtén warm sun y fixed. Órdenes necesarias: c · cc · C.

Inicio: línea 1, columna 1

Texto inicial

```
1     cold moss
2     broken
```

Texto objetivo

```
1     warm sun
2     fixed
```

Practica estos comandos: c, cc, C

Pistas: Operador de cambio: elimina el rango e inicia la inserción. Cambia toda la línea actual e inicia la inserción.

Una ruta posible

`cewarm → <Esc> → jccfixed → <Esc> → ggwCsun → <Esc>`

Misión 41 / Conservar y pegar

Copia una palabra con y y pégala antes del cursor con P. Practica también Y y el borrado sin perder north. Órdenes necesarias: y · Y · P · d.

Inicio: línea 1, columna 1

Texto inicial

```
1     red blue
2     north
```

Texto objetivo

```
1     red red blue
2     north
```

Practica estos comandos: y, Y, P, d

Pistas: Operador de copia: combínalo con un movimiento u objeto. Copia toda la línea actual; equivale a yy en este juego.

Una ruta posible

`ywwPYpdd`

Misión 42 / Una vez más

Borra el primer bad con dw y repite el cambio para los otros dos. Deja solo light. Órdenes necesarias: dw · ..

Inicio: línea 1, columna 1

Texto inicial

```
1     bad bad bad light
```

Texto objetivo

```
1     light
```

Practica estos comandos: dw, .

Pistas: Borra desde el cursor hasta el comienzo de la palabra siguiente. Repite el último cambio, incluido el texto insertado.

Una ruta posible

```
dw..
```

15 / Santuario de Palabras

Elige sentido, no distancia

Eli no mide el texto contando caracteres. Aquí las palabras, oraciones y párrafos son recipientes preparados para tu intención.

Misión 43 / El corazón de una palabra

Compara iw y aw: elimina el interior de moss y después blue con su espacio adyacente. Deja red. Órdenes necesarias: iw · aw.

Inicio: línea 1, columna 6

Texto inicial

```
1      red moss blue
```

Texto objetivo

```
1      red
```

Practica estos comandos: iw, aw

Pistas: La palabra interior sin espacios alrededor; combina con d, y o c. Una palabra con espacio adyacente; objeto para un operador.

Una ruta posible

```
diwwdaw
```

Misión 44 / Una palabra mayor

Trata moss.green como un solo WORD. Practica iW y aW hasta dejar solo red. Órdenes necesarias: iW · aW.

Inicio: línea 1, columna 7

Texto inicial

```
1      red moss.green blue
```

Texto objetivo

```
1      red
```

Practica estos comandos: iW, aW

Pistas: El interior de un WORD: texto sin espacios, incluida puntuación. Un WORD con el espacio en blanco adyacente.

Una ruta posible

```
diWwdaW
```

Misión 45 / Oraciones y párrafos

Copia el interior y toda una oración, luego el interior de un párrafo. Elimina alpha/beta con su separación sin quitar las oraciones ni keep. Órdenes necesarias: is · as · ip · ap.

Inicio: línea 1, columna 1

Texto inicial

```
1     One. Two.
2     [línea vacía]
3     alpha
4     beta
5     [línea vacía]
6     keep
```

Texto objetivo

```
1     One. Two.
2     [línea vacía]
3     keep
```

Practica estos comandos: is, as, ip, ap

Pistas: La oración interior sin el espacio que la separa. Una oración con el espacio que la separa.

Una ruta posible

`yisyasjjyipdap`

16 / Bóveda de Delimitadores

El interior y su marco

Sen guarda cajas dentro de otras cajas. Elige el contenido o el marco completo: distintos nombres de una llave pueden abrir la misma cerradura.

Misión 46 / Cuadrado y círculo

Copia interiores y marcos de [] y () con todos los alias indicados. Termina con (glow) completo en el registro sin nombre. Órdenes necesarias: i[· a[· i] · a] · i(· a(· i) · a) · ib · ab.

Inicio: línea 1, columna 3

Destino del cursor: línea 1, columna 8.

```
1     [moss] (glow)
```

Registro ""

```
1     (glow)
```

Practica estos comandos: i[, a[, i], a], i(, a(, i), a), ib, ab

Pistas: Interior de un par [], sin corchetes; ambos alias equivalen. Un par [] completo, incluidos los corchetes.

Una ruta posible

```
yi[ya[yi]ya]f(yi(ya(yi)ya)yibyab
```

Misión 47 / Ángulos y alias

Compara todos los alias de {} y <>. Conserva al final <dawn> completo, incluido su marco, en el registro. Órdenes necesarias: i{ · a{ · i} · a} · iB · aB · i< · a< · i> · a>.

Inicio: línea 1, columna 3

Destino del cursor: línea 1, columna 9.

```
1     {ember} <dawn>
```

Registro ""

```
1     <dawn>
```

Practica estos comandos: i{, a{, i}, a}, iB, aB, i<, a<, i>, a>

Pistas: Interior de un par {}, sin llaves; iB es un alias. Un par {} completo, incluidas las llaves; aB es un alias.

Una ruta posible

```
yi{ya{yi}ya}yiByaBf<yi<ya<yi>ya>
```

Misión 48 / El marco completo

Cambia solo el interior de (seed) por bloom y conserva los paréntesis. Después elimina todo [noise]. Órdenes necesarias: i(· a[.

Inicio: línea 1, columna 11

Texto inicial

```
1     [noise] (seed) {ash} <sky>
```

Texto objetivo

```
1      (bloom) {ash} <sky>
```

Practica estos comandos: i(, a[

Pistas: Interior de un par (), sin paréntesis; ib es un alias. Un par [] completo, incluidos los corchetes.

Una ruta posible

`ci(bloom → <Esc> → 0da[`

17 / Capilla de los Susurros

Escucha entre los símbolos

Yuna conserva mensajes entre comillas y etiquetas. Para escucharlos, aprende a distinguir la voz del marco que la contiene.

Misión 49 / Dos voces suaves

Compara los objetos interiores y completos de comillas dobles y simples. Termina copiando todo glow con sus comillas simples. Órdenes necesarias: i" · a" · i' · a'.

Inicio: línea 1, columna 3

Destino del cursor: línea 1, columna 8.

```
1     "moss" 'glow'
```

Registro ""

```
1     'glow'
```

Practica estos comandos: i", a", i', a'

Pistas: Texto dentro de comillas dobles, sin las comillas. Texto con las comillas dobles que lo rodean.

Una ruta posible

```
yi"ya"f'yi'ya'
```

Misión 50 / Un susurro y una etiqueta

Copia dentro y alrededor de acentos graves y después de un elemento HTML. Conserva todo glow. Órdenes necesarias: i` · a` · it · at.

Inicio: línea 1, columna 3

Destino del cursor: línea 1, columna 8.

```
1     `seed` <b>glow</b>
```

Registro ""

```
1     <b>glow</b>
```

Practica estos comandos: i`, a`, it, at

Pistas: Texto dentro de acentos graves, sin delimitadores. Texto con los acentos graves que lo rodean.

Una ruta posible

```
yi`ya`f<yityat
```

Misión 51 / Un mensaje anidado

Cambia solo el contenido del elemento p por new, sin quitar las etiquetas. Elimina todo noise entre comillas y conserva el espacio final. Órdenes necesarias: it · a".

Inicio: línea 1, columna 5

Texto inicial

```
1     <p>old</p> "noise"
```

Texto objetivo

```
1     <p>new</p>
```

Practica estos comandos: it, a"

Pistas: Interior de una pareja de etiquetas HTML, sin las etiquetas. Texto con las comillas dobles que lo rodean.

Una ruta posible

`citnew → <Esc> → f"da"`

18 / Campamento de Cartógrafos

Deja un lugar al que volver

Aro dibuja mapas que el viento no puede dispersar. Recuerda una posición, vuelve a su línea o columna exacta y borra los hitos que ya no necesitas.

Misión 52 / Crear un hito

Crea la marca a en la segunda línea. Compara el salto de línea con el de columna exacta y vuelve a la posición guardada. Órdenes necesarias: m · ' · `.

Inicio: línea 2, columna 5

Destino del cursor: línea 2, columna 5.

```
1       north
2       south
3       dawn
```

Marca a: línea 2, columna 5

Practica estos comandos: m, ', `

Pistas: Crea una marca: m y una letra, como ma, recuerdan la posición. Comilla simple y letra de marca saltan al primer carácter de su línea.

Una ruta posible

`magg'agg`a`

Misión 53 / Leer el mapa

Crea a y b en columnas contiguas, abre la lista de marcas y elimina solo a con la abreviatura :delm. Órdenes necesarias: :marks · :delm.

Inicio: línea 1, columna 1

Destino del cursor: línea 1, columna 2.

```
1     path
```

Marca b: línea 1, columna 2

Marcas eliminadas: a

Lista abierta: Marcas

Practica estos comandos: :marks, :delm

Pistas: Muestra las marcas guardadas. Elimina marcas indicadas, como :delmarks a; :delm es la abreviatura.

Una ruta posible

`malmb → :marks → <Enter> → :delm<Space>a → <Enter>`

Misión 54 / Borrar rutas antiguas

Practica el borrado de un rango de marcas, :delmarks! y :delm!. Termina sin a, b ni c. Órdenes necesarias: :delmarks · :delmarks! · :delm!.

Inicio: línea 1, columna 1

Destino del cursor: línea 1, columna 2.

```
1     path
```

Marcas eliminadas: a, b, c

Practica estos comandos: :delmarks, :delmarks!, :delm!

Pistas: Elimina marcas indicadas, como :delmarks a; :delm es la abreviatura. Elimina todas las marcas locales.

Una ruta posible

`malmb → :delmarks<Space>a-b → <Enter> → ma → :delmarks! → <Enter> → mc → :delm! → <Enter>`

19 / Gabinete de Ecos

Un portapapeles puede tener nombre

Esme conserva ecos en cajones separados. Guarda uno, amplía otro y deja desaparecer el ruido sin perder lo importante.

Misión 55 / Un eco sin nombre

Borra a, recupérala del registro de borrado pequeño y copia todo ab mediante el registro sin nombre explícito. Órdenes necesarias: "- · "" · x · yy · P.

Inicio: línea 1, columna 1

Texto objetivo

```
1	ab
```

Registro ""

```
1	ab
```

Registro "-

```
1	a
```

Practica estos comandos: "-, "", x, yy, P

Pistas: Registro de borrado pequeño: texto menor que una línea completa. El registro sin nombre guarda el último texto copiado o borrado.

Una ruta posible

`x"-P""yy`

Misión 56 / Añadir sin sobrescribir

Guarda moss en a, añade glow mediante A y pega ambas líneas. Revisa el registro a. Órdenes necesarias: "a · "A · p.

Inicio: línea 1, columna 1

Texto inicial

```
1     moss
2     glow
```

Texto objetivo

```
1     moss
2     moss
3     glow
4     glow
```

Registro "a

```
1     moss
2     glow
```

Practica estos comandos: "a, "A, p

Pistas: Registro a-z: elige su letra antes del operador para conservar texto. A-Z añade al registro a-z correspondiente en lugar de sustituirlo.

Una ruta posible

```
"ayyj"Ayygg"ap
```

Misión 57 / El silencio protege la copia

Conserva keep en a, borra trash al agujero negro y pega la copia guardada encima de la línea actual. Órdenes necesarias: "_ · "a · dd · P.

Inicio: línea 1, columna 1

Texto inicial

```
1     keep
2     trash
```

Texto objetivo

```
1     keep
2     keep
```

Registro "a

```
1     keep
```

Registro ""

```
1     keep
```

Practica estos comandos: "_, "a, dd, P

Pistas: Agujero negro: borra sin sobrescribir otros registros. Registro a-z: elige su letra antes del operador para conservar texto.

Una ruta posible

```
"ayyj"_dd"aP
```

20 / Cámara de los Nueve

La memoria tiene un orden

Nona recuerda más que la última desaparición. Abre los cajones del historial y descubre que borrar no siempre significa despedirse.

Misión 58 / La última copia

Copia light, borra dust y recupera la copia del registro 0. El registro 1 debe seguir recordando dust. Órdenes necesarias: "0 · dd.

Inicio: línea 1, columna 1

Texto inicial

```
1  light
2  dust
```

Texto objetivo

```
1  light
2  light
```

Registro "0

```
1  light
```

Registro "1

```
1  dust
```

Practica estos comandos: "0, dd

Pistas: El registro 0 conserva la última copia incluso tras un borrado. Corta la línea actual y consérvala en un registro.

Una ruta posible

```
yyjdd"0p
```

Misión 59 / Una cadena de despedidas

Borra nueve líneas y revisa los registros 1 a 9. Restaura el one más antiguo encima de keep. Órdenes necesarias: "1 · "2 · "3 · "4 · "5 · "6 · "7 · "8 · "9.

Inicio: línea 1, columna 1

Texto inicial

```
1     one
2     two
3     three
4     four
5     five
6     six
7     seven
8     eight
9     nine
10    keep
```

Texto objetivo

```
1     one
2     keep
```

Registro "1

```
1     nine
```

Registro "2

```
1     eight
```

Registro "3

```
1     seven
```

Registro "4

```
1     six
```

Registro "5

```
1     five
```

Registro "6

```
1     four
```

Registro "7

```
1     three
```

Registro "8

```
1     two
```

Registro "9

```
1     one
```

Practica estos comandos: "1, "2, "3, "4, "5, "6, "7, "8, "9

Pistas: Historial de borrados grandes: 1 es el más reciente y 9 el más antiguo conservado. Historial de borrados grandes: 1 es el más reciente y 9 el más antiguo conservado.

Una ruta posible

`dd........ → "1pu → "2pu → "3pu → "4pu → "5pu → "6pu → "7pu → "8pu → "9P`

Misión 60 / Leer los cajones

Copia one en a y abre el inspector de registros. Comprueba que a y 0 conservan la misma línea. Órdenes necesarias: :reg.

Inicio: línea 1, columna 1

Destino del cursor: línea 1, columna 1.

```
1	one
2	two
```

Registro "0

```
1	one
```

Registro "a

```
1	one
```

Lista abierta: Registros

Practica estos comandos: :reg

Pistas: Muestra registros; puedes indicar nombres, como :reg a0. Haz primero un paso consciente y comprueba el cursor o texto. Añade repeticiones cuando comprendas una operación.

Una ruta posible

`"ayy → :reg → <Enter>`

21 / Puerto de Escribas

Muchas páginas, un solo viaje

Tero lleva documentos entre puertos virtuales. Guarda lo que quieras conservar y cambia de página sin perder de vista sus nombres.

Misión 61 / Una página recordada

Guarda dawn rises en notes.txt, cambia echo.txt a awake y guárdalo. Vuelve a notes.txt sin perder cambios. Órdenes necesarias: :w · :e · c.

Inicio: línea 1, columna 1

Texto inicial

```
1      dawn
```

Texto objetivo

```
1      dawn rises
```

Búfer activo: notes.txt

Archivo virtual guardado: notes.txt

```
1      dawn rises
```

Archivo virtual guardado: echo.txt

```
1      awake
```

Practica estos comandos: :w, :e, c

Pistas: Guarda el documento virtual; un nombre opcional indica el archivo del juego. Abre un archivo virtual con el nombre indicado.

Una ruta posible

```
A<Space>rises → <Esc> → :w → <Enter> → :e<Space>echo.txt → <Enter> → ccawake → <Esc> → :w → <Enter> → :e<Space>notes.txt → <Enter>
```

Misión 62 / Otro puerto

Abre beta.txt, muestra los búferes y vuelve por nombre a alpha.txt. Órdenes necesarias: :e · :ls · :b.

Inicio: línea 1, columna 1

Destino del cursor: línea 1, columna 1.

```
1      north
```

Búfer activo: alpha.txt

Lista abierta: Búferes

Practica estos comandos: :e, :ls, :b

Pistas: Abre un archivo virtual con el nombre indicado. Muestra los búferes abiertos y sus números.

Una ruta posible

```
:e<Space>beta.txt → <Enter> → :ls → <Enter> → :b<Space>alpha.txt → <Enter>
```

Misión 63 / Cruzar entre búferes

Guarda una copia como final.txt, ábrela y compara el cambio de búfer por número. Termina en final.txt. Órdenes necesarias: :w · :e · :b.

Inicio: línea 1, columna 1

Texto objetivo

```
1     light
```

Búfer activo: final.txt

Archivo virtual guardado: final.txt

```
1     light
```

Practica estos comandos: :w, :e, :b

Pistas: Guarda el documento virtual; un nombre opcional indica el archivo del juego. Abre un archivo virtual con el nombre indicado.

Una ruta posible

```
:w<Space>final.txt → <Enter> → :e<Space>final.txt → <Enter> → :b<Space>1 → <Enter> → :b<Space>2 → <Enter>
```

22 / Huerto Mecánico

Que un buen movimiento dé muchos frutos

Omi cuida los delicados mecanismos del huerto. Graba un ciclo fiable, comprueba dónde termina y deja que trabaje en las siguientes ramas.

Misión 64 / Una máquina pequeña

Graba en a la adición de ! y el movimiento hacia abajo. Reproduce una vez para cambiar las dos primeras líneas dawn. Órdenes necesarias: q · @.

Inicio: línea 1, columna 1

Texto inicial

```
1	dawn
2	dawn
3	dawn
```

Texto objetivo

```
1	dawn!
2	dawn!
3	dawn
```

Macros grabadas: a

Practica estos comandos: q, @

Pistas: q y una letra inician una macro; otra q termina la grabación. Reproduce la macro de un registro, como @a; admite repeticiones.

Una ruta posible

`qaA! → <Esc> → jq@a`

Misión 65 / El huerto se repite

Graba añadir + al principio y bajar una línea. Usa @a y @@ y deja intacto el cuarto seed. Órdenes necesarias: q · @ · @@.

Inicio: línea 1, columna 1

Texto inicial

```
1      seed
2      seed
3      seed
4      seed
```

Texto objetivo

```
1      +seed
2      +seed
3      +seed
4      seed
```

Macros grabadas: a

Practica estos comandos: q, @, @@

Pistas: q y una letra inician una macro; otra q termina la grabación. Reproduce la macro de un registro, como @a; admite repeticiones.

Una ruta posible

`qaI+ → <Esc> → jq@a@@`

Misión 66 / Otra vez, con precisión

Graba borrar una x y reproduce la macro tres veces. Deja solo light. Órdenes necesarias: q · @ · cantidad.

Inicio: línea 1, columna 1

Texto inicial

```
1      xxxxlight
```

Texto objetivo

```
1      light
```

Macros grabadas: a

Practica estos comandos: q, @, count

Pistas: q y una letra inician una macro; otra q termina la grabación. Reproduce la macro de un registro, como @a; admite repeticiones.

Una ruta posible

`qaxq3@a`

23 / Pruebas de Tejedores

Combina los patrones aprendidos

Vel extiende un patrón largo. Los números, las ediciones repetidas y las macros son herramientas distintas: elige la que expresa tu intención.

Misión 67 / Dos contadores

Combina las cantidades de operador y movimiento: borra seis palabras con 2d3w y conserva keep. Órdenes necesarias: cantidad · dw.

Inicio: línea 1, columna 1

Texto inicial

```
1      one two three four five six keep
```

Texto objetivo

```
1      keep
```

Practica estos comandos: count, dw

Pistas: Un número ante operador o movimiento indica repeticiones; 2d3w combina ambos. Borra desde el cursor hasta el comienzo de la palabra siguiente.

Una ruta posible

`2d3w`

Misión 68 / El mismo tejido

Borra x, deshaz y rehace con :undo y :redo y repite después el borrado con el punto. Órdenes necesarias: :undo · :redo · ..

Inicio: línea 1, columna 1

Texto inicial

```
1      xxdawn
```

Texto objetivo

```
1      dawn
```

Practica estos comandos: :undo, :redo, .

Pistas: Deshaz el último cambio; forma de u en la línea de órdenes. Rehaz un cambio; forma de Ctrl-r en la línea de órdenes.

Una ruta posible

`x → :undo → <Enter> → :redo → <Enter> → .`

Misión 69 / Un patrón fiable

Graba quitar ash y bajar, repite dos veces, prueba el registro 0 con keep y deshaz la copia de prueba. Órdenes necesarias: q · @ · cantidad · "0.

Inicio: línea 1, columna 1

Texto inicial

```
1     ash dawn
2     ash dawn
3     ash dawn
4     keep
```

Texto objetivo

```
1     dawn
2     dawn
3     dawn
4     keep
```

Registro "0

```
1     keep
```

Macros grabadas: a

Practica estos comandos: q, @, count, "0

Pistas: q y una letra inician una macro; otra q termina la grabación. Reproduce la macro de un registro, como @a; admite repeticiones.

Una ruta posible

```
qadwjq2@ayy"0Pu
```

24 / Cumbre de los Comienzos

Todo el archipiélago en un registro

Luma espera donde se unen todos los caminos. No ofrece poderes nuevos, sino una última tarea para reunir memoria, precisión e imaginación.

Misión 70 / Un registro sin ruido

Cambia el contenido de la etiqueta a light, elimina todo [noise] y su línea vacía y guarda. Órdenes necesarias: it · a[· dd · :w.

Inicio: línea 1, columna 5

Texto inicial

```
1     <p>old</p>
2     [noise]
3     keep
```

Texto objetivo

```
1     <p>light</p>
2     keep
```

Practica estos comandos: it, a[, dd, :w

Pistas: Interior de una pareja de etiquetas HTML, sin las etiquetas. Un par [] completo, incluidos los corchetes.

Una ruta posible

`citlight → <Esc> → jda[dd → :w → <Enter>`

Misión 71 / El regreso preciso

Recuerda dos posiciones con A y a. Combina saltos exactos y búsquedas en ambos sentidos y vuelve a dawn. Órdenes necesarias: m · ` · / · ? · n.

Inicio: línea 1, columna 6

Destino del cursor: línea 3, columna 6.

```
1     star north
2     star south
3     star dawn
```

Marca A: línea 1, columna 6

Marca a: línea 3, columna 6

Practica estos comandos: m, `, /, ?, n

Pistas: Crea una marca: m y una letra, como ma, recuerdan la posición. Acento grave y letra de marca saltan a la posición exacta.

Una ruta posible

`mA/dawn → <Enter> → ma`A?star → <Enter> → n`a`

Misión 72 / Todas las luces, un cielo

Graba cambiar sleep por wake y aplícalo a tres líneas. Copia la última línea de message.txt a dawn.txt y guarda. Órdenes necesarias: q · @ · @@ · :w · :e · yy · "0 · p.

Inicio: línea 1, columna 1

Texto inicial

```
1     sleep
2     sleep
3     sleep
```

Texto objetivo

```
1     wake
2     wake
3     wake
4     the dawn returns
```

Registro "0

```
1     the dawn returns
```

Búfer activo: dawn.txt

Archivo virtual guardado: dawn.txt

```
1     wake
2     wake
3     wake
4     the dawn returns
```

Archivo virtual guardado: message.txt

```
1     the dawn returns
```

Macros grabadas: a

Practica estos comandos: q, @, @@, :w, :e, yy, "0, p

Pistas: q y una letra inician una macro; otra q termina la grabación. Reproduce la macro de un registro, como @a; admite repeticiones.

Una ruta posible

```
qaccwake → <Esc> → jq@a@@ → :w → <Enter> → :e<Space>message.txt → <Enter> → yy → :e<Space>dawn.txt → <Enter> → "0p → :w → <Enter>
```

Sigue aprendiendo

Este manual describe el subconjunto educativo que admite el juego. Los archivos virtuales permanecen en el navegador; el juego no accede al sistema de archivos de tu dispositivo.

Vim documentation

Motions and text objects
https://vimhelp.org/motion.txt.html

Editing and registers
https://vimhelp.org/change.txt.html

Macros
https://vimhelp.org/usr_10.txt.html

Files and buffers
https://vimhelp.org/usr_22.txt.html

LUMEN

守光人学习手册

24 个区域，72 个任务。理解 Vim 移动与编辑语言的实用指南。

24 座岛屿 / 72 项任务 / 全部免费

简体中文 / 版本 0.3.0

本手册说明游戏支持的教学子集。虚拟文件保留在浏览器中，游戏不会访问设备的文件系统。

请先自行尝试。解法仅为示例，其他正确的按键序列也可能完成目标。

目录

出发之前 3

命令图鉴 5

岛屿攻略 10

01 / 灯笼港 . . . 11
02 / 词语林地 . . . 13
03 / 行间石桥 . . . 14
04 / 回声观星台 . . . 16
05 / 余烬锻坊 . . . 17
06 / 镜影图书馆 . . . 19
07 / 曙光书写室 . . . 21
08 / 晨光之塔 . . . 23
09 / 风之峭壁 . . . 25
10 / 寻迹者山口 . . . 26
11 / 石之环 . . . 27
12 / 地平线之塔 . . . 29
13 / 织字工坊 . . . 31
14 / 语法铸造厂 . . . 33
15 / 词语圣所 . . . 35
16 / 界符宝库 . . . 37
17 / 低语礼堂 . . . 39
18 / 绘图师营地 . . . 41
19 / 回声柜 . . . 43
20 / 九重记忆厅 . . . 45
21 / 书记港 . . . 48
22 / 钟械果园 . . . 50
23 / 织工试炼 . . . 52
24 / 起始之巅 . . . 54

继续学习 56

出发之前

旅程目标

这段旅程教你有意识地操作文字：从单次移动，到文本对象、寄存器与宏。每个区域介绍一小组命令。任务要求使用指定工具，而不只是碰巧得到目标文字。

完成游戏表示练习过本课程，并不等于掌握完整 Vim 的全部功能。欢迎重玩已完成的任务，寻找清晰、可靠、适合自己的解法。

如何游玩

先选择语言，为角色取名并同意使用条款。点击棋盘，让它接收键盘输入。普通模式下，字母执行命令；插入模式下，字母成为文字。Esc 返回普通模式。触屏设备可以使用棋盘下的按键与输入框。

行动前先阅读目标，比较当前内容与目标文字，并查看必需命令。提示会逐步变得具体。重新开始只影响当前任务；地图可以带你回到已解锁区域。 你可以随时通过帮助按钮打开说明和手册。插入模式下 Esc 返回普通模式；普通模式中没有待完成命令时，Esc 打开暂停菜单，也可以点击暂停按钮。 最后三个任务可作为可选计时挑战。参考用时只是额外挑战，不是完成课程的条件。请先练习正确操作，再提高速度。

存档与隐私

角色名字、条款接受记录和进度保存在当前浏览器的 localStorage 中。这不是账号，也不是云端备份。无痕模式、清除网站数据或存储限制可能让存档消失。请下载备份，以便保存或迁移到其他设备。

可选诊断默认关闭，在设置中有独立开关，并非游戏的必要条件。诊断只包含少量技术错误数据，不包含名字、练习文字或完整存档。数据处理细节见使用条款。

按键记法

大小写有区别：w 与 W、p 与 P、i 与 I 是不同命令。3w 表示先按 3，再按 w；Ctrl-r 表示按住 Control 再按 r。+、-、|、$、%、分号与逗号都是真实的命令按键。

ciw 这样的命令由操作符 c 与文本对象 iw 组成。石板上的文件名和目标文字需要原样输入，即使说明使用其他语言。以 : 开头的命令，以及 / 或 ? 搜索，都用 Enter 确认。 请按顺序输入，不要输入步骤之间的箭头。<Esc>、<Enter>、<C-r>、<Space> 分别表示 Escape、Enter、Control+r、空格键；不要输入尖括号。

游戏与真正的 Vim

LUMEN 是针对选定 Vim 学习范围的教育模拟器。棋盘表示文本缓冲区，可见部分就是视图窗口。滚动命令改变视图，而不修改文字。文件是虚拟的：游戏不会读取电脑目录，也不会执行操作系统命令。

命令遵循所教授操作的含义，但游戏不替代具有全部选项、插件和语法例外的完整 Vim。游戏帮助和本手册描述了支持的范围。

命令行

冒号打开命令行。:w 保存虚拟文档，:e 打开另一文档，:ls 列出缓冲区，:b 切换缓冲区。:marks 列出标记；:delmarks 或缩写 :delm 删除标记，:delmarks! 清除本地标记。:reg 显示寄存器内容。

:set nu 与 :set nonu 开启、关闭绝对行号，:set nu! 切换其状态。:set rnu、:set nornu、:set rnu! 对相对行号执行对应操作。Enter 确认命令，Esc 取消。

寄存器

寄存器相当于 Vim 的剪贴板。普通复制与删除使用无名寄存器；双引号前缀可选择其他寄存器。a-z 是命名寄存器，写入 A-Z 会追加到对应的小写寄存器。

寄存器 0 保存最近复制的内容，1-9 保存较大删除的历史，- 保存小范围删除。_ 是黑洞寄存器：删除到这里不会覆盖其他剪贴板内容。用 :reg 检查实际保存了什么。

宏

宏记录一系列操作，也包括光标移动。要在寄存器 a 中录制，按 qa，执行操作，再按 q 结束。@a 回放寄存器 a，@@ 重复上一次回放的宏。回放前的数字指定重复次数。

多次回放前，先测试一次。可靠的宏应停在可预测的位置，为处理下一段文字做好准备。

虚拟文件与缓冲区

缓冲区是打开的文档，文件名标识其虚拟保存版本。:w [名称] 保存，:e 名称 打开，:ls 列出，:b 名称或编号 切换缓冲区。离开文档前，请保存希望保留的更改。

这些操作都发生在游戏及本地存档内。导出旅程得到的是备份，并不会在电脑文件系统中保存真实源文件。不要在练习中输入密码或其他机密信息。

<Esc> 表示 Escape，<Enter> 表示回车，<C-r> 表示 Ctrl+r，<Space> 表示空格。箭头只分隔片段，不需要输入。请注意大小写。

命令图鉴

命令	作用
`0`	移动到当前行的第一列。
`hjkl`	h 向左，j 向下，k 向上，l 向右：最基本的光标移动。
`h`	向左移动一格或一个字符。
`j`	向下移动一行或一格。
`k`	向上移动一行或一格。
`l`	向右移动一格或一个字符。
`w`	移动到下一个词的开头。
`b`	向后移动到词的开头。
`e`	移动到词的末尾。
`^`	移动到本行第一个非空白字符。
`$`	移动到当前行的最后一个字符。
`gg`	移动到第一行。
`G`	移动到最后一行。
`count`	数字表示下一个移动的重复次数，例如 3w 或 2j。
`f`	按 f 再按一个字符，找到它在当前行中的下一次出现。
`;`	重复上一次通过 f 进行的字符查找。
`/`	向前搜索文字，用 Enter 确认。
`n`	沿原搜索方向重复查找。
`N`	沿与上次 / 或 ? 相反的方向搜索。
`x`	删除光标所在字符。
`r`	用接下来输入的字符替换光标所在字符。
`dd`	剪切当前行并存入寄存器。
`dw`	删除从光标到下一个词开头之前的文字。
`u`	撤销上一次文字更改。
`Ctrl-r`	重做被撤销的更改。
`yy`	将当前整行复制到寄存器。

命令	作用
yw	复制从光标到下一个词开头之前的文字。
p	在光标后粘贴文字，整行则粘贴到当前行下方。
i	在光标之前开始插入。
a	在光标之后开始插入。
A	在本行末尾开始插入。
o	在下方新建一行并开始插入。
Esc	返回普通模式，或取消尚未完成的命令。
:w	保存虚拟文档；可选名称指定游戏内文件。
1 ... 9	操作符或移动前的数字指定次数；2d3w 将两个次数相乘。
W	移到下一个由空白分隔的 WORD；标点仍属于同一 WORD。
E	移到当前或下一个非空白 WORD 的末尾。
B	向后移到 WORD 开头；标点不会把它拆成多个词。
ge	向后移到上一个词的末尾。
gE	向后移到上一个 WORD 的末尾。
+	移到下一行首个非空白字符，可加次数。
-	移到上一行首个非空白字符。
|	到达数字指定的列，如 5|；无数字时到第一列。
F	按 F 再输入字符，在当前行向后查找。
t	移到本行下一个目标字符的前一格。
T	向后查找字符，停在其后一格。
,	沿相反方向重复上一次 f、F、t 或 T。
*	向前搜索光标所在的完整单词。
#	向后搜索光标所在的完整单词。
?	向后搜索文字，用 Enter 确认。
%	跳到配对的 ()、[] 或 {} 括号。
[{	向后移到包围当前区域的未匹配左花括号。
]}	向前移到包围当前区域的未匹配右花括号。
[(	向后移到包围当前区域的未匹配左圆括号。

命令	作用
`])`	向前移到包围当前区域的未匹配右圆括号。
`{`	向后移到段落边界；空行分隔段落。
`}`	向前移到下一个段落边界。
`(`	向后移到句子开头。
`)`	向前移到下一句开头。
`zt`	滚动视图，使当前行位于顶部。
`zz`	将当前行置于视图中央，不改变文字。
`zb`	滚动视图，使当前行位于底部。
`H`	移到视图中首个可见行。
`M`	移到视图中间的可见行。
`L`	移到视图中最后一个可见行。
`nu / :set nu`	显示绝对行号。
`nonu / :set nonu`	隐藏绝对行号。
`nu! / :set nu!`	切换绝对行号显示。
`rnu / :set rnu`	显示相对行号，表示与光标行的距离。
`nornu / :set nornu`	隐藏相对行号。
`rnu! / :set rnu!`	切换相对行号显示。
`X`	删除光标前的字符。
`D`	删除从光标到行尾的内容。
`~`	切换光标所在字母的大小写。
`I`	在本行首个非空白字符处开始插入。
`O`	在当前行上方新建一行并进入插入模式。
`s`	删除光标所在字符并开始插入。
`S`	替换当前整行并进入插入模式。
`d`	删除操作符：后接移动或对象，如 d$、diw、d2w。
`y`	复制操作符：与移动或文本对象组合。
`c`	修改操作符：删除指定范围并进入插入模式。
`cc`	修改当前整行并开始插入。

命令	作用
C	修改光标至行尾的内容并开始插入。
Y	复制当前整行；在本游戏中等同于 yy。
.	重复上一次文字修改，包括插入的内容。
iw	词内部，不含周围空白；与 d、y、c 组合。
aw	包含相邻空白的整个词，供操作符使用。
iW	WORD 内部：包含标点的连续非空白内容。
aW	包含周围空白的 WORD。
is	句子内部，不含句间分隔空白。
as	包含分隔空白的句子。
ip	段落内部，不含周围空行。
ap	包含分隔空行的段落。
i[/ i]	配对 [] 内部，不含括号；两个别名等价。
a[/ a]	包含括号的整对 []。
i(/ i) / ib	配对 () 内部，不含括号；ib 是别名。
a(/ a) / ab	包含括号的整对 ()；ab 是别名。
i{ / i} / iB	配对 {} 内部，不含花括号；iB 是别名。
a{ / a} / aB	包含花括号的整对 {}；aB 是别名。
i< / i>	配对 <> 内部，不含尖括号。
a< / a>	包含尖括号的整对 <>。
i"	双引号内部文字，不含引号。
a"	包含双引号的文字。
i'	单引号内部文字，不含引号。
a'	包含单引号的文字。
i`	反引号内部文字，不含反引号。
a`	包含反引号的文字。
it	配对 HTML 标签内部，不含标签。
at	包含开始与结束标签的整个元素。
m	设置标记：按 m 和字母，如 ma，记住当前位置。

命令	作用
'	单引号加标记字母，跳到该行首个非空白字符。
`	反引号加标记字母，跳到保存的精确位置。
:marks	显示已保存的标记。
:delmarks / :delm	删除指定标记，如 :delmarks a；:delm 为缩写。
:delmarks!	删除所有本地标记。
"	寄存器前缀：双引号、寄存器名，再接操作符或 p。
""	无名寄存器保存最近复制或删除的文字。
"-	小删除寄存器：保存不足整行的删除内容。
"_	黑洞寄存器：删除而不覆盖其他寄存器。
"0	寄存器 0 保留最近复制的内容，不被后续删除替换。
:reg / :registers	显示寄存器，可指定名称，如 :reg a0。
:e	打开指定名称的虚拟文件。
:ls	列出已打开的缓冲区及其编号。
:b	按名称或编号切换虚拟缓冲区。
q	q 加字母开始录制宏，再按 q 结束。
@	回放指定寄存器的宏，如 @a；前加数字可重复。
@@	再次回放上次使用的宏。
"1 ... "9	较大删除的历史：1 最新，9 为保留的最早内容。
"a ... "z	命名寄存器 a-z：在操作符前选择字母以保留文字。
"A ... "Z	大写 A-Z 向对应 a-z 寄存器追加内容，而非覆盖。
P	在光标前粘贴文字，或在当前行上方粘贴整行。
:set	修改显示选项，如 :set nu 或 :set rnu!。
:	打开命令行，输入命令并用 Enter 确认。
z	滚动前缀：用 t、z 或 b 完成。
:undo	撤销上一次编辑，是 u 的命令行形式。
:redo	重做撤销的编辑，是 Ctrl-r 的命令行形式。
:delm!	等同于 :delmarks!，清除全部本地标记。

岛屿攻略

请先自行尝试。解法仅为示例，其他正确的按键序列也可能完成目标。

<Esc> 表示 Escape，<Enter> 表示回车，<C-r> 表示 Ctrl+r，<Space> 表示空格。箭头只分隔片段，不需要输入。请注意大小写。

01 / 灯笼港

每段旅程，都从一次移动开始

海浪把你送到一座灯火尽灭的港口。奈莉在码头石缝间找到了散落的火花。收集它们，第一条道路就会重新浮现在世界中。

任务 01 / 第一点火花

用 l 向右前进，收集火花，再用 k 向上走到光门。

起点: 行 4, 列 3

收集光点: (4, 6). 光标目标: (2, 6).

```
1     #########
2     #.......#
3     #.......#
4     #.......#
5     #########
```

练习命令: l, k

提示: l 向右，k 向上。本任务需要使用这两个按键。 收集火花后任务还没结束，请把光带到发亮的门前。

一种可行路线

```
lllkk
```

任务 02 / 潮汐阶梯

使用 h 和 j 离开上方道路，收集火花，再到达左侧的光门。

起点: 行 2, 列 8

收集光点: (4, 6). 光标目标: (4, 3).

```
1     #########
2     #.......#
3     #.......#
4     #.......#
5     #########
```

练习命令: h, j

提示: h 向左，j 向下。这次需要使用两种移动。 先靠近火花，再用 j 向下收集它，随后把光带到门前。

一种可行路线

```
hhjjhhh
```

任务 03 / 点亮码头

收集所有剩下的火花。光芒归来时，港口会指向下一座岛屿。

起点: 行 2, 列 2

收集光点: (2, 8), (6, 8), (6, 2). 光标目标: (4, 4).

```
#########
#.......#
#.#####.#
#...#...#
#.#####.#
#.......#
#########
```

提示: 移动之前，先观察整条道路和光芒的位置。 路走到尽头时，可以回退几格。慢慢尝试就好，这项任务不需要抢时间。

一种可行路线

```
lllllljjjjhhhhhhkkll
```

02 / 词语林地

用词语思考，而非逐格前行

这片林地的树木从词语中生长。寂静之潮打散了它们的根，伊沃再也无法穿行于句子之间。他会教你找到整个词语的起点和终点。

任务 04 / 树根之间

使用 w 在词首之间跳跃，到达指定的位置。

起点: 行 1, 列 1

光标目标: 行 1, 列 10.

```
1     dew moss lantern river
```

练习命令: w

提示: w 移动到下一个词的开头。 空格分隔着词语的小岛。w 用一次移动就能跨过这样的间隔。

一种可行路线

ww

任务 05 / 身后的足迹

找到前面的某个词。用 b 沿着文字小径向后返回。

起点: 行 1, 列 10

光标目标: 行 1, 列 5.

```
1     dew moss lantern river
```

练习命令: b

提示: b 向后移动到词的开头。 如果位于词中间，第一次 b 会回到该词开头，再按一次才会继续向前一个词移动。

一种可行路线

b

任务 06 / 枝梢的尽头

停在正确词语的最后一个字母上，试试 e。

起点: 行 1, 列 1

光标目标: 行 1, 列 11.

```
1     soft silver light
```

练习命令: e

提示: e 移动到词语的末尾。 观察发光的目标。词首与词尾是不同的位置，请选择能准确到达的动作。

一种可行路线

we

03 / 行间石桥

从这一端，直达另一端

这些石桥只记得起点和终点。塞拉会教你找到桥的边缘，也会让移动带上次数。漫长的渡越，不一定需要很多次独立的按键。

任务 07 / 第一块石头

先用 0 到达行首，再用 ^ 到达第一个字母。

起点: 行 1, 列 10

光标目标: 行 1, 列 4.

```
1         stone bridge
```

练习命令: 0, ^

提示: 0 到达第一列，即使那里是空格；^ 会跳过行首的空格。 先按 0，再按 ^。目标是第一个字母，但本任务需要练习这两种行首移动。

一种可行路线

```
0^
```

任务 08 / 彼岸

用 G 跳到最后一行，再用 $ 到达它的末尾。

起点: 行 1, 列 1

光标目标: 行 3, 列 16.

```
1     a quiet river
2     under the bridge
3     the last lantern
```

练习命令: G, $

提示: G 到达整份记录的最后一行，$ 到达当前行的最后一个字符。 将两种移动连起来：G，然后 $。不用逐行向下走。

一种可行路线

```
G$
```

任务 09 / 迈出长步

用 gg 返回顶端，用 3j 向下移动三行，再用 0 选中第一列。

起点: 行 5, 列 5

光标目标: 行 4, 列 1.

```
1     north gate
2     mist
3     old steps
4     star bridge
5     last lantern
```

练习命令: gg, count, 0

提示: gg 到达第一行，j 前面的数字 3 表示重复移动三次。 依次使用 gg、3j、0。数字是命令的一部分，不是要插入的文字。

一种可行路线

```
gg3j0
```

04 / 回声观星台

寻找一眼看不见的道路

岛屿上空的星辰仍然闪耀，但它们的名字散落在石板各处。奥林听见归来的回声。学会寻找字符和词语，你就能读出其中的路。

任务 10 / 选中的星

使用 f 在当前行中找到字母 l。

起点: 行 1, 列 1

光标目标: 行 1, 列 8.

```
1      moss | lantern | stars
```

练习命令: f

提示: f 会等待要寻找的字符。本任务中，这个字符是 l。 先按 f，再按 l，就会到达下一个 l，不用逐个字母走过去。

一种可行路线

```
fl
```

任务 11 / 字符的回声

用 fb 找到下一个 b，再用分号 ; 重复这次查找。

起点: 行 1, 列 1

光标目标: 行 1, 列 11.

```
1      bell bell bell
```

练习命令: f, ;

提示: 按 f 后再按 b。; 会重复上一次单字符查找。 先用 fb，再按 ;，到达同一行中的另一个 b。

一种可行路线

```
fb;
```

任务 12 / 归来的回声

输入 /light 并按 Enter 搜索 light，再依次使用 n、N、n，到达最后一块石板上的回声。

起点: 行 1, 列 1

光标目标: 行 3, 列 10.

```
1      light sleeps
2      follow light
3      wake the light
```

练习命令: /, n, N

提示: n 沿同一方向寻找下一个结果，N 沿相反方向搜索。 输入 /light，按 Enter，然后依次按 n、N、n。本任务需要练习两个搜索方向。

一种可行路线

```
/light → <Enter> → nNn
```

05 / 余烬锻坊

放心修补，更改可以撤销

炉膛中剩下的温度，足以锻造新的字母。阿莎不要求每个动作都完美：熟练的匠人也懂得撤销错误。清理受损的记录，让炉火重新燃起。

任务 13 / 记录中的碎片

用 x 删除多余的字符，恢复正确的文字。

起点: 行 1, 列 6

初始文本

```
1  sparkx
```

目标文本

```
1  spark
```

练习命令: x

提示: x 删除光标所在的字符。 先把光标移到多余的字母上。删除后，检查整段文字，而不只是光标的位置。

一种可行路线

x

任务 14 / 金属的记忆

用 rg 将第一个 c 改成 g，用 u 撤销，再用 Ctrl-r 重做。

起点: 行 1, 列 1

初始文本

```
1  cold ember
2  cold ember
```

目标文本

```
1  gold ember
2  cold ember
```

练习命令: r, u, Ctrl-r

提示: r 会等待替换用的字符。rg 会把 cold 改成 gold。 rg 之后按 u，再按 Ctrl-r。目标是第一行 gold ember，第二行 cold ember。

一种可行路线

`rg → u → <C-r>`

任务 15 / 新的形状

用 dw 删除第一个 broken，再用 dd 剪掉剩下的损坏行，只留下 dawn。

起点: 行 1, 列 1

初始文本

```
1  broken broken
2  dawn
```

目标文本

```
1  dawn
```

练习命令: dw, dd

提示: dw 向下一个词删除，dd 删除当前整行。 先 dw，再 dd。两种不同大小的损坏，需要两种工具。

一种可行路线

```
dwdd
```

06 / 镜影图书馆

好的想法，值得一次回响

书架上的书页映在水中。米拉能在一个词消失之前将它保存下来。借助寄存器，你可以复制一点光，也可以把整行记录移到新的地方。

任务 16 / 双生之行

用 yy 复制一行，再用 p 把它的倒影放到下方。

起点: 行 1, 列 1

初始文本

```
1  light
2  sleep
```

目标文本

```
1  light
2  light
3  sleep
```

练习命令: yy, p

提示: yy 复制当前整行，原文仍会保留。 复制整行后，p 把它粘贴到当前行的下方。先选好副本要放在哪里。

一种可行路线

yyp

任务 17 / 镜中的词

用 yw 保存一个词，再通过 p 放到正确位置。

起点: 行 1, 列 1

初始文本

```
1  glow moss
```

目标文本

```
1  glow glow moss
```

练习命令: yw, p

提示: yw 复制从光标到下一个词开头之前的文字。 复制文字片段时，p 将它放在光标之后。这与粘贴整行的方式不同。

一种可行路线

ywwhp

任务 18 / 新的顺序

使用 dd 剪切一行，再用 p 移动它，恢复正确的顺序。

起点: 行 1, 列 1

初始文本

```
1     mist
2     sun
3     sky
```

目标文本

```
1     sun
2     mist
3     sky
```

练习命令: dd, p

提示: dd 将删除的行保存在 p 所使用的同一个寄存器中。 剪切后移动到目标位置。p 会把记住的那一行放在当前行下方。

一种可行路线

```
ddp
```

07 / 曙光书写室

留下属于你的新记录

伊诺发现编年史中出现了空缺。这些词无法只靠移动找回，必须有人重新写下它们。在这里，你将学会在操作文字与输入文字之间切换的节奏。

任务 19 / 缺少的词

放好光标，使用 i 进入插入模式，输入缺少的文字。

起点: 行 1, 列 5

初始文本

```
1     the returns
```

目标文本

```
1     the dawn returns
```

练习命令: i, Esc

提示: i 从光标之前开始插入文字。 插入模式下，字母会变成文字内容。Esc 可以恢复普通模式与移动命令。

一种可行路线

```
i → dawn<Space> → <Esc>
```

任务 20 / 更多的空间

用 a 添加 s，再按 Esc；随后用 A 在行尾添加一个空格和 shine。

起点: 行 1, 列 4

初始文本

```
1     star
```

目标文本

```
1     stars shine
```

练习命令: a, A, Esc

提示: a 在光标之后插入，A 在整行末尾插入。 用 a、s、Esc 将 star 改成 stars，再用 A、空格、shine、Esc，得到 stars shine。

一种可行路线

```
as → <Esc> → A<Space>shine → <Esc>
```

任务 21 / 封存编年史

用 o 新建一行，输入 returns，再按 Esc，最后用 :w 和 Enter 保存。

起点: 行 1, 列 1

初始文本

```
1  dawn
```

目标文本

```
1  dawn
2  returns
```

练习命令: o, Esc, :w

提示: o 在下方新建一行，并立即进入插入模式。 在 dawn 下方写入 returns，再依次按 Esc、:w、Enter，封存完整的石板。

一种可行路线

`oreturns → <Esc> → :w → <Enter>`

08 / 晨光之塔

所有道路，最终都通向光

塔顶的透镜由最初几座岛屿的碎片组成。结合学到的能力，唤来第一道晨光。它将照亮更多区域，开启通向熟练掌握的道路。

任务 22 / 记忆中的地图

用 dd 剪掉损坏的第一行，用 yy 复制正确的一行，用 p 粘贴，再用 :w 确认。

起点: 行 1, 列 1

初始文本

```
1  lost lost
2  keep the light
```

目标文本

```
1  keep the light
2  keep the light
```

练习命令: dd, yy, p, :w

提示: dd 后剩下 keep the light。yy 记住整行，p 在下方创建副本。 依次使用 dd、yy、p，再输入 :w 并按 Enter。石板上应留下两行相同的 keep the light。

一种可行路线

`ddyyp:w → <Enter>`

任务 23 / 最后的修补

用 /sleep 和 Enter 找到 sleep，用 dw 删除它，再用 A 添加 sing，最后用 :w 保存。

起点: 行 1, 列 1

初始文本

```
1  moss covers dawn
2  the bells sleep
```

目标文本

```
1  moss covers dawn
2  the bells sing
```

练习命令: /, dw, A, :w

提示: 搜索会让光标停在 sleep 上。dw 删除这个词，A 让你从行尾继续写。 第一行保持不变，第二行应为 the bells sing。按 Esc 结束插入，再用 :w 和 Enter 保存。

一种可行路线

`/sleep → <Enter> → dwAsing → <Esc> → :w → <Enter>`

任务 24 / 让光归来

删除第一行，复制剩下的一行，再另起一行写入 the dawn returns，保存修复后的编年史。

起点: 行 1, 列 1

初始文本

```
1     the night stays
2     one small spark
```

目标文本

```
1     one small spark
2     one small spark
3     the dawn returns
```

练习命令: dd, yy, p, o, :w

提示: dd 删除损坏行；yy 和 p 创建第二行 one small spark；o 为最后一行腾出空间。 在 dd、yy、p 之后，使用 o 并输入 the dawn returns，再按 Esc、:w、Enter，唤来第一道晨光，开启后面的道路。

一种可行路线

```
ddyypothe<Space>dawn<Space>returns → <Esc> → :w → <Enter>
```

09 / 风之峭壁

当一个词不只是一个词

风带来散落的文字缎带。塔维教你跨过整段不含空白的文字，再沿原路返回，而不用逐字计数。

任务 25 / 连续的缎带

用 W 跨过整个 one-two，停在 three.four 的开头。 必需命令: W.

起点: 行 1, 列 1

光标目标: 行 1, 列 9.

```
1     one-two three.four five
```

练习命令: W

提示: 移到下一个由空白分隔的 WORD；标点仍属于同一 WORD。 先有意识地执行一次，再检查光标或文字。理解单次操作后，再添加重复次数。

一种可行路线

```
W
```

任务 26 / 沿山脊返回

比较 E、B 与 ge、gE，最后返回第一列。 必需命令: E · B · ge · gE.

起点: 行 1, 列 23

光标目标: 行 1, 列 1.

```
1     one-two three.four five
```

练习命令: E, B, ge, gE

提示: 移到当前或下一个非空白 WORD 的末尾。 向后移到 WORD 开头；标点不会把它拆成多个词。

一种可行路线

```
BEgegE0
```

任务 27 / 丈量步伐

练习 +、- 和移动次数，再用列跳转到 third 的字母 t。 必需命令: + · - · | · 次数.

起点: 行 1, 列 4

光标目标: 行 3, 列 2.

```
1      first
2        second
3      third
```

练习命令: +, -, |, count

提示: 移到下一行首个非空白字符，可加次数。 移到上一行首个非空白字符。

一种可行路线

```
+-2+2|
```

10 / 寻迹者山口

也要看看身后的路

山口的路由细小刻痕标出。林恩教你双向寻找，准确停在下一块安全的石头旁。

任务 28 / 逆着足迹

向后查找 a，比较 t 与 T，再用逗号反向，停在最后一个 a 之前。 必需命令: F · t · T · ,.

起点: 行 1, 列 13

光标目标: 行 1, 列 12.

```
1     a x a y a z a
```

练习命令: F, t, T, ,

提示: 按 F 再输入字符，在当前行向后查找。 移到本行下一个目标字符的前一格。

一种可行路线

```
FataTa,
```

任务 29 / 词语的回应

用 * 和 # 双向寻找完整单词 glow，最后停在第一行第二个 glow 上。 必需命令: * · #.

起点: 行 1, 列 1

光标目标: 行 1, 列 11.

```
1     glow moss glow
2     glow
```

练习命令: *, #

提示: 向前搜索光标所在的完整单词。 向后搜索光标所在的完整单词。

一种可行路线

```
*#*
```

任务 30 / 反向的道路

用 ? 向后搜索 star，练习 n 和 N，停在中间的结果上。 必需命令: ? · n · N.

起点: 行 3, 列 1

光标目标: 行 2, 列 1.

```
1     star
2     star
3     star
```

练习命令: ?, n, N

提示: 向后搜索文字，用 Enter 确认。 沿原搜索方向重复查找。

一种可行路线

```
?star → <Enter> → nN
```

11 / 石之环

每次开启，都有闭合

乌玛守护环中之环。读懂括号、句子边界与段落间的呼吸，石阵便不再是迷宫。

任务 31 / 成对的石头

用 % 与块边界移动探索嵌套括号，最后返回最外层的左花括号。 必需命令: % · [(·]) · [{ ·]}.

起点: 行 2, 列 5

光标目标: 行 1, 列 1.

```
1     {
2       [a(b)c]
3     }
```

练习命令: %, [(,]), [{,]}

提示: 跳到配对的 ()、[] 或 {} 括号。 向后移到包围当前区域的未匹配左圆括号。

一种可行路线

```
% → [( → ]) → [{ → ]} → %
```

任务 32 / 段落之间

在段落之间前进和返回，停在 north 与 south 之间的空行。 必需命令: { · }.

起点: 行 1, 列 1

光标目标: 行 3, 列 1.

```
1     north
2     path
3     [空行]
4     south
5     gate
6     [空行]
7     dawn
```

练习命令: {, }

提示: 向后移到段落边界；空行分隔段落。 向前移到下一个段落边界。

一种可行路线

```
}}{
```

任务 33 / 呼吸的空间

双向访问句子开头，再返回 Second 的开头。 必需命令: (•).

起点: 行 1, 列 1

光标目标: 行 1, 列 14.

```
1     First light. Second light! Third light?
```

练习命令: (,)

提示: 向后移到句子开头。 向前移到下一句开头。

一种可行路线

```
)) (
```

12 / 地平线之塔

改变视角，而非文字

塔窗沿着卷轴移动。诺娅带你区分光标所在的位置，与窗口此刻展现的内容。

任务 34 / 升起的窗口

尝试三种窗口位置，最后让当前第五行位于视图顶部。 必需命令: zt · zz · zb.

起点: 行 5, 列 1

光标目标: 行 5, 列 1.

```
1     line one
2     line two
3     line three
4     line four
5     line five
6     line six
7     line seven
8     line eight
9     line nine
```

首个可见行: 5

练习命令: zt, zz, zb

提示: 滚动视图，使当前行位于顶部。 将当前行置于视图中央，不改变文字。

一种可行路线

`ztzzzbzt`

任务 35 / 稳定的地平线

将第五行居中，访问窗口的顶部与底部，再返回中间。 必需命令: H · M · L.

起点: 行 5, 列 1

光标目标: 行 5, 列 1.

```
1      line one
2      line two
3      line three
4      line four
5      line five
6      line six
7      line seven
8      line eight
9      line nine
```

首个可见行: 3

练习命令: H, M, L

提示: 移到视图中首个可见行。 移到视图中间的可见行。

一种可行路线

`zzHLM`

任务 36 / 身旁的数字

练习开启、关闭和切换两种行号，最后开启绝对行号并关闭相对行号。 必需命令: nu · nonu · nu! · rnu · nornu · rnu!.

起点: 行 1, 列 1

光标目标: 行 1, 列 1.

```
1      north
2      south
```

绝对行号: 开启

相对行号: 关闭

练习命令: nu, nonu, nu!, rnu, nornu, rnu!

提示: 显示绝对行号。 隐藏绝对行号。

一种可行路线

`:set<Space>nonu → <Enter> → :set<Space>nu → <Enter> → :set<Space>nu! → <Enter> → :set<Space>nu! → <Enter> → :set<Space>rnu → <Enter> → :set<Space>nornu → <Enter> → :set<Space>rnu! → <Enter> → :set<Space>rnu! → <Enter>`

13 / 织字工坊

剪开、添补、缝合

艾达修补以字母织成的布。有时要伸向光标前方，有时来到上一行，有时则要重新开始整段针脚。

任务 37 / 丝线之前

删除光标前的 x，将 DAWN 改成小写并去掉 junk，保留 dawn 后的空格。 必需命令: X · ~ · D.

起点: 行 1, 列 2

初始文本

```
1     xDAWN junk
```

目标文本

```
1     dawn
```

练习命令: X, ~, D

提示: 删除光标前的字符。 切换光标所在字母的大小写。

一种可行路线

```
X4~wD
```

任务 38 / 剪开与缝合

用 I 在 glow 前添加 the 并保留缩进，再用 O 在上方新建 header 行。 必需命令: I · O.

起点: 行 1, 列 7

初始文本

```
1        glow
```

目标文本

```
1     header
2        the glow
```

练习命令: I, O

提示: 在本行首个非空白字符处开始插入。 在当前行上方新建一行并进入插入模式。

一种可行路线

```
Ithe<Space> → <Esc> → Oheader → <Esc>
```

任务 39 / 新的一层

用 s 将 x 换成 sun，再用 S 将第二整行换成 moon。 必需命令: s · S.

起点: 行 1, 列 1

初始文本

```
1     x
2     bad line
```

目标文本

```
1     sun
2     moon
```

练习命令: s, S

提示: 删除光标所在字符并开始插入。 替换当前整行并进入插入模式。

一种可行路线

`ssun → <Esc> → jSmoon → <Esc>`

14 / 语法铸造厂

动作与它的范围

凯莉丝把命令铸成两部分：做什么，以及对什么做。组合操作符与移动，发现短短表达中的意义。

任务 40 / 动作与道路

比较 c 改词、cc 改整行、C 改行尾，得到 warm sun 与 fixed。 必需命令: c · cc · C.

起点: 行 1, 列 1

初始文本

```
1     cold moss
2     broken
```

目标文本

```
1     warm sun
2     fixed
```

练习命令: c, cc, C

提示: 修改操作符：删除指定范围并进入插入模式。 修改当前整行并开始插入。

一种可行路线

`cewarm → <Esc> → jccfixed → <Esc> → ggwCsun → <Esc>`

任务 41 / 保留并粘贴

用 y 复制一个词，用 P 在光标前粘贴，再练习 Y 与删除，并保留 north。 必需命令: y · Y · P · d.

起点: 行 1, 列 1

初始文本

```
1     red blue
2     north
```

目标文本

```
1     red red blue
2     north
```

练习命令: y, Y, P, d

提示: 复制操作符：与移动或文本对象组合。 复制当前整行；在本游戏中等同于 yy。

一种可行路线

`ywwPYpdd`

任务 42 / 再做一次

用 dw 删除第一个 bad，再重复修改两次，只留下 light。 必需命令: dw · ..

起点: 行 1, 列 1

初始文本

```
1     bad bad bad light
```

目标文本

```
1     light
```

练习命令: dw, .

提示: 删除从光标到下一个词开头之前的文字。 重复上一次文字修改，包括插入的内容。

一种可行路线

```
dw..
```

15 / 词语圣所

选择意义，而非距离

伊莱不靠数字符丈量文字。在这里，词、句、段落成为承载你意图的现成容器。

任务 43 / 词的核心

比较 iw 与 aw：删除 moss 内部，再删除 blue 与相邻空白，只留下 red。 必需命令: iw · aw.

起点: 行 1, 列 6

初始文本

```
1      red moss blue
```

目标文本

```
1      red
```

练习命令: iw, aw

提示: 词内部，不含周围空白；与 d、y、c 组合。 包含相邻空白的整个词，供操作符使用。

一种可行路线

`diwwdaw`

任务 44 / 更大的词

将 moss.green 视作同一个 WORD，练习 iW、aW，只留下 red。 必需命令: iW · aW.

起点: 行 1, 列 7

初始文本

```
1      red moss.green blue
```

目标文本

```
1      red
```

练习命令: iW, aW

提示: WORD 内部：包含标点的连续非空白内容。 包含周围空白的 WORD。

一种可行路线

`diWwdaW`

任务 45 / 句子与段落

复制句子内部与整体，再复制段落内部；删除 alpha/beta 段及其间隔，保留句子与 keep。 必需命令: is · as · ip · ap.

起点: 行 1, 列 1

初始文本

```
1     One. Two.
2     [空行]
3     alpha
4     beta
5     [空行]
6     keep
```

目标文本

```
1     One. Two.
2     [空行]
3     keep
```

练习命令: is, as, ip, ap

提示: 句子内部，不含句间分隔空白。 包含分隔空白的句子。

一种可行路线

```
yisyasjjyipdap
```

16 / 界符宝库

内部与它的外框

森恩守护盒中之盒。选择内容或整个外框；不同名字的钥匙，有时能打开同一把锁。

任务 46 / 方形与圆形

用列出的所有别名复制 []、() 的内部与外框，最后让无名寄存器保存完整的 (glow)。 必需命令: i[· a[· i] · a] · i(· a(· i) · a) · ib · ab.

起点: 行 1, 列 3

光标目标: 行 1, 列 8.

```
1     [moss] (glow)
```

寄存器 ""

```
1     (glow)
```

练习命令: i[, a[, i], a], i(, a(, i), a), ib, ab

提示: 配对 [] 内部，不含括号；两个别名等价。 包含括号的整对 []。

一种可行路线

```
yi[ya[yi]ya]f(yi(ya(yi)ya)yibyab
```

任务 47 / 尖角与别名

比较 {} 与 <> 的所有别名，最后在寄存器中保存包含外框的完整 <dawn>。 必需命令: i{ · a{ · i} · a} · iB · aB · i< · a< · i> · a>.

起点: 行 1, 列 3

光标目标: 行 1, 列 9.

```
1     {ember} <dawn>
```

寄存器 ""

```
1     <dawn>
```

练习命令: i{, a{, i}, a}, iB, aB, i<, a<, i>, a>

提示: 配对 {} 内部，不含花括号；iB 是别名。 包含花括号的整对 {}；aB 是别名。

一种可行路线

```
yi{ya{yi}ya}yiByaBf<yi<ya<yi>ya>
```

任务 48 / 完整的外框

只将 (seed) 内部改成 bloom，保留括号，再删除整个 [noise]。 必需命令: i(· a[.

起点: 行 1, 列 11

初始文本

```
1     [noise] (seed) {ash} <sky>
```

目标文本

```
1      (bloom) {ash} <sky>
```

练习命令: i(, a[

提示: 配对 () 内部，不含括号；ib 是别名。 包含括号的整对 []。

一种可行路线

```
ci(bloom → <Esc> → 0da[
```

17 / 低语礼堂

聆听符号之间

优娜把消息保存在引号和标签中。要听见内容，就要区分声音与承载它的外框。

任务 49 / 两种轻声

比较双引号与单引号对象的内部和整体，最后复制包含单引号的完整 glow。 必需命令: i" · a" · i' · a'.

起点: 行 1, 列 3

光标目标: 行 1, 列 8.

```
1     "moss" 'glow'
```

寄存器 ""

```
1     'glow'
```

练习命令: i", a", i', a'

提示: 双引号内部文字，不含引号。 包含双引号的文字。

一种可行路线

```
yi"ya"f'yi'ya'
```

任务 50 / 低语与标签

复制反引号内部及整体，再复制 HTML 元素，保留完整的 glow。 必需命令: i` · a` · it · at.

起点: 行 1, 列 3

光标目标: 行 1, 列 8.

```
1     `seed` <b>glow</b>
```

寄存器 ""

```
1     <b>glow</b>
```

练习命令: i`, a`, it, at

提示: 反引号内部文字，不含反引号。 包含反引号的文字。

一种可行路线

```
yi`ya`f<yityat
```

任务 51 / 嵌套的消息

只将 p 元素内容改成 new 并保留标签，删除整个引号中的 noise，保留行尾空格。 必需命令: it · a".

起点: 行 1, 列 5

初始文本

```
1     <p>old</p> "noise"
```

目标文本

```
1     <p>new</p>
```

练习命令: it, a"

提示: 配对 HTML 标签内部，不含标签。 包含双引号的文字。

一种可行路线

`citnew → <Esc> → f"da"`

18 / 绘图师营地

留下可以返回的坐标

阿罗画出不会被风吹散的地图。记住位置，返回对应行或精确列，再移除不再需要的路标。

任务 52 / 立下路标

在第二行设置标记 a，比较按行跳转与精确列跳转，再返回保存的位置。 必需命令: m ・ ' ・ `.

起点: 行 2, 列 5

光标目标: 行 2, 列 5.

```
1       north
2       south
3       dawn
```

标记 a: 行 2, 列 5

练习命令: m, ', `

提示: 设置标记：按 m 和字母，如 ma，记住当前位置。 单引号加标记字母，跳到该行首个非空白字符。

一种可行路线

```
magg'agg`a
```

任务 53 / 读懂地图

在相邻列设置 a、b，打开标记列表，再用缩写 :delm 只删除 a。 必需命令: :marks ・ :delm.

起点: 行 1, 列 1

光标目标: 行 1, 列 2.

```
1     path
```

标记 b: 行 1, 列 2

已删除标记: a

打开的列表: 标记

练习命令: :marks, :delm

提示: 显示已保存的标记。 删除指定标记，如 :delmarks a；:delm 为缩写。

一种可行路线

```
malmb → :marks → <Enter> → :delm<Space>a → <Enter>
```

任务 54 / 擦去旧路

练习删除标记范围、:delmarks! 和 :delm!，最后清除 a、b、c。 必需命令: :delmarks · :delmarks! · :delm!.

起点: 行 1, 列 1

光标目标: 行 1, 列 2.

```
1     path
```

已删除标记: a, b, c

练习命令: :delmarks, :delmarks!, :delm!

提示: 删除指定标记，如 :delmarks a；:delm 为缩写。 删除所有本地标记。

一种可行路线

`malmb → :delmarks<Space>a-b → <Enter> → ma → :delmarks! → <Enter> → mc → :delm! → <Enter>`

19 / 回声柜

剪贴板也可以有名字

埃丝梅将回声分别存入抽屉。保留一个，向另一个追加，让杂音消失而不盖住重要内容。

任务 55 / 无名的回声

删除 a，从小删除寄存器恢复，再通过显式无名寄存器复制整个 ab。 必需命令: "- · "" · x · yy · P.

起点: 行 1, 列 1

目标文本

```
1  ab
```

寄存器 ""

```
1  ab
```

寄存器 "-

```
1  a
```

练习命令: "-, "", x, yy, P

提示: 小删除寄存器：保存不足整行的删除内容。 无名寄存器保存最近复制或删除的文字。

一种可行路线

```
x"-P""yy
```

任务 56 / 追加而非覆盖

将 moss 存入 a，用 A 追加 glow，再粘贴两行，并检查寄存器 a。 必需命令: "a · "A · p.

起点: 行 1, 列 1

初始文本

```
1  moss
2  glow
```

目标文本

```
1  moss
2  moss
3  glow
4  glow
```

寄存器 "a

```
1  moss
2  glow
```

练习命令: "a, "A, p

提示: 命名寄存器 a-z：在操作符前选择字母以保留文字。 大写 A-Z 向对应 a-z 寄存器追加内容，而非覆盖。

一种可行路线

```
"ayyj"Ayygg"ap
```

任务 57 / 安静保护副本

将 keep 存入 a，把 trash 删除到黑洞，再将保留的副本粘贴到当前行上方。 必需命令: "_ · "a · dd · P.

起点: 行 1, 列 1

初始文本

```
1  keep
2  trash
```

目标文本

```
1  keep
2  keep
```

寄存器 "a

```
1  keep
```

寄存器 ""

```
1  keep
```

练习命令: "_, "a, dd, P

提示: 黑洞寄存器：删除而不覆盖其他寄存器。 命名寄存器 a-z：在操作符前选择字母以保留文字。

一种可行路线

```
"ayyj"_dd"aP
```

20 / 九重记忆厅

记忆也有顺序

诺娜记得的不只是最近一次消失。打开历史的抽屉，发现删除并不一定意味着告别。

任务 58 / 最后的副本

复制 light，删除 dust，再从寄存器 0 恢复副本；寄存器 1 仍应保留 dust。 必需命令: "0 · dd.

起点: 行 1, 列 1

初始文本

```
1  light
2  dust
```

目标文本

```
1  light
2  light
```

寄存器 "0

```
1  light
```

寄存器 "1

```
1  dust
```

练习命令: "0, dd

提示: 寄存器 0 保留最近复制的内容，不被后续删除替换。 剪切当前行并存入寄存器。

一种可行路线

```
yyjdd"0p
```

任务 59 / 一连串告别

删除九行，检查历史寄存器 1 至 9，再把最早的 one 恢复到 keep 上方。 必需命令: "1 · "2 · "3 · "4 · "5 · "6 · "7 · "8 · "9.

起点: 行 1, 列 1

初始文本

```
1     one
2     two
3     three
4     four
5     five
6     six
7     seven
8     eight
9     nine
10    keep
```

目标文本

```
1     one
2     keep
```

寄存器 "1

```
1     nine
```

寄存器 "2

```
1     eight
```

寄存器 "3

```
1     seven
```

寄存器 "4

```
1     six
```

寄存器 "5

```
1     five
```

寄存器 "6

```
1     four
```

寄存器 "7

```
1     three
```

寄存器 "8

```
1     two
```

寄存器 "9

```
1     one
```

练习命令: "1, "2, "3, "4, "5, "6, "7, "8, "9

提示: 较大删除的历史：1 最新，9 为保留的最早内容。 较大删除的历史：1 最新，9 为保留的最早内容。

一种可行路线

`dd........ → "1pu → "2pu → "3pu → "4pu → "5pu → "6pu → "7pu → "8pu → "9P`

任务 60 / 读懂抽屉

将 one 复制到 a，再打开寄存器查看器，确认 a 与 0 保存相同行。 必需命令: :reg.

起点: 行 1, 列 1

光标目标: 行 1, 列 1.

```
1	one
2	two
```

寄存器 "0

```
1	one
```

寄存器 "a

```
1	one
```

打开的列表: 寄存器

练习命令: :reg

提示: 显示寄存器，可指定名称，如 :reg a0。 先有意识地执行一次，再检查光标或文字。理解单次操作后，再添加重复次数。

一种可行路线

`"ayy → :reg → <Enter>`

21 / 书记港

许多页面，同一旅程

特罗在虚拟港口间摆渡文档。保存想留下的内容，再切换打开的页面，不遗失它们的名字。

任务 61 / 记住的页面

将 dawn rises 保存到 notes.txt，把 echo.txt 改为 awake 并保存，再返回 notes.txt，保留全部更改。 必需命令: :w • :e • c.

起点: 行 1, 列 1

初始文本

```
1  dawn
```

目标文本

```
1  dawn rises
```

活动缓冲区: notes.txt

已保存虚拟文件: notes.txt

```
1  dawn rises
```

已保存虚拟文件: echo.txt

```
1  awake
```

练习命令: :w, :e, c

提示: 保存虚拟文档；可选名称指定游戏内文件。 打开指定名称的虚拟文件。

一种可行路线

```
A<Space>rises → <Esc> → :w → <Enter> → :e<Space>echo.txt → <Enter> → ccawake → <Esc> → :w → <Enter> → :e<Space>notes.txt → <Enter>
```

任务 62 / 另一处港湾

打开 beta.txt，列出缓冲区，再按名称返回 alpha.txt。 必需命令: :e • :ls • :b.

起点: 行 1, 列 1

光标目标: 行 1, 列 1.

```
1  north
```

活动缓冲区: alpha.txt

打开的列表: 缓冲区

练习命令: :e, :ls, :b

提示: 打开指定名称的虚拟文件。 列出已打开的缓冲区及其编号。

一种可行路线

```
:e<Space>beta.txt → <Enter> → :ls → <Enter> → :b<Space>alpha.txt → <Enter>
```

任务 63 / 缓冲区之间

将副本保存为 final.txt，打开它并练习按编号切换缓冲区，最后停在 final.txt。 必需命令: :w · :e · :b.

起点: 行 1, 列 1

目标文本

```
1     light
```

活动缓冲区: final.txt

已保存虚拟文件: final.txt

```
1     light
```

练习命令: :w, :e, :b

提示: 保存虚拟文档；可选名称指定游戏内文件。 打开指定名称的虚拟文件。

一种可行路线

```
:w<Space>final.txt → <Enter> → :e<Space>final.txt → <Enter> → :b<Space>1 → <Enter> → :b<Space>2 → <Enter>
```

22 / 钟械果园

让一个好动作结出许多果实

奥米照料果园精细的机关。录制一个可靠循环，确认它停在哪里，再让它处理接下来的枝条。

任务 64 / 小小机器

在 a 中录制添加 ! 并下移一行，回放一次，修改前两行 dawn。 必需命令: q · @.

起点: 行 1, 列 1

初始文本

```
1	dawn
2	dawn
3	dawn
```

目标文本

```
1	dawn!
2	dawn!
3	dawn
```

已录制宏: a

练习命令: q, @

提示: q 加字母开始录制宏，再按 q 结束。 回放指定寄存器的宏，如 @a；前加数字可重复。

一种可行路线

`qaA! → <Esc> → jq@a`

任务 65 / 果园重复运转

录制在行首添加 + 并下移，用 @a 和 @@，保留第四行 seed 不变。 必需命令: q · @ · @@.

起点: 行 1, 列 1

初始文本

```
1  seed
2  seed
3  seed
4  seed
```

目标文本

```
1  +seed
2  +seed
3  +seed
4  seed
```

已录制宏: a

练习命令: q, @, @@

提示: q 加字母开始录制宏，再按 q 结束。 回放指定寄存器的宏，如 @a；前加数字可重复。

一种可行路线

`qaI+ → <Esc> → jq@a@@`

任务 66 / 再一次，准确地

录制删除一个 x，再回放三次，只留下 light。 必需命令: q · @ · 次数.

起点: 行 1, 列 1

初始文本

```
1  xxxxlight
```

目标文本

```
1  light
```

已录制宏: a

练习命令: q, @, count

提示: q 加字母开始录制宏，再按 q 结束。 回放指定寄存器的宏，如 @a；前加数字可重复。

一种可行路线

`qaxq3@a`

23 / 织工试炼

组合已经学会的图案

薇尔展开长长的纹样。次数、重复编辑和宏是不同工具；选择最能表达你意图的一种。

任务 67 / 两个次数

结合操作符与移动次数，用一次 2d3w 删除六个词，保留 keep。 必需命令: 次数 · dw.

起点: 行 1, 列 1

初始文本

```
1      one two three four five six keep
```

目标文本

```
1      keep
```

练习命令: count, dw

提示: 操作符或移动前的数字指定次数；2d3w 将两个次数相乘。 删除从光标到下一个词开头之前的文字。

一种可行路线

```
2d3w
```

任务 68 / 同样的编织

删除 x，用 :undo 与 :redo 撤销和重做，再用点号重复删除。 必需命令: :undo · :redo · .

起点: 行 1, 列 1

初始文本

```
1      xxdawn
```

目标文本

```
1      dawn
```

练习命令: :undo, :redo, .

提示: 撤销上一次编辑，是 u 的命令行形式。 重做撤销的编辑，是 Ctrl-r 的命令行形式。

一种可行路线

```
x → :undo → <Enter> → :redo → <Enter> → .
```

任务 69 / 可靠的图案

录制删除 ash 并下移，重复两次，用 keep 测试寄存器 0，再撤销试贴的副本。 必需命令: q · @ · 次数 · "0.

起点: 行 1, 列 1

初始文本

```
1	ash dawn
2	ash dawn
3	ash dawn
4	keep
```

目标文本

```
1	dawn
2	dawn
3	dawn
4	keep
```

寄存器 "0

```
1	keep
```

已录制宏: a

练习命令: q, @, count, "0

提示: q 加字母开始录制宏，再按 q 结束。 回放指定寄存器的宏，如 @a；前加数字可重复。

一种可行路线

```
qadwjq2@ayy"0Pu
```

24 / 起始之巅

将整个群岛写入一份记录

露玛在所有道路汇合处等待。她不赐予新力量，只给出最后一项任务，让记忆、精准与想象相遇。

任务 70 / 没有杂音的记录

把标签内容改成 light，删除整个 [noise] 及其空行，再保存。 必需命令: it · a[· dd · :w.

起点: 行 1, 列 5

初始文本

```
1	<p>old</p>
2	[noise]
3	keep
```

目标文本

```
1	<p>light</p>
2	keep
```

练习命令: it, a[, dd, :w

提示: 配对 HTML 标签内部，不含标签。 包含括号的整对 []。

一种可行路线

```
citlight → <Esc> → jda[dd → :w → <Enter>
```

任务 71 / 精确的归路

用 A 与 a 记住两个位置，结合精确跳转与双向搜索，最后回到 dawn。 必需命令: m · ` · / · ? · n.

起点: 行 1, 列 6

光标目标: 行 3, 列 6.

```
1	star north
2	star south
3	star dawn
```

标记 A: 行 1, 列 6

标记 a: 行 3, 列 6

练习命令: m, `, /, ?, n

提示: 设置标记：按 m 和字母，如 ma，记住当前位置。 反引号加标记字母，跳到保存的精确位置。

一种可行路线

```
mA/dawn → <Enter> → ma`A?star → <Enter> → n`a
```

任务 72 / 所有光，同一片天

录制将 sleep 改为 wake，应用于三行，再从 message.txt 复制最后一行到 dawn.txt 并保存。 必需命令: q · @ · @@ · :w · :e · yy · "0 · p.

起点: 行 1, 列 1

初始文本

```
1     sleep
2     sleep
3     sleep
```

目标文本

```
1     wake
2     wake
3     wake
4     the dawn returns
```

寄存器 "0

```
1     the dawn returns
```

活动缓冲区: dawn.txt

已保存虚拟文件: dawn.txt

```
1     wake
2     wake
3     wake
4     the dawn returns
```

已保存虚拟文件: message.txt

```
1     the dawn returns
```

已录制宏: a

练习命令: q, @, @@, :w, :e, yy, "0, p

提示: q 加字母开始录制宏，再按 q 结束。 回放指定寄存器的宏，如 @a；前加数字可重复。

一种可行路线

```
qaccwake → <Esc> → jq@a@@ → :w → <Enter> → :e<Space>message.txt → <Enter> → yy → :e<Space>dawn.txt → <Enter> → "0p → :w → <Enter>
```

继续学习

本手册说明游戏支持的教学子集。虚拟文件保留在浏览器中，游戏不会访问设备的文件系统。

Vim documentation

Motions and text objects
https://vimhelp.org/motion.txt.html

Editing and registers
https://vimhelp.org/change.txt.html

Macros
https://vimhelp.org/usr_10.txt.html

Files and buffers
https://vimhelp.org/usr_22.txt.html

LUMEN

빛의 수호자 학습서

24개의 지역, 72개의 임무. Vim의 이동과 편집 언어를 익히는 실용 안내서입니다.

섬 24개 / 임무 72개 / 모두 무료

한국어 / 판 0.3.0

이 안내서는 게임에서 지원하는 교육용 일부 기능을 설명합니다. 가상 파일은 브라우저에만 있으며 기기의 파일 시스템에 접근하지 않습니다.

먼저 직접 시도해 보세요. 해답은 예시이며 다른 올바른 키 순서로도 목표를 달성할 수 있습니다.

목차

여행을 시작하기 전에 3

명령 도감 5

섬별 안내 10

01 / 등불 항구 . . . 11
02 / 단어의 숲 . . . 13
03 / 행과 행 사이의 다리 . . . 14
04 / 메아리 관측소 . . . 16
05 / 불씨 대장간 . . . 17
06 / 거울 도서관 . . . 19
07 / 새벽의 서고 . . . 21
08 / 아침의 탑 . . . 23
09 / 바람의 절벽 . . . 25
10 / 추적자의 고개 . . . 26
11 / 돌의 고리 . . . 27
12 / 수평선의 탑 . . . 29
13 / 직조공의 작업실 . . . 31
14 / 문법 주조소 . . . 33
15 / 단어의 성소 . . . 35
16 / 구분자의 금고 . . . 37
17 / 속삭임 예배당 . . . 39
18 / 지도 제작자의 야영지 . . . 41
19 / 메아리의 서랍장 . . . 43
20 / 아홉 기억의 방 . . . 45
21 / 서기들의 항구 . . . 48
22 / 태엽 과수원 . . . 50
23 / 직조공의 시험 . . . 52
24 / 시작의 정상 . . . 54

계속 배우기 56

여행을 시작하기 전에

여행의 목표

이 여정은 한 번의 이동부터 텍스트 객체, 레지스터, 매크로까지 의도적으로 내용을 다루는 법을 가르칩니다. 각 지역은 작은 명령 묶음을 소개합니다. 임무에서는 우연히 목표 내용만 만드는 것이 아니라 지정한 도구를 사용해야 합니다.

게임을 마쳤다는 것은 이 학습 과정을 연습했다는 뜻이지, 실제 Vim의 모든 기능을 익혔다는 뜻은 아닙니다. 완료한 임무를 다시 방문해 명확하고 반복 가능한 자신만의 풀이를 찾아보세요.

플레이 방법

언어를 선택하고 캐릭터의 이름을 정한 뒤 이용약관에 동의하세요. 보드를 클릭하면 키보드 입력을 받습니다. 일반 모드의 글자는 명령이고, 입력 모드의 글자는 내용이 됩니다. Esc로 일반 모드에 돌아옵니다. 터치 기기에서는 보드 아래의 키와 입력란을 이용하세요.

행동하기 전에 목표, 목표 내용과 현재 내용의 차이, 필수 명령을 확인하세요. 힌트는 단계적으로 구체화됩니다. 다시 시작은 현재 임무에만 적용되며, 지도로 이미 열린 지역을 다시 방문할 수 있습니다. 도움말 버튼으로 언제든 설명과 학습서를 열 수 있습니다. 입력 모드에서 Esc는 일반 모드로 돌아갑니다. 일반 모드에서 미완성 명령이 없으면 일시 정지를 엽니다. 일시 정지 버튼도 있습니다. 마지막 세 임무는 선택 시간 도전으로 플레이할 수 있습니다. 권장 시간은 추가 도전이지 과정 완료의 조건이 아닙니다. 속도보다 정확성을 먼저 연습하세요.

저장과 개인정보

이름, 약관 동의와 진행 상황은 이 브라우저의 localStorage에 저장됩니다. 계정이나 클라우드 백업이 아닙니다. 비공개 탐색, 사이트 데이터 삭제, 저장 공간 제한으로 사라질 수 있습니다. 보관하거나 다른 기기로 옮기려면 백업을 다운로드하세요.

선택적 진단은 기본으로 꺼져 있고 설정에 별도 스위치가 있습니다. 플레이에 필수는 아닙니다. 이름, 연습 내용, 전체 저장 파일이 아닌 제한된 기술 오류 데이터만 포함합니다. 자세한 데이터 처리는 약관을 확인하세요.

키 표기 읽기

대소문자를 구분합니다. w와 W, p와 P, i와 I는 다른 명령입니다. 3w는 3 다음에 w를 누르며, Ctrl-r은 Control을 누른 채 r을 누릅니다. +, -, |, $, %, 세미콜론, 쉼표도 실제 명령 키입니다.

ciw는 연산자 c와 텍스트 객체 iw를 조합합니다. 안내 언어가 달라도 석판의 이름과 목표 내용은 보이는 그대로 입력하세요. : 명령과 / 또는 ? 검색은 Enter로 확인합니다. 순서대로 입력하되 단계를 구분하는 화살표는 입력하지 않습니다. <Esc>, <Enter>, <C-r>, <Space>는 Escape, Enter, Control+r, 공백 키이며 꺾쇠괄호는 입력하지 않습니다.

게임과 실제 Vim

LUMEN은 선택한 Vim 학습 범위를 다루는 교육용 시뮬레이터입니다. 보드는 텍스트 버퍼를, 보이는 부분은 화면 창을 나타냅니다. 스크롤 명령은 내용이 아니라 화면을 바꿉니다. 파일은 가상이며 컴퓨터의 폴더를 읽거나 운영체제 명령을 실행하지 않습니다.

명령은 학습하는 편집 동작을 따르지만, 모든 옵션과 플러그인, 구문 예외가 있는 실제 Vim을 대체하지는 않습니다. 게임 도움말과 이 학습서는 지원 범위를 설명합니다.

명령줄

콜론은 명령줄을 엽니다. :w는 가상 문서 저장, :e는 다른 문서 열기, :ls는 버퍼 목록, :b는 버퍼 전환입니다. :marks는 마크 목록을 보여 주고 :delmarks 또는 줄임말 :delm은 마크를 삭제합니다. :delmarks!는 로컬 마크를 모두 지웁니다. :reg는 레지스터 내용을 보여 줍니다.

:set nu와 :set nonu는 절대 행 번호를 켜고 끄며 :set nu!는 전환합니다. :set rnu, :set nornu, :set rnu!는 상대 행 번호를 제어합니다. Enter는 확인, Esc는 취소입니다.

레지스터

레지스터는 Vim의 클립보드입니다. 일반 복사와 삭제는 이름 없는 레지스터를 쓰며, 큰따옴표 접두사로 다른 레지스터를 고릅니다. a-z는 이름 있는 레지스터이고 A-Z에 쓰면 해당 소문자 레지스터 뒤에 덧붙입니다.

0은 마지막 복사, 1-9는 큰 삭제의 이력, -는 작은 삭제를 보관합니다. _는 블랙홀 레지스터로, 다른 클립보드를 덮어쓰지 않고 삭제합니다. :reg로 실제 저장된 내용을 확인하세요.

매크로

매크로는 커서 이동을 포함한 동작의 연속을 기록합니다. 레지스터 a에 녹화하려면 qa를 누르고 동작을 수행한 뒤 q로 끝내세요. @a는 a를 재생하고 @@는 마지막으로 재생한 매크로를 반복합니다. 앞의 숫자로 반복 횟수를 지정합니다.

여러 번 재생하기 전에 한 번 실행해 보세요. 좋은 매크로는 다음 내용을 처리할 준비가 된 예측 가능한 위치에서 끝납니다.

가상 파일과 버퍼

버퍼는 열려 있는 문서이고 파일 이름은 가상 저장본을 식별합니다. :w [이름]은 저장, :e 이름은 열기, :ls는 목록, :b 이름 또는 번호는 전환입니다. 문서를 떠나기 전에 필요한 변경을 저장하세요.

모든 작업은 게임과 로컬 저장 안에서 이루어집니다. 여정 내보내기는 백업이지 컴퓨터 파일 시스템에 실제 소스 파일을 저장하는 기능이 아닙니다. 비밀번호나 기밀 정보를 연습에 입력하지 마세요.

<Esc>는 Escape, <Enter>는 Enter, <C-r>은 Ctrl+r, <Space>는 공백입니다. 화살표는 조각을 구분할 뿐 입력하는 키가 아닙니다. 대소문자를 구분하세요.

명령 도감

명령	동작
`0`	현재 행의 첫 번째 열로 이동합니다.
`hjkl`	h 왼쪽, j 아래, k 위, l 오른쪽. 기본 커서 이동입니다.
`h`	왼쪽으로 한 칸 또는 한 문자 이동합니다.
`j`	아래로 한 행 또는 한 칸 이동합니다.
`k`	위로 한 행 또는 한 칸 이동합니다.
`l`	오른쪽으로 한 칸 또는 한 문자 이동합니다.
`w`	다음 단어의 시작으로 이동합니다.
`b`	뒤쪽 단어의 시작으로 이동합니다.
`e`	단어의 끝으로 이동합니다.
`^`	현재 행에서 공백이 아닌 첫 문자로 이동합니다.
`$`	현재 행의 마지막 문자로 이동합니다.
`gg`	첫 번째 행으로 이동합니다.
`G`	마지막 행으로 이동합니다.
`count`	숫자는 다음 움직임의 반복 횟수입니다. 예: 3w, 2j.
`f`	f 다음에 문자를 눌러 현재 행에서 그 문자가 다음으로 나오는 곳을 찾습니다.
`;`	f로 한 마지막 문자 찾기를 반복합니다.
`/`	내용을 앞으로 검색하고 Enter로 확인합니다.
`n`	기존 검색 방향으로 검색을 반복합니다.
`N`	마지막 / 또는 ? 검색의 반대 방향으로 반복합니다.
`x`	커서 아래 문자를 삭제합니다.
`r`	커서 아래 문자를 다음에 입력하는 문자로 바꿉니다.
`dd`	현재 행을 잘라 레지스터에 보관합니다.
`dw`	커서에서 다음 단어의 시작 전까지 삭제합니다.
`u`	마지막 내용 변경을 취소합니다.
`Ctrl-r`	취소한 변경을 다시 실행합니다.
`yy`	현재 행 전체를 레지스터에 복사합니다.

명령	동작
yw	커서에서 다음 단어의 시작 전까지 복사합니다.
p	내용은 커서 뒤에, 행 전체는 현재 행 아래에 붙입니다.
i	커서 앞에서 입력을 시작합니다.
a	커서 뒤에서 입력을 시작합니다.
A	현재 행 끝에서 입력을 시작합니다.
o	아래에 새 행을 열고 입력을 시작합니다.
Esc	일반 모드로 돌아가거나 대기 중인 명령을 취소합니다.
:w	가상 문서를 저장합니다. 선택한 이름은 게임 안의 파일을 가리킵니다.
1 ... 9	연산자나 이동 앞의 숫자는 횟수입니다. 2d3w는 두 횟수를 결합합니다.
W	공백으로 구분되는 다음 WORD의 시작으로 갑니다. 문장부호도 WORD에 포함됩니다.
E	공백으로 구분되는 WORD의 끝으로 이동합니다.
B	문장부호로 나누지 않고 이전 WORD의 시작으로 갑니다.
ge	이전 단어의 끝으로 이동합니다.
gE	이전 WORD의 끝으로 이동합니다.
+	다음 행의 공백이 아닌 첫 문자로 갑니다. 횟수를 붙일 수 있습니다.
-	이전 행의 공백이 아닌 첫 문자로 갑니다.
|	5|처럼 지정한 열로 갑니다. 숫자가 없으면 첫 번째 열입니다.
F	F 다음에 문자를 눌러 현재 행에서 뒤로 찾습니다.
t	현재 행의 다음 지정 문자 바로 앞까지 이동합니다.
T	뒤로 문자를 찾아 그 문자 바로 뒤에 멈춥니다.
,	마지막 f, F, t, T를 반대 방향으로 반복합니다.
*	커서 아래의 완전한 단어를 앞으로 검색합니다.
#	커서 아래의 완전한 단어를 뒤로 검색합니다.
?	뒤로 검색을 시작하고 Enter로 확인합니다.
%	짝이 되는 (), [], {} 구분자로 이동합니다.
[{	바깥 블록의 대응되지 않은 여는 중괄호로 돌아갑니다.
]}	바깥 블록의 대응되지 않은 닫는 중괄호로 갑니다.
[(	바깥의 대응되지 않은 여는 소괄호로 돌아갑니다.

명령	동작
`])`	바깥의 대응되지 않은 닫는 소괄호로 갑니다.
`{`	빈 행으로 나뉘는 문단의 이전 경계로 갑니다.
`}`	다음 문단 경계로 이동합니다.
`(`	이전 문장의 시작으로 이동합니다.
`)`	다음 문장의 시작으로 이동합니다.
`zt`	현재 행이 화면 맨 위에 오도록 스크롤합니다.
`zz`	내용을 바꾸지 않고 현재 행을 화면 가운데로 옮깁니다.
`zb`	현재 행이 화면 맨 아래에 오도록 스크롤합니다.
`H`	화면에 보이는 첫 행으로 이동합니다.
`M`	화면에 보이는 가운데 행으로 이동합니다.
`L`	화면에 보이는 마지막 행으로 이동합니다.
`nu / :set nu`	절대 행 번호를 켭니다.
`nonu / :set nonu`	절대 행 번호를 끕니다.
`nu! / :set nu!`	절대 행 번호 표시를 전환합니다.
`rnu / :set rnu`	커서 행과의 거리를 나타내는 상대 행 번호를 켭니다.
`nornu / :set nornu`	상대 행 번호를 끕니다.
`rnu! / :set rnu!`	상대 행 번호 표시를 전환합니다.
`X`	커서 앞 문자를 삭제합니다.
`D`	커서부터 행 끝까지 삭제합니다.
`~`	커서 아래 글자의 대소문자를 바꿉니다.
`I`	행의 공백이 아닌 첫 문자에서 입력을 시작합니다.
`O`	현재 행 위에 새 행을 열고 입력 모드로 들어갑니다.
`s`	커서 아래 문자를 지우고 입력을 시작합니다.
`S`	현재 행 전체를 바꾸기 위해 입력 모드로 들어갑니다.
`d`	삭제 연산자입니다. d$, diw, d2w처럼 이동이나 객체를 붙입니다.
`y`	복사 연산자입니다. 이동이나 텍스트 객체와 조합합니다.
`c`	지정 범위를 지우고 입력을 시작하는 변경 연산자입니다.
`cc`	현재 행 전체를 바꾸고 입력을 시작합니다.

명령	동작
C	커서부터 행 끝까지 바꾸고 입력을 시작합니다.
Y	현재 행 전체를 복사합니다. 이 게임에서는 yy와 같습니다.
.	입력한 내용을 포함해 마지막 편집을 반복합니다.
iw	주변 공백 없는 단어 내부입니다. d, y, c와 조합합니다.
aw	인접 공백을 포함한 단어 객체입니다.
iW	문장부호를 포함한 연속된 비공백 WORD 내부입니다.
aW	주변 공백을 포함한 WORD입니다.
is	구분 공백을 제외한 문장 내부입니다.
as	구분 공백을 포함한 문장입니다.
ip	주변 빈 행을 제외한 문단 내부입니다.
ap	구분하는 빈 행을 포함한 문단입니다.
i[/ i]	짝이 되는 [] 안쪽이며 괄호는 제외합니다. 두 별칭은 같습니다.
a[/ a]	괄호를 포함한 [] 전체입니다.
i(/ i) / ib	괄호를 제외한 () 내부입니다. ib도 같은 뜻입니다.
a(/ a) / ab	괄호를 포함한 () 전체입니다. ab도 같은 뜻입니다.
i{ / i} / iB	중괄호를 제외한 {} 내부입니다. iB도 같은 뜻입니다.
a{ / a} / aB	중괄호를 포함한 {} 전체입니다. aB도 같은 뜻입니다.
i< / i>	꺾쇠괄호를 제외한 <> 내부입니다.
a< / a>	꺾쇠괄호를 포함한 <> 전체입니다.
i"	큰따옴표 안의 내용이며 따옴표는 제외합니다.
a"	주변 큰따옴표를 포함한 내용입니다.
i'	작은따옴표를 제외한 내부 내용입니다.
a'	주변 작은따옴표를 포함한 내용입니다.
i`	백틱을 제외한 내부 내용입니다.
a`	주변 백틱을 포함한 내용입니다.
it	태그를 제외한 대응 HTML 태그 쌍 내부입니다.
at	여는 태그와 닫는 태그를 포함한 요소 전체입니다.
m	m과 문자로 현재 위치를 마크합니다. 예: ma.

명령	동작
'	작은따옴표와 마크 문자로 해당 행의 첫 비공백 문자로 갑니다.
`	백틱과 마크 문자로 저장한 정확한 위치로 이동합니다.
:marks	저장된 마크를 표시합니다.
:delmarks / :delm	지정 마크를 지웁니다. 예: :delmarks a. :delm은 줄임말입니다.
:delmarks!	모든 로컬 마크를 지웁니다.
"	큰따옴표, 레지스터 이름, 연산자 또는 p 순으로 선택합니다.
""	이름 없는 레지스터는 최근 복사하거나 삭제한 내용을 보관합니다.
"-	작은 삭제 레지스터는 행 전체보다 작은 삭제를 보관합니다.
"_	블랙홀: 다른 레지스터를 덮어쓰지 않고 삭제합니다.
"0	0은 나중에 삭제해도 마지막 복사 내용을 보관합니다.
:reg / :registers	레지스터를 표시합니다. :reg a0처럼 이름을 지정할 수 있습니다.
:e	지정 이름의 가상 파일을 엽니다.
:ls	열린 버퍼와 번호를 표시합니다.
:b	이름이나 번호로 가상 버퍼를 전환합니다.
q	q와 문자로 매크로 녹화를 시작하고 다시 q로 끝냅니다.
@	@a처럼 지정 레지스터의 매크로를 재생합니다. 횟수를 붙일 수 있습니다.
@@	마지막 사용한 매크로를 다시 재생합니다.
"1 ... "9	큰 삭제의 이력입니다. 1이 최신, 9가 보관된 것 중 가장 오래된 내용입니다.
"a ... "z	이름 있는 a-z 레지스터입니다. 연산자 앞에서 문자로 선택합니다.
"A ... "Z	A-Z는 해당 a-z 레지스터를 바꾸지 않고 뒤에 덧붙입니다.
P	내용을 커서 앞에 붙이거나 행 전체를 현재 행 위에 놓습니다.
:set	표시 옵션을 바꿉니다. 예: :set nu, :set rnu!.
:	명령줄을 열어 명령을 입력하고 Enter로 확인합니다.
z	스크롤 접두사입니다. t, z, b로 완성합니다.
:undo	마지막 편집을 취소합니다. u의 명령줄 형태입니다.
:redo	취소한 편집을 재실행합니다. Ctrl-r의 명령줄 형태입니다.
:delm!	:delmarks!의 줄임말로 모든 로컬 마크를 지웁니다。

섬별 안내

먼저 직접 시도해 보세요. 해답은 예시이며 다른 올바른 키 순서로도 목표를 달성할 수 있습니다.

<Esc>는 Escape, <Enter>는 Enter, <C-r>은 Ctrl+r, <Space>는 공백입니다. 화살표는 조각을 구분할 뿐 입력하는 키가 아닙니다. 대소문자를 구분하세요.

01 / 등불 항구

모든 여정은 한 번의 움직임에서 시작됩니다

파도가 모든 등불이 꺼진 항구로 당신을 데려왔습니다. 네리는 부두의 돌 틈에서 흩어진 불씨를 찾았습니다. 불씨를 모으면 세상에 첫 번째 길이 다시 드러납니다.

임무 01 / 첫 번째 불씨

l로 오른쪽으로 움직여 불씨를 모은 뒤 k로 올라가 빛나는 문에 도착하세요.

시작: 행 4, 열 3

빛 모으기: (4, 6). 커서 도착점: (2, 6).

```
1      #########
2      #.......#
3      #.......#
4      #.......#
5      #########
```

연습할 명령: l, k

힌트: l은 오른쪽, k는 위로 이동합니다. 이번 임무에서는 두 키를 모두 사용하세요. 불씨를 모아도 여정은 끝나지 않습니다. 그 빛을 빛나는 문까지 가져가세요.

가능한 한 가지 경로

`lllkk`

임무 02 / 밀물의 계단

h와 j로 위쪽 길을 벗어나 불씨를 모은 뒤 왼쪽 문에 도착하세요.

시작: 행 2, 열 8

빛 모으기: (4, 6). 커서 도착점: (4, 3).

```
1      #########
2      #.......#
3      #.......#
4      #.......#
5      #########
```

연습할 명령: h, j

힌트: h는 왼쪽, j는 아래로 이동합니다. 두 움직임이 모두 필요합니다. 불씨에 가까이 간 뒤 j로 내려가 모으세요. 되찾은 빛을 문까지 가져가세요.

가능한 한 가지 경로

`hhjjhhh`

임무 03 / 부두를 밝히다

남은 불씨를 모두 모으세요. 빛이 돌아오면 항구가 다음 섬을 가리켜 줍니다.

시작: 행 2, 열 2

빛 모으기: (2, 8), (6, 8), (6, 2). 커서 도착점: (4, 4).

```
#########
#.......#
#.#####.#
#...#...#
#.#####.#
#.......#
#########
```

힌트: 움직이기 전에 길 전체와 빛의 위치를 살펴보세요. 길이 막히면 몇 칸 돌아가도 됩니다. 서두를 필요 없이 천천히 시도해 보세요.

가능한 한 가지 경로

`lllllljjjjhhhhhhkkll`

02 / 단어의 숲

한 칸이 아니라 단어로 생각하세요

이 숲의 나무는 단어에서 자랍니다. 고요의 밀물이 뿌리를 흩뜨린 뒤, 이보는 문장 사이를 건널 수 없게 되었습니다. 이보가 단어 전체의 시작과 끝을 찾는 움직임을 알려 줍니다.

임무 04 / 뿌리에서 뿌리로

w로 단어의 시작 사이를 건너뛰어 표시된 위치에 도착하세요.

시작: 행 1, 열 1

커서 도착점: 행 1, 열 10.

```
1      dew moss lantern river
```

연습할 명령: w

힌트: w는 다음 단어의 시작으로 이동합니다. 공백은 단어로 이루어진 섬 사이의 틈입니다. w는 그 틈을 한 번에 건넙니다.

가능한 한 가지 경로

ww

임무 05 / 뒤에 남은 발자국

앞서 지나온 단어를 찾으세요. b로 기록의 길을 거슬러 돌아갑니다.

시작: 행 1, 열 10

커서 도착점: 행 1, 열 5.

```
1      dew moss lantern river
```

연습할 명령: b

힌트: b는 뒤쪽 단어의 시작으로 이동합니다. 단어 중간에 있다면 첫 번째 b는 그 단어의 시작으로 갑니다. 다시 누르면 더 앞의 단어로 돌아갑니다.

가능한 한 가지 경로

b

임무 06 / 가지 끝에서

알맞은 단어의 마지막 글자에 멈추세요. e를 사용해 봅니다.

시작: 행 1, 열 1

커서 도착점: 행 1, 열 11.

```
1      soft silver light
```

연습할 명령: e

힌트: e는 단어의 끝으로 이동합니다. 빛나는 목표를 보세요. 단어의 시작과 끝은 다른 위치입니다. 정확히 도착할 수 있는 움직임을 고르세요.

가능한 한 가지 경로

we

03 / 행과 행 사이의 다리

한쪽 끝에서 반대편 끝까지

돌다리들은 시작과 끝만 기억합니다. 셀라는 다리의 가장자리를 찾고 움직임에 횟수를 더하는 법을 알려 줍니다. 먼 길이라고 언제나 키를 여러 번 따로 눌러야 하는 것은 아닙니다.

임무 07 / 첫 번째 돌

0으로 행의 시작에 간 뒤 ^로 첫 글자에 도착하세요.

시작: 행 1, 열 10

커서 도착점: 행 1, 열 4.

```
1         stone bridge
```

연습할 명령: 0, ^

힌트: 0은 공백이 있어도 첫 번째 열로 갑니다. ^는 앞의 공백을 건너뜁니다. 0 다음에 ^를 누르세요. 목표는 첫 글자이지만 이번에는 두 시작점을 모두 연습합니다.

가능한 한 가지 경로

```
0^
```

임무 08 / 건너편 강둑

G로 마지막 행에 간 뒤 $로 그 끝까지 이동하세요.

시작: 행 1, 열 1

커서 도착점: 행 3, 열 16.

```
1     a quiet river
2     under the bridge
3     the last lantern
```

연습할 명령: G, $

힌트: G는 기록 전체의 마지막 행으로, $는 현재 행의 마지막 문자로 갑니다. G 다음에 $를 사용하세요. 한 행씩 내려갈 필요가 없습니다.

가능한 한 가지 경로

```
G$
```

임무 09 / 커다란 한 걸음

gg로 맨 위에 돌아가고 3j로 세 행 아래로 간 뒤 0으로 첫 번째 열을 고르세요.

시작: 행 5, 열 5

커서 도착점: 행 4, 열 1.

```
1     north gate
2     mist
3     old steps
4     star bridge
5     last lantern
```

연습할 명령: gg, count, 0

힌트: gg는 첫 행으로 갑니다. j 앞의 숫자 3은 세 번 반복하라는 뜻입니다. gg, 3j, 0 순서로 사용하세요. 숫자는 입력할 내용이 아니라 명령의 일부입니다.

가능한 한 가지 경로

`gg3j0`

04 / 메아리 관측소

한눈에 보이지 않는 것을 찾으세요

섬 위의 별은 여전히 빛나지만, 별의 이름은 여러 석판에 흩어졌습니다. 오린은 돌아오는 메아리를 듣습니다. 문자와 단어를 찾는 법을 배우면 그 사이의 길을 읽을 수 있습니다.

임무 10 / 선택한 별 하나

f를 사용해 현재 행에서 문자 l을 찾으세요.

시작: 행 1, 열 1

커서 도착점: 행 1, 열 8.

```
1      moss | lantern | stars
```

연습할 명령: f

힌트: f는 찾을 문자 입력을 기다립니다. 이번 임무에서는 l입니다. f 다음에 l을 누르세요. 모든 글자를 지나가지 않고 다음 l에 도착합니다.

가능한 한 가지 경로

```
fl
```

임무 11 / 문자의 메아리

fb로 다음 b를 찾고 세미콜론 ;로 같은 찾기를 반복하세요.

시작: 행 1, 열 1

커서 도착점: 행 1, 열 11.

```
1      bell bell bell
```

연습할 명령: f, ;

힌트: f 다음에 b를 누르세요. ;는 마지막 단일 문자 찾기를 반복합니다. fb 다음에 ;를 사용하면 같은 행의 또 다른 b에 도착합니다.

가능한 한 가지 경로

```
fb;
```

임무 12 / 돌아오는 메아리

/light와 Enter로 light를 검색하세요. n, N, 다시 n을 사용해 마지막 석판의 메아리에 도착하세요.

시작: 행 1, 열 1

커서 도착점: 행 3, 열 10.

```
1      light sleeps
2      follow light
3      wake the light
```

연습할 명령: /, n, N

힌트: n은 같은 방향으로 다음 결과를, N은 반대 방향으로 검색합니다. /light를 입력하고 Enter를 누른 뒤 n, N, n을 누르세요. 두 검색 방향을 모두 연습합니다.

가능한 한 가지 경로

```
/light → <Enter> → nNn
```

05 / 불씨 대장간

과감히 고치세요. 변경은 되돌릴 수 있습니다

화로에는 새 글자를 벼려 낼 만큼의 온기가 남아 있습니다. 아샤는 완벽한 움직임을 요구하지 않습니다. 좋은 장인은 실수를 되돌리는 법도 아니까요. 손상된 기록을 고쳐 불을 되살리세요.

임무 13 / 기록에 박힌 파편

x로 불필요한 문자 하나를 지워 올바른 기록을 되찾으세요.

시작: 행 1, 열 6

시작 텍스트

```
1  sparkx
```

목표 텍스트

```
1  spark
```

연습할 명령: x

힌트: x는 커서 아래의 문자를 삭제합니다. 먼저 커서를 불필요한 글자 위에 놓으세요. 삭제한 뒤에는 커서 위치뿐 아니라 전체 내용을 확인하세요.

가능한 한 가지 경로

x

임무 14 / 금속의 기억

rg로 첫 c를 g로 바꾸고 u로 취소한 뒤 Ctrl-r로 다시 실행하세요.

시작: 행 1, 열 1

시작 텍스트

```
1  cold ember
2  cold ember
```

목표 텍스트

```
1  gold ember
2  cold ember
```

연습할 명령: r, u, Ctrl-r

힌트: r은 바꿀 문자 입력을 기다립니다. rg는 cold를 gold로 만듭니다. rg 다음에 u와 Ctrl-r을 사용하세요. 목표는 첫 행 gold ember, 둘째 행 cold ember입니다.

가능한 한 가지 경로

`rg → u → <C-r>`

임무 15 / 새로운 모양

dw로 첫 broken을 지우고 dd로 남은 손상된 행을 잘라 내세요. dawn만 남기면 됩니다.

시작: 행 1, 열 1

시작 텍스트

```
1      broken broken
2      dawn
```

목표 텍스트

```
1      dawn
```

연습할 명령: dw, dd

힌트: dw는 다음 단어까지 내용을 지우고 dd는 현재 행 전체를 지웁니다. 먼저 dw, 다음은 dd입니다. 다른 크기의 손상에는 다른 도구가 필요합니다.

가능한 한 가지 경로

`dwdd`

06 / 거울 도서관

좋은 생각에는 메아리가 어울립니다

책장 위의 책은 물 위에 페이지를 비춥니다. 미라는 단어가 사라지기 전에 간직하는 법을 압니다. 레지스터를 이용하면 작은 빛을 복사하거나 행 전체를 새로운 곳으로 옮길 수 있습니다.

임무 16 / 쌍둥이 행

yy로 행을 복사하고 p로 그 아래에 같은 행을 놓으세요.

시작: 행 1, 열 1

시작 텍스트

```
1     light
2     sleep
```

목표 텍스트

```
1     light
2     light
3     sleep
```

연습할 명령: yy, p

힌트: yy는 원본을 지우지 않고 현재 행 전체를 복사합니다. 행을 복사한 뒤 p를 누르면 현재 행 아래에 붙여 넣습니다. 먼저 복사본을 놓을 위치를 고르세요.

가능한 한 가지 경로

```
yyp
```

임무 17 / 거울 속의 단어

yw로 단어를 간직한 뒤 p로 알맞은 곳에 놓으세요.

시작: 행 1, 열 1

시작 텍스트

```
1     glow moss
```

목표 텍스트

```
1     glow glow moss
```

연습할 명령: yw, p

힌트: yw는 커서에서 다음 단어의 시작 전까지 복사합니다. 문자열 조각을 복사했다면 p는 커서 뒤에 붙여 넣습니다. 행 전체를 붙이는 방식과 다릅니다.

가능한 한 가지 경로

```
ywwhp
```

임무 18 / 새로운 순서

dd로 행을 잘라 낸 뒤 p로 옮겨 올바른 순서를 되찾으세요.

시작: 행 1, 열 1

시작 텍스트

```
1     mist
2     sun
3     sky
```

목표 텍스트

```
1     sun
2     mist
3     sky
```

연습할 명령: dd, p

힌트: dd는 지운 행을 p가 사용하는 같은 레지스터에 보관합니다. 잘라 낸 뒤 목적지로 이동하세요. p는 기억한 행을 현재 행 아래에 놓습니다.

가능한 한 가지 경로

```
ddp
```

07 / 새벽의 서고

새로운 기록을 남겨 보세요

에노가 연대기에서 빈 곳을 발견했습니다. 움직임만으로는 이 단어들을 되찾을 수 없습니다. 누군가 다시 써야 합니다. 이곳에서는 내용을 다루는 것과 입력하는 것 사이의 리듬을 배웁니다.

임무 19 / 빠진 단어

커서를 놓고 i로 입력 모드에 들어가 빠진 내용을 입력하세요.

시작: 행 1, 열 5

시작 텍스트

```
1     the returns
```

목표 텍스트

```
1     the dawn returns
```

연습할 명령: i, Esc

힌트: i는 커서 앞에서 입력을 시작합니다. 입력 모드에서는 누른 글자가 내용이 됩니다. Esc를 누르면 이동 명령을 쓰는 일반 모드로 돌아갑니다.

가능한 한 가지 경로

`i → dawn<Space> → <Esc>`

임무 20 / 더 쓸 수 있는 자리

a로 s를 더하고 Esc로 돌아오세요. 그다음 A로 행 끝에 공백과 shine을 붙이세요.

시작: 행 1, 열 4

시작 텍스트

```
1     star
```

목표 텍스트

```
1     stars shine
```

연습할 명령: a, A, Esc

힌트: a는 커서 뒤에서, A는 행 전체의 끝에서 입력합니다. a, s, Esc로 star를 stars로 만드세요. 이어서 A, 공백, shine, Esc로 stars shine을 완성합니다.

가능한 한 가지 경로

`as → <Esc> → A<Space>shine → <Esc>`

임무 21 / 연대기의 인장

o로 새 행을 열고 returns를 쓰세요. Esc로 나온 뒤 :w와 Enter로 저장하세요.

시작: 행 1, 열 1

시작 텍스트

```
1  dawn
```

목표 텍스트

```
1  dawn
2  returns
```

연습할 명령: o, Esc, :w

힌트: o는 아래에 새 행을 열고 바로 입력 모드로 들어갑니다. dawn 아래에 returns를 쓰세요. Esc, :w, Enter로 완성한 석판을 봉인합니다.

가능한 한 가지 경로

`oreturns → <Esc> → :w → <Enter>`

08 / 아침의 탑

모든 길은 빛을 향합니다

첫 섬들의 조각으로 만든 렌즈가 탑 꼭대기에서 기다립니다. 배운 기술을 모아 첫 새벽을 불러오세요. 그 빛이 더 많은 지역과 숙련으로 가는 길을 드러냅니다.

임무 22 / 기억 속의 지도

dd로 손상된 첫 행을 잘라 내고 yy로 올바른 행을 복사한 뒤 p로 붙이고 :w로 확인하세요.

시작: 행 1, 열 1

시작 텍스트

```
1     lost lost
2     keep the light
```

목표 텍스트

```
1     keep the light
2     keep the light
```

연습할 명령: dd, yy, p, :w

힌트: dd 뒤에는 keep the light가 남습니다. yy로 행 전체를 기억하고 p로 아래에 복사합니다. dd, yy, p 다음에 :w와 Enter를 사용하세요. 같은 keep the light 행이 두 개 남아야 합니다.

가능한 한 가지 경로

`ddyyp:w → <Enter>`

임무 23 / 마지막 수선

/sleep과 Enter로 sleep을 찾고 dw로 지우세요. A로 sing을 더한 뒤 :w로 저장하세요.

시작: 행 1, 열 1

시작 텍스트

```
1     moss covers dawn
2     the bells sleep
```

목표 텍스트

```
1     moss covers dawn
2     the bells sing
```

연습할 명령: /, dw, A, :w

힌트: 검색하면 커서가 sleep에 놓입니다. dw로 그 단어를 지우고 A로 행을 마무리합니다. 첫 행은 그대로 두세요. 둘째 행은 the bells sing이어야 합니다. Esc로 입력을 마치고 :w와 Enter로 저장하세요.

가능한 한 가지 경로

`/sleep → <Enter> → dwAsing → <Esc> → :w → <Enter>`

임무 24 / 빛이 돌아오도록

첫 행을 지우고 남은 행을 복사한 뒤 새 행에 the dawn returns를 쓰세요. 복원한 연대기를 저장하세요.

시작: 행 1, 열 1

시작 텍스트

```
1     the night stays
2     one small spark
```

목표 텍스트

```
1     one small spark
2     one small spark
3     the dawn returns
```

연습할 명령: dd, yy, p, o, :w

힌트: dd는 손상된 행을 지웁니다. yy와 p는 두 번째 one small spark를 만듭니다. o는 마지막 행의 자리를 엽니다. dd, yy, p 다음에 o를 누르고 the dawn returns를 쓰세요. Esc, :w, Enter로 첫 새벽을 부르고 다음 길을 엽니다.

가능한 한 가지 경로

`ddyypothe<Space>dawn<Space>returns → <Esc> → :w → <Enter>`

09 / 바람의 절벽

단어보다 큰 단어

바람이 찢어진 글의 띠를 나릅니다. 타비는 글자를 하나씩 세지 않고 공백 없는 부분을 건너고 되돌아오는 법을 알려 줍니다.

임무 25 / 끊기지 않은 띠

W로 one-two 전체를 건너 three.four의 시작에 도착하세요. 필수 명령: W.

시작: 행 1, 열 1

커서 도착점: 행 1, 열 9.

```
1     one-two three.four five
```

연습할 명령: W

힌트: 공백으로 구분되는 다음 WORD의 시작으로 갑니다. 문장부호도 WORD에 포함됩니다. 먼저 한 동작을 의도적으로 실행하고 커서나 내용을 확인하세요. 단일 동작을 이해한 뒤 반복 횟수를 더하세요.

가능한 한 가지 경로

```
W
```

임무 26 / 능선을 따라 돌아오기

E와 B를 ge, gE와 비교한 뒤 첫 번째 열로 돌아오세요. 필수 명령: E · B · ge · gE.

시작: 행 1, 열 23

커서 도착점: 행 1, 열 1.

```
1     one-two three.four five
```

연습할 명령: E, B, ge, gE

힌트: 공백으로 구분되는 WORD의 끝으로 이동합니다. 문장부호로 나누지 않고 이전 WORD의 시작으로 갑니다.

가능한 한 가지 경로

```
BEgegE0
```

임무 27 / 헤아린 걸음

+와 - 및 횟수를 연습하고 열 이동으로 third의 t에 도착하세요. 필수 명령: + · - · | · 횟수.

시작: 행 1, 열 4

커서 도착점: 행 3, 열 2.

```
1      first
2        second
3      third
```

연습할 명령: +, -, |, count

힌트: 다음 행의 공백이 아닌 첫 문자로 갑니다. 횟수를 붙일 수 있습니다. 이전 행의 공백이 아닌 첫 문자로 갑니다.

가능한 한 가지 경로

```
+-2+2|
```

10 / 추적자의 고개

뒤쪽 길도 살펴보세요

작은 새김이 고개의 길을 알립니다. 린은 양방향으로 찾아 다음 안전한 돌의 정확한 자리에 멈추는 법을 가르칩니다.

임무 28 / 흔적을 거슬러

a를 뒤로 찾고 t와 T를 비교한 뒤 쉼표로 방향을 바꿔 마지막 a 바로 앞에 멈추세요. 필수 명령: F · t · T · ,.

시작: 행 1, 열 13

커서 도착점: 행 1, 열 12.

```
1     a x a y a z a
```

연습할 명령: F, t, T, ,

힌트: F 다음에 문자를 눌러 현재 행에서 뒤로 찾습니다. 현재 행의 다음 지정 문자 바로 앞까지 이동합니다.

가능한 한 가지 경로

`FataTa,`

임무 29 / 단어의 응답

*와 #으로 완전한 glow를 양방향으로 찾고 첫 행의 두 번째 glow에 멈추세요. 필수 명령: * · #.

시작: 행 1, 열 1

커서 도착점: 행 1, 열 11.

```
1     glow moss glow
2     glow
```

연습할 명령: *, #

힌트: 커서 아래의 완전한 단어를 앞으로 검색합니다. 커서 아래의 완전한 단어를 뒤로 검색합니다.

가능한 한 가지 경로

`*#*`

임무 30 / 반대 방향의 길

?로 star를 뒤로 검색하고 n과 N을 연습해 가운데 결과에 멈추세요. 필수 명령: ? · n · N.

시작: 행 3, 열 1

커서 도착점: 행 2, 열 1.

```
1     star
2     star
3     star
```

연습할 명령: ?, n, N

힌트: 뒤로 검색을 시작하고 Enter로 확인합니다. 기존 검색 방향으로 검색을 반복합니다.

가능한 한 가지 경로

`?star → <Enter> → nN`

11 / 돌의 고리

열림에는 닫힘이 있습니다

우마는 고리 속의 고리를 지킵니다. 괄호, 문장의 경계, 문단의 숨을 읽으면 돌들은 더 이상 미로가 아닙니다.

임무 31 / 짝을 이룬 돌

%와 블록 경계 이동으로 중첩 괄호를 살펴보고 가장 바깥의 여는 중괄호로 돌아오세요. 필수 명령: % · [(·]) · [{ ·]}.

시작: 행 2, 열 5

커서 도착점: 행 1, 열 1.

```
1     {
2       [a(b)c]
3     }
```

연습할 명령: %, [(,]), [{,]}

힌트: 짝이 되는 (), [], {} 구분자로 이동합니다. 바깥의 대응되지 않은 여는 소괄호로 돌아갑니다.

가능한 한 가지 경로

`% → [( → ]) → [{ → ]} → %`

임무 32 / 문단 사이

문단 사이를 앞뒤로 이동하고 north와 south 사이의 빈 행에 멈추세요. 필수 명령: { · }.

시작: 행 1, 열 1

커서 도착점: 행 3, 열 1.

```
1     north
2     path
3     [빈 행]
4     south
5     gate
6     [빈 행]
7     dawn
```

연습할 명령: {, }

힌트: 빈 행으로 나뉘는 문단의 이전 경계로 갑니다. 다음 문단 경계로 이동합니다.

가능한 한 가지 경로

`}}{`

임무 33 / 숨 쉴 공간

문장의 시작을 양방향으로 방문한 뒤 Second의 시작으로 돌아오세요. 필수 명령: (·).

시작: 행 1, 열 1

커서 도착점: 행 1, 열 14.

```
1     First light. Second light! Third light?
```

연습할 명령: (,)

힌트: 이전 문장의 시작으로 이동합니다. 다음 문장의 시작으로 이동합니다.

가능한 한 가지 경로

```
) ) (
```

12 / 수평선의 탑

내용이 아닌 시점을 바꾸세요

탑의 창은 두루마리를 따라 움직입니다. 노아는 커서의 위치와 지금 창에 보이는 내용의 차이를 보여 줍니다.

임무 34 / 창이 올라가다

세 창 위치를 시험한 뒤 현재 다섯 번째 행을 화면 맨 위에 놓으세요. 필수 명령: zt · zz · zb.

시작: 행 5, 열 1

커서 도착점: 행 5, 열 1.

```
1      line one
2      line two
3      line three
4      line four
5      line five
6      line six
7      line seven
8      line eight
9      line nine
```

맨 위에 보이는 행: 5

연습할 명령: zt, zz, zb

힌트: 현재 행이 화면 맨 위에 오도록 스크롤합니다. 내용을 바꾸지 않고 현재 행을 화면 가운데로 옮깁니다.

가능한 한 가지 경로

`ztzzzbzt`

임무 35 / 고정된 수평선

다섯 번째 행을 가운데 놓고 화면의 위아래를 방문한 뒤 가운데로 돌아오세요. 필수 명령: H · M · L.

시작: 행 5, 열 1

커서 도착점: 행 5, 열 1.

```
1      line one
2      line two
3      line three
4      line four
5      line five
6      line six
7      line seven
8      line eight
9      line nine
```

맨 위에 보이는 행: 3

연습할 명령: H, M, L

힌트: 화면에 보이는 첫 행으로 이동합니다. 화면에 보이는 가운데 행으로 이동합니다.

가능한 한 가지 경로

`zzHLM`

임무 36 / 곁에 있는 숫자

두 행 번호를 켜고 끄고 전환하세요. 마지막에는 절대 번호를 켜고 상대 번호를 끄세요. 필수 명령: nu · nonu · nu! · rnu · nornu · rnu!.

시작: 행 1, 열 1

커서 도착점: 행 1, 열 1.

```
1      north
2      south
```

절대 행 번호: 켜짐

상대 행 번호: 꺼짐

연습할 명령: nu, nonu, nu!, rnu, nornu, rnu!

힌트: 절대 행 번호를 켭니다. 절대 행 번호를 끕니다.

가능한 한 가지 경로

`:set<Space>nonu → <Enter> → :set<Space>nu → <Enter> → :set<Space>nu! → <Enter> → :set<Space>nu! → <Enter> → :set<Space>rnu → <Enter> → :set<Space>nornu → <Enter> → :set<Space>rnu! → <Enter> → :set<Space>rnu! → <Enter>`

13 / 직조공의 작업실

자르고 더하고 꿰매세요

아다는 글자로 짠 천을 고칩니다. 때로는 커서 앞, 때로는 행 위로 가고, 때로는 바느질 전체를 다시 시작해야 합니다.

임무 37 / 실 앞에서

커서 앞 x를 지우고 DAWN을 소문자로 바꾼 뒤 junk를 지우세요. dawn 뒤 공백은 남기세요. 필수 명령: X · ~ · D.

시작: 행 1, 열 2

시작 텍스트

```
1     xDAWN junk
```

목표 텍스트

```
1     dawn
```

연습할 명령: X, ~, D

힌트: 커서 앞 문자를 삭제합니다. 커서 아래 글자의 대소문자를 바꿉니다.

가능한 한 가지 경로

`X4~wD`

임무 38 / 자르고 꿰매기

I로 들여쓰기를 지킨 채 glow 앞에 the를 쓰고 O로 위에 header 행을 만드세요. 필수 명령: I · O.

시작: 행 1, 열 7

시작 텍스트

```
1        glow
```

목표 텍스트

```
1     header
2        the glow
```

연습할 명령: I, O

힌트: 행의 공백이 아닌 첫 문자에서 입력을 시작합니다. 현재 행 위에 새 행을 열고 입력 모드로 들어갑니다.

가능한 한 가지 경로

`Ithe<Space> → <Esc> → Oheader → <Esc>`

임무 39 / 새로운 겹

s로 x를 sun으로 바꾸고 S로 둘째 행 전체를 moon으로 바꾸세요. 필수 명령: s · S.

시작: 행 1, 열 1

시작 텍스트

```
1      x
2      bad line
```

목표 텍스트

```
1      sun
2      moon
```

연습할 명령: s, S

힌트: 커서 아래 문자를 지우고 입력을 시작합니다. 현재 행 전체를 바꾸기 위해 입력 모드로 들어갑니다.

가능한 한 가지 경로

`ssun → <Esc> → jSmoon → <Esc>`

14 / 문법 주조소

행동과 그 범위

케리스는 무엇을 할지, 무엇에 할지라는 두 부분으로 명령을 만듭니다. 연산자와 이동을 조합해 짧은 표현의 뜻을 발견하세요.

임무 40 / 동사와 길

c의 단어 변경, cc의 행 변경, C의 행 끝 변경을 비교해 warm sun과 fixed를 만드세요. 필수 명령: c · cc · C.

시작: 행 1, 열 1

시작 텍스트

```
1      cold moss
2      broken
```

목표 텍스트

```
1      warm sun
2      fixed
```

연습할 명령: c, cc, C

힌트: 지정 범위를 지우고 입력을 시작하는 변경 연산자입니다. 현재 행 전체를 바꾸고 입력을 시작합니다.

가능한 한 가지 경로

`cewarm → <Esc> → jccfixed → <Esc> → ggwCsun → <Esc>`

임무 41 / 보관하고 붙이기

y로 단어를 복사해 P로 커서 앞에 놓으세요. Y와 삭제도 연습하며 north는 지키세요. 필수 명령: y · Y · P · d.

시작: 행 1, 열 1

시작 텍스트

```
1      red blue
2      north
```

목표 텍스트

```
1      red red blue
2      north
```

연습할 명령: y, Y, P, d

힌트: 복사 연산자입니다. 이동이나 텍스트 객체와 조합합니다. 현재 행 전체를 복사합니다. 이 게임에서는 yy와 같습니다.

가능한 한 가지 경로

`ywwPYpdd`

임무 42 / 한 번 더

dw로 첫 bad를 지우고 두 번 반복해 light만 남기세요. 필수 명령: dw · ..

시작: 행 1, 열 1

시작 텍스트

```
1     bad bad bad light
```

목표 텍스트

```
1     light
```

연습할 명령: dw, .

힌트: 커서에서 다음 단어의 시작 전까지 삭제합니다. 입력한 내용을 포함해 마지막 편집을 반복합니다.

가능한 한 가지 경로

`dw..`

15 / 단어의 성소

거리 대신 의미를 고르세요

엘리는 글자 수로 내용을 재지 않습니다. 이곳에서 단어와 문장과 문단은 의도를 담을 준비가 된 그릇이 됩니다.

임무 43 / 단어의 중심

iw와 aw를 비교해 moss 내부를 지우고 blue와 인접 공백을 지워 red만 남기세요. 필수 명령: iw · aw.

시작: 행 1, 열 6

시작 텍스트

```
1      red moss blue
```

목표 텍스트

```
1      red
```

연습할 명령: iw, aw

힌트: 주변 공백 없는 단어 내부입니다. d, y, c와 조합합니다. 인접 공백을 포함한 단어 객체입니다.

가능한 한 가지 경로

`diwwdaw`

임무 44 / 더 큰 단어

moss.green을 하나의 WORD로 다루세요. iW와 aW를 연습해 red만 남기세요. 필수 명령: iW · aW.

시작: 행 1, 열 7

시작 텍스트

```
1      red moss.green blue
```

목표 텍스트

```
1      red
```

연습할 명령: iW, aW

힌트: 문장부호를 포함한 연속된 비공백 WORD 내부입니다. 주변 공백을 포함한 WORD입니다.

가능한 한 가지 경로

`diWwdaW`

임무 45 / 문장과 문단

문장 내부와 전체, 문단 내부를 복사하세요. 문장과 keep은 남기고 alpha/beta 문단과 간격을 지우세요. 필수 명령: is · as · ip · ap.

시작: 행 1, 열 1

시작 텍스트

```
1     One. Two.
2     [빈 행]
3     alpha
4     beta
5     [빈 행]
6     keep
```

목표 텍스트

```
1     One. Two.
2     [빈 행]
3     keep
```

연습할 명령: is, as, ip, ap

힌트: 구분 공백을 제외한 문장 내부입니다. 구분 공백을 포함한 문장입니다.

가능한 한 가지 경로

`yisyasjjyipdap`

16 / 구분자의 금고

안쪽과 그것을 감싼 틀

센은 상자 속의 상자를 지킵니다. 내용만 고르거나 틀까지 고르세요. 이름이 다른 열쇠가 같은 자물쇠를 열기도 합니다.

임무 46 / 네모와 원

모든 제시 별칭으로 []와 ()의 안쪽과 틀을 복사하세요. 마지막 이름 없는 레지스터에는 (glow) 전체를 남기세요. 필수 명령: i[· a[· i] · a] · i(· a(· i) · a) · ib · ab.

시작: 행 1, 열 3

커서 도착점: 행 1, 열 8.

```
1     [moss] (glow)
```

레지스터 ""

```
1     (glow)
```

연습할 명령: i[, a[, i], a], i(, a(, i), a), ib, ab

힌트: 짝이 되는 [] 안쪽이며 괄호는 제외합니다. 두 별칭은 같습니다. 괄호를 포함한 [] 전체입니다.

가능한 한 가지 경로

`yi[ya[yi]ya]f(yi(ya(yi)ya)yibyab`

임무 47 / 각도와 별칭

{}와 <>의 모든 별칭을 비교하고 틀을 포함한 <dawn> 전체를 레지스터에 남기세요. 필수 명령: i{ · a{ · i} · a} · iB · aB · i< · a< · i> · a>.

시작: 행 1, 열 3

커서 도착점: 행 1, 열 9.

```
1     {ember} <dawn>
```

레지스터 ""

```
1     <dawn>
```

연습할 명령: i{, a{, i}, a}, iB, aB, i<, a<, i>, a>

힌트: 중괄호를 제외한 {} 내부입니다. iB도 같은 뜻입니다. 중괄호를 포함한 {} 전체입니다. aB도 같은 뜻입니다.

가능한 한 가지 경로

`yi{ya{yi}ya}yiByaBf<yi<ya<yi>ya>`

임무 48 / 틀 전체

(seed) 안쪽만 bloom으로 바꿔 괄호를 지킨 뒤 [noise] 전체를 지우세요. 필수 명령: i(· a[.

시작: 행 1, 열 11

시작 텍스트

```
1     [noise] (seed) {ash} <sky>
```

목표 텍스트

```
1      (bloom) {ash} <sky>
```

연습할 명령: i(, a[

힌트: 괄호를 제외한 () 내부입니다. ib도 같은 뜻입니다. 괄호를 포함한 [] 전체입니다.

가능한 한 가지 경로

`ci(bloom → <Esc> → 0da[`

17 / 속삭임 예배당

기호 사이를 들어 보세요

유나는 따옴표와 태그 속에 메시지를 간직합니다. 내용을 들으려면 목소리와 그것을 감싼 틀을 구별해야 합니다.

임무 49 / 두 개의 작은 목소리

큰따옴표와 작은따옴표의 내부와 전체를 비교하고 마지막에 작은따옴표 포함 glow 전체를 복사하세요. 필수 명령: i" · a" · i' · a'.

시작: 행 1, 열 3

커서 도착점: 행 1, 열 8.

```
1     "moss" 'glow'
```

레지스터 ""

```
1     'glow'
```

연습할 명령: i", a", i', a'

힌트: 큰따옴표 안의 내용이며 따옴표는 제외합니다. 주변 큰따옴표를 포함한 내용입니다.

가능한 한 가지 경로

`yi"ya"f'yi'ya'`

임무 50 / 속삭임과 태그

백틱 내부와 전체를 복사한 뒤 HTML 요소도 복사해 glow 전체를 남기세요. 필수 명령: i` · a` · it · at.

시작: 행 1, 열 3

커서 도착점: 행 1, 열 8.

```
1     `seed` <b>glow</b>
```

레지스터 ""

```
1     <b>glow</b>
```

연습할 명령: i`, a`, it, at

힌트: 백틱을 제외한 내부 내용입니다. 주변 백틱을 포함한 내용입니다.

가능한 한 가지 경로

`yi`ya`f<yityat`

임무 51 / 겹쳐진 메시지

p 태그는 남기고 내용만 new로 바꾼 뒤 따옴표 포함 noise 전체를 지우세요. 끝 공백은 남기세요. 필수 명령: it · a".

시작: 행 1, 열 5

시작 텍스트

```
1      <p>old</p> "noise"
```

목표 텍스트

```
1      <p>new</p>
```

연습할 명령: it, a"

힌트: 태그를 제외한 대응 HTML 태그 쌍 내부입니다. 주변 큰따옴표를 포함한 내용입니다.

가능한 한 가지 경로

`citnew → <Esc> → f"da"`

18 / 지도 제작자의 야영지

돌아올 수 있는 점을 남기세요

아로는 바람에도 흩어지지 않는 지도를 그립니다. 위치를 기억하고 행이나 정확한 열로 돌아온 뒤 낡은 표지를 지우세요.

임무 52 / 표지 세우기

둘째 행에 마크 a를 놓고 행 이동과 정확한 열 이동을 비교한 뒤 저장 위치로 돌아오세요. 필수 명령: m · ' · `.

시작: 행 2, 열 5

커서 도착점: 행 2, 열 5.

```
1       north
2       south
3       dawn
```

마크 a: 행 2, 열 5

연습할 명령: m, ', `

힌트: m과 문자로 현재 위치를 마크합니다. 예: ma. 작은따옴표와 마크 문자로 해당 행의 첫 비공백 문자로 갑니다.

가능한 한 가지 경로

`magg'agg`a`

임무 53 / 지도 읽기

이웃한 열에 a와 b를 놓고 마크 목록을 연 뒤 :delm으로 a만 지우세요. 필수 명령: :marks · :delm.

시작: 행 1, 열 1

커서 도착점: 행 1, 열 2.

```
1     path
```

마크 b: 행 1, 열 2

삭제된 마크: a

열린 목록: 마크

연습할 명령: :marks, :delm

힌트: 저장된 마크를 표시합니다. 지정 마크를 지웁니다. 예: :delmarks a. :delm은 줄임말입니다.

가능한 한 가지 경로

`malmb → :marks → <Enter> → :delm<Space>a → <Enter>`

임무 54 / 옛길 지우기

마크 범위 삭제, :delmarks!, :delm!을 연습하고 a, b, c를 모두 지우세요. 필수 명령: :delmarks · :delmarks! · :delm!.

시작: 행 1, 열 1

커서 도착점: 행 1, 열 2.

```
1     path
```

삭제된 마크: a, b, c

연습할 명령: :delmarks, :delmarks!, :delm!

힌트: 지정 마크를 지웁니다. 예: :delmarks a. :delm은 줄임말입니다. 모든 로컬 마크를 지웁니다.

가능한 한 가지 경로

`malmb → :delmarks<Space>a-b → <Enter> → ma → :delmarks! → <Enter> → mc → :delm! → <Enter>`

19 / 메아리의 서랍장

클립보드에도 이름이 있습니다

에스메는 메아리를 각각의 서랍에 보관합니다. 하나는 지키고 다른 곳에는 더하며, 중요한 내용을 덮지 않고 잡음을 보내세요.

임무 55 / 이름 없는 메아리

a를 지운 뒤 작은 삭제 레지스터에서 복원하고 명시적인 이름 없는 레지스터로 ab 전체를 복사하세요. 필수 명령: "- · "" · x · yy · P.

시작: 행 1, 열 1

목표 텍스트

```
1      ab
```

레지스터 ""

```
1      ab
```

레지스터 "-

```
1      a
```

연습할 명령: "-, "", x, yy, P

힌트: 작은 삭제 레지스터는 행 전체보다 작은 삭제를 보관합니다. 이름 없는 레지스터는 최근 복사하거나 삭제한 내용을 보관합니다.

가능한 한 가지 경로

`x"-P""yy`

임무 56 / 덮어쓰지 않고 더하기

a에 moss를 넣고 A로 glow를 덧붙인 뒤 두 행을 붙여 넣고 a의 내용을 확인하세요. 필수 명령: "a · "A · p.

시작: 행 1, 열 1

시작 텍스트

```
1      moss
2      glow
```

목표 텍스트

```
1      moss
2      moss
3      glow
4      glow
```

레지스터 "a

```
1      moss
2      glow
```

연습할 명령: "a, "A, p

힌트: 이름 있는 a-z 레지스터입니다. 연산자 앞에서 문자로 선택합니다. A-Z는 해당 a-z 레지스터를 바꾸지 않고 뒤에 덧붙입니다.

가능한 한 가지 경로

`"ayyj"Ayygg"ap`

임무 57 / 고요가 복사본을 지키다

a에 keep을 보관하고 trash를 블랙홀로 지운 뒤 보관한 복사본을 현재 행 위에 놓으세요. 필수 명령: "_ · "a · dd · P.

시작: 행 1, 열 1

시작 텍스트

```
1  keep
2  trash
```

목표 텍스트

```
1  keep
2  keep
```

레지스터 "a

```
1  keep
```

레지스터 ""

```
1  keep
```

연습할 명령: "_, "a, dd, P

힌트: 블랙홀: 다른 레지스터를 덮어쓰지 않고 삭제합니다. 이름 있는 a-z 레지스터입니다. 연산자 앞에서 문자로 선택합니다.

가능한 한 가지 경로

`"ayyj"_dd"aP`

20 / 아홉 기억의 방

기억에도 순서가 있습니다

노나는 가장 최근의 사라짐만 기억하지 않습니다. 이력의 서랍을 열면 삭제가 반드시 이별은 아니라는 것을 알게 됩니다.

임무 58 / 마지막 복사본

light를 복사하고 dust를 지운 뒤 0에서 복사본을 복원하세요. 1에는 dust가 남아야 합니다. 필수 명령: "0 · dd.

시작: 행 1, 열 1

시작 텍스트

```
1     light
2     dust
```

목표 텍스트

```
1     light
2     light
```

레지스터 "0

```
1     light
```

레지스터 "1

```
1     dust
```

연습할 명령: "0, dd

힌트: 0은 나중에 삭제해도 마지막 복사 내용을 보관합니다. 현재 행을 잘라 레지스터에 보관합니다.

가능한 한 가지 경로

```
yyjdd"0p
```

임무 59 / 이어지는 이별

아홉 행을 지우고 이력 레지스터 1부터 9를 확인한 뒤 가장 오래된 one을 keep 위에 복원하세요. 필수 명령: "1 · "2 · "3 · "4 · "5 · "6 · "7 · "8 · "9.

시작: 행 1, 열 1

시작 텍스트

```
1     one
2     two
3     three
4     four
5     five
6     six
7     seven
8     eight
9     nine
10    keep
```

목표 텍스트

```
1     one
2     keep
```

레지스터 "1

```
1     nine
```

레지스터 "2

```
1     eight
```

레지스터 "3

```
1     seven
```

레지스터 "4

```
1     six
```

레지스터 "5

```
1     five
```

레지스터 "6

```
1     four
```

레지스터 "7

```
1     three
```

레지스터 "8

```
1     two
```

레지스터 "9

```
1     one
```

연습할 명령: "1, "2, "3, "4, "5, "6, "7, "8, "9

힌트: 큰 삭제의 이력입니다. 1이 최신, 9가 보관된 것 중 가장 오래된 내용입니다. 큰 삭제의 이력입니다. 1이 최신, 9가 보관된 것 중 가장 오래된 내용입니다.

가능한 한 가지 경로

```
dd........ → "1pu → "2pu → "3pu → "4pu → "5pu → "6pu → "7pu → "8pu → "9P
```

임무 60 / 서랍 읽기

one을 a에 복사하고 레지스터 창을 열어 a와 0에 같은 행이 있는지 확인하세요. 필수 명령: :reg.

시작: 행 1, 열 1

커서 도착점: 행 1, 열 1.

```
1      one
2      two
```

레지스터 "0

```
1      one
```

레지스터 "a

```
1      one
```

열린 목록: 레지스터

연습할 명령: :reg

힌트: 레지스터를 표시합니다. :reg a0처럼 이름을 지정할 수 있습니다. 먼저 한 동작을 의도적으로 실행하고 커서나 내용을 확인하세요. 단일 동작을 이해한 뒤 반복 횟수를 더하세요.

가능한 한 가지 경로

```
"ayy → :reg → <Enter>
```

21 / 서기들의 항구

많은 페이지, 하나의 여정

테로는 가상의 항구 사이로 문서를 나릅니다. 간직할 것을 저장하고 이름을 놓치지 않은 채 열린 페이지 사이를 오가세요.

임무 61 / 기억한 페이지

notes.txt에 dawn rises를 저장하고 echo.txt를 awake로 고쳐 저장한 뒤 변경을 지키며 notes.txt로 돌아오세요. 필수 명령: :w · :e · c.

시작: 행 1, 열 1

시작 텍스트

```
1  dawn
```

목표 텍스트

```
1  dawn rises
```

활성 버퍼: notes.txt

저장된 가상 파일: notes.txt

```
1  dawn rises
```

저장된 가상 파일: echo.txt

```
1  awake
```

연습할 명령: :w, :e, c

힌트: 가상 문서를 저장합니다. 선택한 이름은 게임 안의 파일을 가리킵니다. 지정 이름의 가상 파일을 엽니다.

가능한 한 가지 경로

```
A<Space>rises → <Esc> → :w → <Enter> → :e<Space>echo.txt → <Enter> → ccawake → <Esc> → :w → <Enter> → :e<Space>notes.txt → <Enter>
```

임무 62 / 또 다른 항구

beta.txt를 열고 버퍼 목록을 표시한 뒤 이름으로 alpha.txt로 돌아오세요. 필수 명령: :e · :ls · :b.

시작: 행 1, 열 1

커서 도착점: 행 1, 열 1.

```
1  north
```

활성 버퍼: alpha.txt

열린 목록: 버퍼

연습할 명령: :e, :ls, :b

힌트: 지정 이름의 가상 파일을 엽니다. 열린 버퍼와 번호를 표시합니다.

가능한 한 가지 경로

```
:e<Space>beta.txt → <Enter> → :ls → <Enter> → :b<Space>alpha.txt → <Enter>
```

임무 63 / 버퍼 사이의 항해

final.txt로 복사본을 저장하고 열어 번호로 버퍼를 전환한 뒤 final.txt에서 끝내세요. 필수 명령: :w · :e · :b.

시작: 행 1, 열 1

목표 텍스트

```
1     light
```

활성 버퍼: final.txt

저장된 가상 파일: final.txt

```
1     light
```

연습할 명령: :w, :e, :b

힌트: 가상 문서를 저장합니다. 선택한 이름은 게임 안의 파일을 가리킵니다. 지정 이름의 가상 파일을 엽니다.

가능한 한 가지 경로

```
:w<Space>final.txt → <Enter> → :e<Space>final.txt → <Enter> → :b<Space>1 → <Enter> → :b<Space>2 → <Enter>
```

22 / 태엽 과수원

좋은 움직임이 많은 열매를 맺도록

오미는 과수원의 섬세한 기계를 돌봅니다. 믿을 만한 한 주기를 녹화하고 끝나는 위치를 확인한 뒤 다음 가지에도 적용하세요.

임무 64 / 작은 기계

a에 ! 추가와 아래 이동을 녹화하고 한 번 재생해 첫 두 dawn 행을 바꾸세요. 필수 명령: q · @.

시작: 행 1, 열 1

시작 텍스트

```
1	dawn
2	dawn
3	dawn
```

목표 텍스트

```
1	dawn!
2	dawn!
3	dawn
```

녹화된 매크로: a

연습할 명령: q, @

힌트: q와 문자로 매크로 녹화를 시작하고 다시 q로 끝냅니다. @a처럼 지정 레지스터의 매크로를 재생합니다. 횟수를 붙일 수 있습니다.

가능한 한 가지 경로

`qaA! → <Esc> → jq@a`

임무 65 / 과수원이 반복하다

행 앞에 +를 쓰고 아래로 이동하는 동작을 녹화해 @a와 @@를 사용하세요. 넷째 seed는 남기세요. 필수 명령: q · @ · @@.

시작: 행 1, 열 1

시작 텍스트

```
1  seed
2  seed
3  seed
4  seed
```

목표 텍스트

```
1  +seed
2  +seed
3  +seed
4  seed
```

녹화된 매크로: a

연습할 명령: q, @, @@

힌트: q와 문자로 매크로 녹화를 시작하고 다시 q로 끝냅니다. @a처럼 지정 레지스터의 매크로를 재생합니다. 횟수를 붙일 수 있습니다.

가능한 한 가지 경로

`qaI+ → <Esc> → jq@a@@`

임무 66 / 한 번 더, 정확하게

x 하나 삭제를 녹화하고 세 번 재생해 light만 남기세요. 필수 명령: q · @ · 횟수.

시작: 행 1, 열 1

시작 텍스트

```
1  xxxxlight
```

목표 텍스트

```
1  light
```

녹화된 매크로: a

연습할 명령: q, @, count

힌트: q와 문자로 매크로 녹화를 시작하고 다시 q로 끝냅니다. @a처럼 지정 레지스터의 매크로를 재생합니다. 횟수를 붙일 수 있습니다.

가능한 한 가지 경로

`qaxq3@a`

23 / 직조공의 시험

배운 무늬를 조합하세요

벨이 긴 무늬를 펼칩니다. 횟수, 반복 편집, 매크로는 다른 도구입니다. 하려는 일을 가장 잘 나타내는 것을 고르세요.

임무 67 / 두 개의 횟수

연산자와 이동 횟수를 결합한 2d3w로 여섯 단어를 지우고 keep을 남기세요. 필수 명령: 횟수 · dw.

시작: 행 1, 열 1

시작 텍스트

```
1      one two three four five six keep
```

목표 텍스트

```
1      keep
```

연습할 명령: count, dw

힌트: 연산자나 이동 앞의 숫자는 횟수입니다. 2d3w는 두 횟수를 결합합니다. 커서에서 다음 단어의 시작 전까지 삭제합니다.

가능한 한 가지 경로

`2d3w`

임무 68 / 같은 짜임

x를 지우고 :undo와 :redo로 취소·재실행한 뒤 점으로 삭제를 반복하세요. 필수 명령: :undo · :redo · .

시작: 행 1, 열 1

시작 텍스트

```
1      xxdawn
```

목표 텍스트

```
1      dawn
```

연습할 명령: :undo, :redo, .

힌트: 마지막 편집을 취소합니다. u의 명령줄 형태입니다. 취소한 편집을 재실행합니다. Ctrl-r의 명령줄 형태입니다.

가능한 한 가지 경로

`x → :undo → <Enter> → :redo → <Enter> → .`

임무 69 / 믿을 수 있는 무늬

ash 삭제와 아래 이동을 녹화해 두 번 반복하고 keep으로 0을 시험한 뒤 임시 복사본을 취소하세요. 필수 명령: q · @ · 횟수 · "0.

시작: 행 1, 열 1

시작 텍스트

```
1      ash dawn
2      ash dawn
3      ash dawn
4      keep
```

목표 텍스트

```
1      dawn
2      dawn
3      dawn
4      keep
```

레지스터 "0

```
1      keep
```

녹화된 매크로: a

연습할 명령: q, @, count, "0

힌트: q와 문자로 매크로 녹화를 시작하고 다시 q로 끝냅니다. @a처럼 지정 레지스터의 매크로를 재생합니다. 횟수를 붙일 수 있습니다.

가능한 한 가지 경로

`qadwjq2@ayy"0Pu`

24 / 시작의 정상

하나의 기록 속에 모든 군도가

루마가 모든 길이 만나는 곳에서 기다립니다. 새로운 힘 대신 기억과 정확함, 상상력을 모으는 마지막 일을 부탁합니다.

임무 70 / 잡음 없는 기록

태그 내용만 light로 바꾸고 [noise] 전체와 빈 행을 지운 뒤 저장하세요. 필수 명령: it · a[· dd · :w.

시작: 행 1, 열 5

시작 텍스트

```
1      <p>old</p>
2      [noise]
3      keep
```

목표 텍스트

```
1      <p>light</p>
2      keep
```

연습할 명령: it, a[, dd, :w

힌트: 태그를 제외한 대응 HTML 태그 쌍 내부입니다. 괄호를 포함한 [] 전체입니다.

가능한 한 가지 경로

`citlight → <Esc> → jda[dd → :w → <Enter>`

임무 71 / 정확한 귀환

A와 a로 두 위치를 기억하세요. 정확한 이동과 양방향 검색을 조합해 dawn으로 돌아오세요. 필수 명령: m · ` · / · ? · n.

시작: 행 1, 열 6

커서 도착점: 행 3, 열 6.

```
1      star north
2      star south
3      star dawn
```

마크 A: 행 1, 열 6

마크 a: 행 3, 열 6

연습할 명령: m, `, /, ?, n

힌트: m과 문자로 현재 위치를 마크합니다. 예: ma. 백틱과 마크 문자로 저장한 정확한 위치로 이동합니다.

가능한 한 가지 경로

`mA/dawn → <Enter> → ma`A?star → <Enter> → n`a`

임무 72 / 모든 빛, 하나의 하늘

sleep을 wake로 바꾸는 동작을 세 행에 적용하고 message.txt의 마지막 행을 dawn.txt로 복사해 저장하세요. 필수 명령: q · @ · @@ · :w · :e · yy · "0 · p.

시작: 행 1, 열 1

시작 텍스트

```
1      sleep
2      sleep
3      sleep
```

목표 텍스트

```
1      wake
2      wake
3      wake
4      the dawn returns
```

레지스터 "0

```
1      the dawn returns
```

활성 버퍼: dawn.txt

저장된 가상 파일: dawn.txt

```
1      wake
2      wake
3      wake
4      the dawn returns
```

저장된 가상 파일: message.txt

```
1      the dawn returns
```

녹화된 매크로: a

연습할 명령: q, @, @@, :w, :e, yy, "0, p

힌트: q와 문자로 매크로 녹화를 시작하고 다시 q로 끝냅니다. @a처럼 지정 레지스터의 매크로를 재생합니다. 횟수를 붙일 수 있습니다.

가능한 한 가지 경로

`qaccwake → <Esc> → jq@a@@ → :w → <Enter> → :e<Space>message.txt → <Enter> → yy → :e<Space>dawn.txt → <Enter> → "0p → :w → <Enter>`

계속 배우기

이 안내서는 게임에서 지원하는 교육용 일부 기능을 설명합니다. 가상 파일은 브라우저에만 있으며 기기의 파일 시스템에 접근하지 않습니다.

Vim documentation

Motions and text objects
https://vimhelp.org/motion.txt.html

Editing and registers
https://vimhelp.org/change.txt.html

Macros
https://vimhelp.org/usr_10.txt.html

Files and buffers
https://vimhelp.org/usr_22.txt.html

LUMEN



光の守り手の学習ガイド

24の地域、72の課題。Vimの移動と編集の言葉を学ぶ実践ガイドです。

24 の島 / 72 の課題 / すべて無料

日本語 / 版 0.3.0

本書はゲームが対応する教育用の一部機能を説明します。仮想ファイルはブラウザー内に保存され、端末のファイルシステムにはアクセスしません。

まず自分で試してみましょう。解答は一例です。ほかの正しいキー操作でも目標を達成できる場合があります。

目次

旅に出る前に 3

コマンド図鑑 5

島々の案内 10

01 / 灯籠の港 . . . 11
02 / 言葉の木立 . . . 13
03 / 行間の石橋 . . . 14
04 / こだまの天文台 . . . 16
05 / 残り火の鍛冶場 . . . 17
06 / 鏡の図書館 . . . 19
07 / 夜明けの書斎 . . . 21
08 / 朝の塔 . . . 23
09 / 風の崖 . . . 25
10 / 追跡者の峠 . . . 26
11 / 石の輪 . . . 27
12 / 地平線の塔 . . . 29
13 / 織り手の工房 . . . 31
14 / 文法の鋳造所 . . . 33
15 / 言葉の聖域 . . . 35
16 / 区切りの宝物庫 . . . 37
17 / ささやきの礼拝堂 . . . 39
18 / 地図職人の野営地 . . . 41
19 / こだまの戸棚 . . . 43
20 / 九つの記憶の部屋 . . . 45
21 / 書記たちの港 . . . 48
22 / からくり果樹園 . . . 50
23 / 織り手の試練 . . . 52
24 / 始まりの頂 . . . 54

さらに学ぶ 56

旅に出る前に

旅の目標

この旅では、一度の移動からテキストオブジェクト、レジスター、マクロまで、意図を持って文章を操作する方法を学びます。各地域では小さなコマンドのまとまりを紹介します。課題では偶然目標の文章を作るだけでなく、指定された道具を使います。

ゲームの完了は、この学習課程を練習したことを意味します。完全なVimの全機能を習得したという意味ではありません。終えた課題にも戻り、明快で繰り返せる自分の解き方を探しましょう。

遊び方

言語を選び、キャラクターに名前を付け、利用規約に同意します。盤面をクリックするとキーボード入力を受け付けます。ノーマルモードでは文字がコマンドに、挿入モードでは文章になります。Escでノーマルモードに戻ります。タッチ端末では盤面の下のキーと入力欄を使えます。

動く前に目標を読み、現在の文章と目標の違い、必要なコマンドを確認しましょう。ヒントは段階的に詳しくなります。やり直しは現在の課題だけに適用され、地図から解放済みの地域へ戻れます。 ヘルプボタンから、いつでも説明とガイドを開けます。挿入モードのEscはノーマルモードに戻り、ノーマルモードで入力途中のコマンドがなければポーズを開きます。ポーズボタンも使えます。 最後の三つの課題は任意のタイムチャレンジとして遊べます。目安の時間は追加の挑戦であり、課程を終える条件ではありません。速さより先に正確さを練習しましょう。

保存とプライバシー

名前、規約への同意、進行状況は、このブラウザーのlocalStorageに保存されます。アカウントやクラウドのバックアップではありません。プライベートブラウズ、サイトデータの削除、保存容量の制限で消える場合があります。保管や別の端末への移動にはバックアップをダウンロードしてください。

任意の診断は最初は無効で、設定に独立したスイッチがあります。遊ぶための条件ではありません。限定的な技術エラーデータを扱い、名前、練習の文章、セーブ全体は含めません。詳しいデータの扱いは利用規約をご覧ください。

キー表記の読み方

大文字と小文字は別です。wとW、pとP、iとIは異なるコマンドです。3wは3、続いてwを押す意味です。Ctrl-rはControlを押しながらrを押します。+、-、|、$、%、セミコロン、コンマも実際のコマンドキーです。

ciwはオペレーターcとテキストオブジェクトiwの組み合わせです。説明の言語が違っても、石板の名前や目標の文章は表示どおりに入力します。:で始まるコマンドと、/または?の検索はEnterで確定します。 手順を区切る矢印は入力せず、順番にキーを押します。<Esc>、<Enter>、<C-r>、<Space>はEscape、Enter、Control+r、空白キーを意味し、山括弧は入力しません。

ゲームと本物の Vim

LUMENは、選んだVimの学習範囲を扱う教育用シミュレーターです。盤面は文章のバッファー、その見える部分は表示ウィンドウを表します。スクロールは文章ではなく表示を変えます。ファイルは仮想で、端末のフォルダーを読んだりOSのコマンドを実行したりしません。

コマンドは学習する編集操作に沿いますが、すべての設定、プラグイン、構文の例外を持つ完全なVimを置き換えるものではありません。ゲーム内ヘルプとこのガイドで対応範囲を説明します。

コマンドライン

コロンでコマンドラインを開きます。:wは仮想文書の保存、:eは別の文書を開く、:lsはバッファー一覧、:bは切り替えです。:marksはマーク一覧を表示し、:delmarksと省略形の:delmはマークを削除します。:delmarks!はローカルマークを消去し、:regはレジスターの内容を表示します。

:set nuと:set nonuは絶対行番号を有効・無効にし、:set nu!は切り替えます。:set rnu、:set nornu、:set rnu!は相対行番号を操作します。Enterで確定し、Escで取り消します。

レジスター

レジスターはVimのクリップボードです。通常のコピーと削除は名前なしレジスターを使い、二重引用符の接頭辞で別のものを選びます。a-zは名前付きで、A-Zへ書くと対応する小文字のレジスターに追加されます。

0は直前のコピー、1-9は大きな削除の履歴、-は小さな削除を保持します。_はブラックホールで、ほかの内容を上書きせず削除できます。:regで実際に入った内容を確認しましょう。

マクロ

マクロはカーソル移動を含む一連の操作を記録します。レジスターaへ記録するにはqaを押し、操作を行い、qで記録を終了します。@aはaを再生し、@@は最後に再生したマクロを繰り返します。前の数字で回数を指定できます。

何度も再生する前に、一回分を試しましょう。便利なマクロは、次の部分を処理できる予測可能な位置で終わります。

仮想ファイルとバッファー

バッファーは開いている文書、ファイル名は仮想の保存版を識別する名前です。:w [名前]で保存、:e 名前で開き、:lsで一覧を表示、:b 名前または番号で切り替えます。文書を離れる前に、残したい変更を保存しましょう。

すべての操作はゲームとそのローカルセーブの中で行います。旅のエクスポートはバックアップであり、端末のファイルシステムへ実際のソースファイルを保存するものではありません。パスワードなどの秘密情報を練習に入力しないでください。

<Esc> は Escape、<Enter> は Enter、<C-r> は Ctrl+r、<Space> は空白です。矢印は断片の区切りで、入力するキーではありません。大文字と小文字を区別してください。

コマンド図鑑

コマンド	動作
`0`	現在の行の最初の列へ移動します。
`hjkl`	hは左、jは下、kは上、lは右。基本のカーソル移動です。
`h`	左へ1マス、または1文字移動します。
`j`	下へ1行、または1マス移動します。
`k`	上へ1行、または1マス移動します。
`l`	右へ1マス、または1文字移動します。
`w`	次の単語の先頭へ移動します。
`b`	単語の先頭へ向かって後ろに移動します。
`e`	単語の末尾へ移動します。
`^`	現在の行で空白ではない最初の文字へ移動します。
`$`	現在の行の最後の文字へ移動します。
`gg`	最初の行へ移動します。
`G`	最後の行へ移動します。
`count`	数字は次の動きを繰り返す回数です。例：3w、2j。
`f`	fのあとに文字を押すと、その行で次に現れる同じ文字を探します。
`;`	fで行った直前の文字検索を繰り返します。
`/`	文字列を前方検索し、Enterで確定します。
`n`	元の検索方向に繰り返し検索します。
`N`	最後の/または?と反対方向に検索します。
`x`	カーソル下の文字を削除します。
`r`	カーソルの下の文字を、次に入力する文字で置き換えます。
`dd`	現在の行を切り取り、レジスターに保存します。
`dw`	カーソルから次の単語の先頭の手前まで削除します。
`u`	直前の文章の変更を元に戻します。
`Ctrl-r`	元に戻した変更をやり直します。
`yy`	現在の行全体をレジスターへコピーします。

コマンド	動作
yw	カーソルから次の単語の先頭の手前までコピーします。
p	文章はカーソルのあと、行全体は現在の行の下へ貼り付けます。
i	カーソルの前から入力を始めます。
a	カーソルのあとから入力を始めます。
A	現在の行の末尾から入力を始めます。
o	下に新しい行を開いて入力を始めます。
Esc	ノーマルモードに戻るか、入力途中のコマンドを取り消します。
:w	仮想文書を保存します。任意の名前でゲーム内のファイルを指定できます。
1 ... 9	オペレーターや移動の前の数字は回数です。2d3wは両方の回数を組み合わせます。
W	空白で区切る次のWORDの先頭へ移動します。句読点もWORDの一部です。
E	空白で区切るWORDの末尾へ移動します。
B	句読点で分けずに、前のWORDの先頭へ戻ります。
ge	前の単語の末尾へ戻ります。
gE	前のWORDの末尾へ戻ります。
+	次の行の最初の非空白文字へ移動します。回数も指定できます。
-	前の行の最初の非空白文字へ移動します。
|	5|のように指定した列へ移動します。数字がなければ1列目です。
F	Fのあとに文字を押し、現在の行を後ろ向きに探します。
t	現在の行の次の指定文字の直前まで移動します。
T	後ろ向きに文字を探し、その直後で止まります。
,	直前のf、F、t、Tを反対方向に繰り返します。
*	カーソル下の単語全体を前方へ検索します。
#	カーソル下の単語全体を後方へ検索します。
?	後方への文字列検索を開き、Enterで確定します。
%	対応する()、[]、{}の括弧へ移動します。
[{	囲んでいるブロックの対応していない開き波括弧へ戻ります。
]}	囲んでいるブロックの対応していない閉じ波括弧へ進みます。

コマンド	動作
[(	囲んでいる対応していない開き丸括弧へ戻ります。
])	囲んでいる対応していない閉じ丸括弧へ進みます。
{	空行で区切られる段落の前の境界へ戻ります。
}	次の段落の境界へ進みます。
(	文の先頭へ向かって戻ります。
)	次の文の先頭へ進みます。
zt	現在の行が表示範囲の上端になるようにします。
zz	文章を変えずに現在の行を表示の中央へ置きます。
zb	現在の行が表示範囲の下端になるようにします。
H	表示範囲で最初に見える行へ移動します。
M	表示範囲の中央の行へ移動します。
L	表示範囲で最後に見える行へ移動します。
nu / :set nu	絶対行番号を表示します。
nonu / :set nonu	絶対行番号を非表示にします。
nu! / :set nu!	絶対行番号の表示を切り替えます。
rnu / :set rnu	カーソル行からの距離を示す相対行番号を表示します。
nornu / :set nornu	相対行番号を非表示にします。
rnu! / :set rnu!	相対行番号の表示を切り替えます。
X	カーソルの前の文字を削除します。
D	カーソルから行末まで削除します。
~	カーソル下の文字の大文字・小文字を切り替えます。
I	行の最初の非空白文字から挿入を始めます。
O	現在の行の上に新しい行を開き、挿入を始めます。
s	カーソル下の文字を消して挿入を始めます。
S	現在の行全体を変更するため挿入モードに入ります。
d	削除オペレーターです。d$、diw、d2wのように移動や対象を続けます。
y	コピーのオペレーターです。移動やテキストオブジェクトと組み合わせます。
c	範囲を削除して挿入を始める変更オペレーターです。

コマンド	動作
cc	現在の行全体を変更し、挿入を始めます。
C	カーソルから行末まで変更し、挿入を始めます。
Y	現在の行全体をコピーします。このゲームではyyと同じです。
.	入力した文章を含め、直前の編集を繰り返します。
iw	周囲の空白を含めない単語内部です。d、y、cと組み合わせます。
aw	隣接する空白を含む単語の対象範囲です。
iW	句読点を含む、空白で区切るWORDの内部です。
aW	周囲の空白を含むWORDです。
is	区切りの空白を含めない文の内部です。
as	区切りの空白を含む文です。
ip	周囲の空行を含めない段落内部です。
ap	区切りの空行を含む段落です。
i[/ i]	対応する[]の内側です。括弧を除き、両方の別名は同じです。
a[/ a]	括弧を含む[]全体です。
i(/ i) / ib	対応する()の内側です。括弧を除き、ibも同じ意味です。
a(/ a) / ab	括弧を含む()全体です。abも同じ意味です。
i{ / i} / iB	対応する{}の内側です。括弧を除き、iBも同じ意味です。
a{ / a} / aB	括弧を含む{}全体です。aBも同じ意味です。
i< / i>	対応する<>の内側です。山括弧は含めません。
a< / a>	山括弧を含む<>全体です。
i"	二重引用符の内側の文章です。引用符を除きます。
a"	周囲の二重引用符を含む文章です。
i'	一重引用符の内側です。区切り記号を除きます。
a'	周囲の一重引用符を含む文章です。
i`	バッククォートの内側です。区切り記号を除きます。
a`	周囲のバッククォートを含む文章です。
it	対応するHTMLタグの内側です。タグ自体を除きます。
at	開始タグと終了タグを含む要素全体です。

コマンド	動作
`m`	mと文字で現在位置を記録します。例：ma。
`'`	一重引用符とマーク名で、その行の最初の非空白文字へ飛びます。
`` ` ``	バッククォートとマーク名で、記録した正確な位置へ飛びます。
`:marks`	記録済みのマークを表示します。
`:delmarks / :delm`	指定マークを削除します。例：:delmarks a。:delmは省略形です。
`:delmarks!`	ローカルマークをすべて削除します。
`"`	二重引用符、レジスター名、オペレーターかpの順で選びます。
`""`	名前なしレジスターは直近にコピー・削除した文章を保持します。
`"-`	小削除レジスターは一行未満の削除を保持します。
`"_`	ブラックホールです。ほかのレジスターを上書きせず削除します。
`"0`	0は、そのあと削除しても直前のコピーを保持します。
`:reg / :registers`	レジスターを表示します。:reg a0のように名前も指定できます。
`:e`	指定名の仮想ファイルを開きます。
`:ls`	開いているバッファーと番号を表示します。
`:b`	名前または番号で仮想バッファーを切り替えます。
`q`	qと文字で記録開始、もう一度qで終了します。
`@`	@aのようにレジスターのマクロを再生します。回数も指定できます。
`@@`	直前に使ったマクロをもう一度再生します。
`"1 ... "9`	大きな削除の履歴です。1が最新、9が保持する中で最も古いものです。
`"a ... "z`	名前付きa-zレジスターです。オペレーターの前に文字で選びます。
`"A ... "Z`	A-Zは対応するa-zを置き換えず、後ろへ追加します。
`P`	文章をカーソルの前、行全体なら現在の行の上へ貼り付けます。
`:set`	表示設定を変えます。例：:set nu、:set rnu!。
`:`	コマンドラインを開き、入力してEnterで確定します。
`z`	スクロールの接頭辞です。t、z、bで完成させます。
`:undo`	最後の編集を元に戻します。uのコマンドライン形式です。
`:redo`	元に戻した編集をやり直します。Ctrl-rのコマンドライン形式です。
`:delm!`	:delmarks!の省略形で、ローカルマークをすべて削除します。

島々の案内

まず自分で試してみましょう。解答は一例です。ほかの正しいキー操作でも目標を達成できる場合があります。

<Esc> は Escape、<Enter> は Enter、<C-r> は Ctrl+r、<Space> は空白です。矢印は断片の区切りで、入力するキーではありません。大文字と小文字を区別してください。

01 / 灯籠の港

すべての旅は、ひとつの動きから

波があなたを、すべての灯りが消えた港へと運びました。ネリは波止場の石の間に散らばった火花を見つけています。それを集めれば、最初の道が世界に姿を取り戻します。

課題 01 / 最初の火花

lで右へ進んで火花を集め、それからkで上に登って門に着きましょう。

開始位置: 行 4, 列 3

光を集める: (4, 6). カーソルの目的地: (2, 6).

```
1     #########
2     #.......#
3     #.......#
4     #.......#
5     #########
```

練習するコマンド: l, k

ヒント: lは右、kは上に動きます。この課題では両方のキーを使います。 火花を集めて終わりではありません。その光を輝く門まで運びましょう。

解き方の一例

`lllkk`

課題 02 / 潮の階段

hとjで上の道を離れて火花を集め、左側の門に着きましょう。

開始位置: 行 2, 列 8

光を集める: (4, 6). カーソルの目的地: (4, 3).

```
1     #########
2     #.......#
3     #.......#
4     #.......#
5     #########
```

練習するコマンド: h, j

ヒント: hは左、jは下に動きます。両方の動きが必要です。 火花に近づいたら、jで下へ進んで集めましょう。取り戻した光を門へ運びます。

解き方の一例

`hhjjhhh`

課題 03 / 波止場に灯りを

残りの火花をすべて集めましょう。光が戻れば、港は次の島を示してくれます。

開始位置: 行 2, 列 2

光を集める: (2, 8), (6, 8), (6, 2). カーソルの目的地: (4, 4).

```
#########
#.......#
#.#####.#
#...#...#
#.#####.#
#.......#
#########
```

ヒント: 動く前に、道全体と光の位置を見渡してみましょう。 行き止まりなら、数歩戻っても大丈夫です。急ぐ必要はありません。落ち着いて試してみましょう。

解き方の一例

`lllllljjjjhhhhhhkkll`

02 / 言葉の木立

一文字ずつではなく、言葉で考える

この木立の木々は言葉から育ちます。静寂の潮が根を散らしてから、イヴォは文と文の間を渡れなくなりました。単語の始まりと終わりを見つける動きを教えてもらいましょう。

課題 04 / 根から根へ

wで単語の先頭を渡り歩き、示された位置に着きましょう。

開始位置: 行 1, 列 1

カーソルの目的地: 行 1, 列 10.

```
1     dew moss lantern river
```

練習するコマンド: w

ヒント: wは次の単語の先頭へ移動します。 空白は言葉の島を隔てるすき間です。wなら、そのすき間を一度の動きで越えられます。

解き方の一例

ww

課題 05 / 背後の足跡

前に通った単語を見つけましょう。bで記録の道を戻ります。

開始位置: 行 1, 列 10

カーソルの目的地: 行 1, 列 5.

```
1     dew moss lantern river
```

練習するコマンド: b

ヒント: bは単語の先頭へ向かって後ろに移動します。 単語の途中にいる場合、最初のbはその単語の先頭へ戻ります。もう一度押すと、さらに前の単語へ進みます。

解き方の一例

b

課題 06 / 枝の先で

正しい単語の最後の文字に止まりましょう。eを試してみます。

開始位置: 行 1, 列 1

カーソルの目的地: 行 1, 列 11.

```
1     soft silver light
```

練習するコマンド: e

ヒント: eは単語の末尾へ移動します。 光る目標を見てみましょう。単語の先頭と末尾は別の場所です。ぴったり届く動きを選びます。

解き方の一例

we

03 / 行間の石橋

こちらの端から、向こうの端へ

石橋が覚えているのは、始まりと終わりだけ。セラは橋の端を見つけ、動きに回数を添える方法を教えてくれます。長い道でも、何度も別々にキーを押す必要はありません。

課題 07 / 最初の石

0で行の先頭を訪れ、それから^で最初の文字に着きましょう。

開始位置: 行 1, 列 10

カーソルの目的地: 行 1, 列 4.

```
1         stone bridge
```

練習するコマンド: 0, ^

ヒント: 0は空白があっても最初の列に進みます。^は先頭の空白を飛ばします。 0のあとに^を押しましょう。目標は最初の文字ですが、二つの始まり方を両方練習します。

解き方の一例

`0^`

課題 08 / 向こう岸

Gで最後の行に飛び、それから$でその末尾に着きましょう。

開始位置: 行 1, 列 1

カーソルの目的地: 行 3, 列 16.

```
1     a quiet river
2     under the bridge
3     the last lantern
```

練習するコマンド: G, $

ヒント: Gは記録全体の最後の行へ、$は現在の行の最後の文字へ進みます。 G、それから$と組み合わせましょう。一行ずつ下がる必要はありません。

解き方の一例

`G$`

課題 09 / 大きな一歩

ggで一番上に戻り、3jで三行下がり、0で最初の列を選びましょう。

開始位置: 行 5, 列 5

カーソルの目的地: 行 4, 列 1.

```
1     north gate
2     mist
3     old steps
4     star bridge
5     last lantern
```

練習するコマンド: gg, count, 0

ヒント: ggは最初の行へ進みます。jの前の数字3は、その動きを三回繰り返す意味です。 gg、3j、0の順に使いましょう。数字は書き込む文章ではなくコマンドの一部です。

解き方の一例

`gg3j0`

04 / こだまの天文台

ひと目では見えないものを探す

島の上では今も星が輝いていますが、その名前は石板のあちこちに散らばっています。オリンには戻ってくるこだまが聞こえるそうです。文字や言葉を探し、その中の道を読み解きましょう。

課題 10 / 選んだひとつの星

fを使い、現在の行で文字lを見つけましょう。

開始位置: 行 1, 列 1

カーソルの目的地: 行 1, 列 8.

```
1     moss | lantern | stars
```

練習するコマンド: f

ヒント: fは探す文字の入力を待ちます。この課題ではlです。 f、それからlを押します。一文字ずつ進まなくても、次のlに着きます。

解き方の一例

```
fl
```

課題 11 / 文字のこだま

fbで次のbを探し、セミコロン;でその検索を繰り返しましょう。

開始位置: 行 1, 列 1

カーソルの目的地: 行 1, 列 11.

```
1     bell bell bell
```

練習するコマンド: f, ;

ヒント: fのあとにbを押します。;は直前の一文字検索を繰り返します。 fbのあとに;を使い、同じ行のさらに次のbへ進みましょう。

解き方の一例

```
fb;
```

課題 12 / 帰ってくるこだま

/lightとEnterでlightを検索します。n、N、再びnを使い、最後の石板のこだまへ進みましょう。

開始位置: 行 1, 列 1

カーソルの目的地: 行 3, 列 10.

```
1     light sleeps
2     follow light
3     wake the light
```

練習するコマンド: /, n, N

ヒント: nは同じ方向の次の結果へ進み、Nは反対方向に検索します。 /lightを入力してEnterを押し、n、N、nと続けます。この課題では両方向を練習しましょう。

解き方の一例

```
/light → <Enter> → nNn
```

05 / 残り火の鍛冶場

思い切って直そう。変更は元に戻せる

炉には新しい文字を鍛えるだけの温もりが残っています。アシャは完璧な動きを求めません。腕のよい職人は、間違いを元に戻す方法も知っているからです。傷んだ記録を直し、火をよみがえらせましょう。

課題 13 / 記録に刺さったかけら

xで余分な文字を消し、正しい文章を取り戻しましょう。

開始位置: 行 1, 列 6

開始時のテキスト

```
1  sparkx
```

目標のテキスト

```
1  spark
```

練習するコマンド: x

ヒント: xはカーソルの下にある文字を削除します。 まず余分な文字の上へカーソルを置きます。削除したら、カーソルの位置だけでなく文章全体を確認しましょう。

解き方の一例

x

課題 14 / 金属の記憶

rgで最初のcをgに置き換え、uで元に戻し、Ctrl-rでやり直しましょう。

開始位置: 行 1, 列 1

開始時のテキスト

```
1  cold ember
2  cold ember
```

目標のテキスト

```
1  gold ember
2  cold ember
```

練習するコマンド: r, u, Ctrl-r

ヒント: rは置き換える文字を待ちます。rgでcoldがgoldになります。 rgのあとにu、Ctrl-rを使います。目標は一行目がgold ember、二行目がcold emberです。

解き方の一例

`rg → u → <C-r>`

課題 15 / 新しいかたち

dwで最初のbrokenを消し、ddで残った傷んだ行を切り取りましょう。dawnだけを残します。

開始位置: 行 1, 列 1

開始時のテキスト

```
1      broken broken
2      dawn
```

目標のテキスト

```
1      dawn
```

練習するコマンド: dw, dd

ヒント: dwは次の単語へ向かって削除し、ddは現在の行全体を削除します。 まずdw、それからdd。大きさの違う傷には、二つの道具を使い分けます。

解き方の一例

dwdd

06 / 鏡の図書館

よい考えには、もうひとつの響きを

本棚の本は、水面にページを映しています。ミラは言葉が消える前に取っておく方法を知っています。レジスターを使えば、小さな光をコピーしたり、行全体を別の場所へ運んだりできます。

課題 16 / 双子の行

yyで行をコピーし、pでその下に映し身を置きましょう。

開始位置: 行 1, 列 1

開始時のテキスト

```
1  light
2  sleep
```

目標のテキスト

```
1  light
2  light
3  sleep
```

練習するコマンド: yy, p

ヒント: yyは元の行を消さずに、現在の行全体をコピーします。 行をコピーしたあと、pは現在の行の下に貼り付けます。先にコピーを置く場所を選びましょう。

解き方の一例

```
yyp
```

課題 17 / 鏡の中の言葉

ywで単語を取っておき、pで正しい場所へ置きましょう。

開始位置: 行 1, 列 1

開始時のテキスト

```
1  glow moss
```

目標のテキスト

```
1  glow glow moss
```

練習するコマンド: yw, p

ヒント: ywはカーソルから次の単語の先頭の手前までをコピーします。 文章の一部をコピーした場合、pはカーソルのあとに貼り付けます。行全体を貼る操作とは異なります。

解き方の一例

```
ywwhp
```

課題 18 / 新しい順番

ddで行を切り取り、pで移動して正しい順番に並べ直しましょう。

開始位置: 行 1, 列 1

開始時のテキスト

```
1  mist
2  sun
3  sky
```

目標のテキスト

```
1  sun
2  mist
3  sky
```

練習するコマンド: dd, p

ヒント: ddは削除した行を、pが使う同じレジスターに保存します。 切り取ったら移動先へ進みます。pは覚えておいた行を、現在の行の下へ置きます。

解き方の一例

`ddp`

07 / 夜明けの書斎

新しい記録を残そう

エノは年代記に空白を見つけました。動くだけでは、その言葉は戻りません。誰かが書き足す必要があります。ここでは、文章を操作するときと入力するときの切り替えを覚えましょう。

課題 19 / 足りない言葉

カーソルを合わせ、iで挿入モードに入り、足りない文章を入力しましょう。

開始位置: 行 1, 列 5

開始時のテキスト

```
1  the returns
```

目標のテキスト

```
1  the dawn returns
```

練習するコマンド: i, Esc

ヒント: iはカーソルの前から入力を始めます。 挿入モードでは押した文字が文章になります。Escで移動コマンドを使えるノーマルモードへ戻れます。

解き方の一例

`i → dawn<Space> → <Esc>`

課題 20 / 書き足すための場所

aでsを書き足し、Escで戻ります。それからAで行末に空白とshineを加えましょう。

開始位置: 行 1, 列 4

開始時のテキスト

```
1  star
```

目標のテキスト

```
1  stars shine
```

練習するコマンド: a, A, Esc

ヒント: aはカーソルのあと、Aは行全体の末尾から入力します。 a、s、Escでstarをstarsにします。続けてA、空白、shine、Escでstars shineになります。

解き方の一例

`as → <Esc> → A<Space>shine → <Esc>`

課題 21 / 年代記の封印

oで新しい行を開き、returnsを入力します。Escで戻り、:wとEnterで保存しましょう。

開始位置: 行 1, 列 1

開始時のテキスト

```
1  dawn
```

目標のテキスト

```
1  dawn
2  returns
```

練習するコマンド: o, Esc, :w

ヒント: oは下に新しい行を開き、そのまま挿入モードに入ります。 dawnの下にreturnsを書きましょう。Esc、:w、Enterで完成した石板を封じます。

解き方の一例

`oreturns → <Esc> → :w → <Enter>`

08 / 朝の塔

すべての道は、光へと続く

最初の島々のかけらで作ったレンズが、頂で待っています。学んだ技術を合わせて、最初の夜明けを呼びましょう。その光が、さらなる地域と熟達への道を照らします。

課題 22 / 記憶の中の地図

ddで傷んだ最初の行を切り取り、yyで正しい行をコピーし、pで貼り付け、:wで確定しましょう。

開始位置: 行 1, 列 1

開始時のテキスト

```
1     lost lost
2     keep the light
```

目標のテキスト

```
1     keep the light
2     keep the light
```

練習するコマンド: dd, yy, p, :w

ヒント: ddのあとにはkeep the lightが残ります。yyで行全体を覚え、pで下にコピーします。 dd、yy、p、それから:wとEnterを使います。同じkeep the lightの行を二つ残しましょう。

解き方の一例

`ddyyp:w → <Enter>`

課題 23 / 最後の修復

/sleepとEnterでsleepを探し、dwで削除します。Aでsingを加え、:wで保存しましょう。

開始位置: 行 1, 列 1

開始時のテキスト

```
1     moss covers dawn
2     the bells sleep
```

目標のテキスト

```
1     moss covers dawn
2     the bells sing
```

練習するコマンド: /, dw, A, :w

ヒント: 検索するとsleepに着きます。dwでその単語を消し、Aで行を書き終えます。 一行目はそのままにします。二行目はthe bells singです。Escで入力を終え、:wとEnterで保存しましょう。

解き方の一例

`/sleep → <Enter> → dwAsing → <Esc> → :w → <Enter>`

課題 24 / 光よ、もう一度

最初の行を削除し、残った行をコピーし、新しい行にthe dawn returnsを書きましょう。直した年代記を保存します。

開始位置: 行 1, 列 1

開始時のテキスト

```
1     the night stays
2     one small spark
```

目標のテキスト

```
1     one small spark
2     one small spark
3     the dawn returns
```

練習するコマンド: dd, yy, p, o, :w

ヒント: ddで傷んだ行を消し、yyとpで二つ目のone small sparkを作ります。oで最後の行を開きます。 dd、yy、pのあと、oでthe dawn returnsを入力します。Esc、:w、Enterで最初の夜明けを呼び、次の道を開きます。

解き方の一例

`ddyypothe<Space>dawn<Space>returns → <Esc> → :w → <Enter>`

09 / 風の崖

単語よりも大きな言葉

風がちぎれた文字のリボンを運びます。タヴィは文字を数えず、空白のないひとかたまりを渡って戻る方法を教えてくれます。

課題 25 / 途切れないリボン

Wでone-two全体を越え、three.fourの先頭に着きましょう。 必要なコマンド: W.

開始位置: 行 1, 列 1

カーソルの目的地: 行 1, 列 9.

```
1     one-two three.four five
```

練習するコマンド: W

ヒント: 空白で区切る次のWORDの先頭へ移動します。句読点もWORDの一部です。 まず一度、意図した操作をしてカーソルや文章を確認します。一回の動きが分かってから繰り返しを加えましょう。

解き方の一例

```
W
```

課題 26 / 尾根を戻る

EとBをge、gEと比べ、最後に最初の列へ戻りましょう。 必要なコマンド: E · B · ge · gE.

開始位置: 行 1, 列 23

カーソルの目的地: 行 1, 列 1.

```
1     one-two three.four five
```

練習するコマンド: E, B, ge, gE

ヒント: 空白で区切るWORDの末尾へ移動します。 句読点で分けずに、前のWORDの先頭へ戻ります。

解き方の一例

```
BEgegE0
```

課題 27 / 測った歩幅

+と-と回数を練習し、列指定でthirdのtに着きましょう。 必要なコマンド: + · - · | · 回数.

開始位置: 行 1, 列 4

カーソルの目的地: 行 3, 列 2.

```
1      first
2        second
3      third
```

練習するコマンド: +, -, |, count

ヒント: 次の行の最初の非空白文字へ移動します。回数も指定できます。 前の行の最初の非空白文字へ移動します。

解き方の一例

```
+-2+2|
```

10 / 追跡者の峠

後ろの道も見てみよう

小さな刻みが峠の道を示します。リンは両方向を探し、次の安全な石が始まる場所にぴたりと止まる方法を教えます。

課題 28 / 跡を逆にたどる

aを後ろ向きに探し、tとTを比べ、コンマで反転して最後のaの直前に止まりましょう。 必要なコマンド: F・t・T・,.

開始位置: 行 1, 列 13

カーソルの目的地: 行 1, 列 12.

```
1     a x a y a z a
```

練習するコマンド: F, t, T, ,

ヒント: Fのあとに文字を押し、現在の行を後ろ向きに探します。 現在の行の次の指定文字の直前まで移動します。

解き方の一例

`FataTa,`

課題 29 / 言葉が呼び返す

*と#で単語glow全体を両方向に探し、一行目の二つ目のglowに止まりましょう。 必要なコマンド: *・#.

開始位置: 行 1, 列 1

カーソルの目的地: 行 1, 列 11.

```
1     glow moss glow
2     glow
```

練習するコマンド: *, #

ヒント: カーソル下の単語全体を前方へ検索します。 カーソル下の単語全体を後方へ検索します。

解き方の一例

`*#*`

課題 30 / 逆向きの道

?でstarを後方検索し、nとNを練習して中央の結果に止まりましょう。 必要なコマンド: ?・n・N.

開始位置: 行 3, 列 1

カーソルの目的地: 行 2, 列 1.

```
1     star
2     star
3     star
```

練習するコマンド: ?, n, N

ヒント: 後方への文字列検索を開き、Enterで確定します。 元の検索方向に繰り返し検索します。

解き方の一例

`?star → <Enter> → nN`

11 / 石の輪

開くものには閉じるものがある

ウマは輪の中に隠れた輪を守ります。括弧、文の境目、段落の息づかいを読めば、石は迷路ではなくなります。

課題 31 / 対になる石

%とブロック境界の移動で入れ子の括弧を調べ、外側の開き波括弧に戻りましょう。 必要なコマンド: % · [(·]) · [{ ·]}.

開始位置: 行 2, 列 5

カーソルの目的地: 行 1, 列 1.

```
{
  [a(b)c]
}
```

練習するコマンド: %, [(,]), [{,]}

ヒント: 対応する()、[]、{}の括弧へ移動します。 囲んでいる対応していない開き丸括弧へ戻ります。

解き方の一例

`% → [( → ]) → [{ → ]} → %`

課題 32 / 段落の間

段落の間を前後に移動し、northとsouthの間の空行に止まりましょう。 必要なコマンド: { · }.

開始位置: 行 1, 列 1

カーソルの目的地: 行 3, 列 1.

```
north
path

south
gate

dawn
```

練習するコマンド: {, }

ヒント: 空行で区切られる段落の前の境界へ戻ります。 次の段落の境界へ進みます。

解き方の一例

`}}{`

課題 33 / 呼吸のための場所

文の先頭を両方向に訪れ、Secondの先頭へ戻りましょう。 必要なコマンド: (·).

開始位置: 行 1, 列 1

カーソルの目的地: 行 1, 列 14.

```
1     First light. Second light! Third light?
```

練習するコマンド: (,)

ヒント: 文の先頭へ向かって戻ります。 次の文の先頭へ進みます。

解き方の一例

```
)) (
```

12 / 地平線の塔

文章ではなく、見方を変える

塔の窓は巻物に沿って動きます。ノアはカーソルの場所と、窓から今見える部分の違いを見せてくれます。

課題 34 / 窓が上がる

三つの窓位置を試し、最後は現在の五行目を表示の上端に置きましょう。 必要なコマンド: zt · zz · zb.

開始位置: 行 5, 列 1

カーソルの目的地: 行 5, 列 1.

```
1     line one
2     line two
3     line three
4     line four
5     line five
6     line six
7     line seven
8     line eight
9     line nine
```

最初に表示される行: 5

練習するコマンド: zt, zz, zb

ヒント: 現在の行が表示範囲の上端になるようにします。 文章を変えずに現在の行を表示の中央へ置きます。

解き方の一例

```
ztzzzbzt
```

課題 35 / 動かない地平線

五行目を中央にし、窓の上端と下端を訪れて中央へ戻りましょう。 必要なコマンド: H・M・L.

開始位置: 行 5, 列 1

カーソルの目的地: 行 5, 列 1.

```
1      line one
2      line two
3      line three
4      line four
5      line five
6      line six
7      line seven
8      line eight
9      line nine
```

最初に表示される行: 3

練習するコマンド: H, M, L

ヒント: 表示範囲で最初に見える行へ移動します。 表示範囲の中央の行へ移動します。

解き方の一例

`zzHLM`

課題 36 / そばにある数字

両方の行番号を有効・無効・切り替えで試し、絶対番号を有効、相対番号を無効で終えましょう。 必要なコマンド: nu・nonu・nu!・rnu・nornu・rnu!.

開始位置: 行 1, 列 1

カーソルの目的地: 行 1, 列 1.

```
1      north
2      south
```

絶対行番号: 有効

相対行番号: 無効

練習するコマンド: nu, nonu, nu!, rnu, nornu, rnu!

ヒント: 絶対行番号を表示します。 絶対行番号を非表示にします。

解き方の一例

`:set<Space>nonu → <Enter> → :set<Space>nu → <Enter> → :set<Space>nu! → <Enter> → :set<Space>nu! → <Enter> → :set<Space>rnu → <Enter> → :set<Space>nornu → <Enter> → :set<Space>rnu! → <Enter> → :set<Space>rnu! → <Enter>`

13 / 織り手の工房

切って、足して、縫い合わせる

アダは文字で織った布を直します。カーソルの前へ、行の上へ、ときには縫い目全体を最初から作り直します。

課題 37 / 糸の前に

カーソル前のxを消し、DAWNを小文字にしてjunkを削ります。dawnのあとの空白は残しましょう。 必要なコマンド: X · ~ · D.

開始位置: 行 1, 列 2

開始時のテキスト

```
1     xDAWN junk
```

目標のテキスト

```
1     dawn
```

練習するコマンド: X, ~, D

ヒント: カーソルの前の文字を削除します。 カーソル下の文字の大文字・小文字を切り替えます。

解き方の一例

`X4~wD`

課題 38 / 切って縫う

Iで字下げを残してglowの前にtheを加え、Oで上にheaderの行を作りましょう。 必要なコマンド: I · O.

開始位置: 行 1, 列 7

開始時のテキスト

```
1        glow
```

目標のテキスト

```
1     header
2        the glow
```

練習するコマンド: I, O

ヒント: 行の最初の非空白文字から挿入を始めます。 現在の行の上に新しい行を開き、挿入を始めます。

解き方の一例

`Ithe<Space> → <Esc> → Oheader → <Esc>`

課題 39 / 新しい重なり

sで一文字のxをsunに、Sで二行目全体をmoonに変えましょう。 必要なコマンド: s · S.

開始位置: 行 1, 列 1

開始時のテキスト

```
1  x
2  bad line
```

目標のテキスト

```
1  sun
2  moon
```

練習するコマンド: s, S

ヒント: カーソル下の文字を消して挿入を始めます。 現在の行全体を変更するため挿入モードに入ります。

解き方の一例

`ssun → <Esc> → jSmoon → <Esc>`

14 / 文法の鋳造所

動作と、その届く範囲

ケリスは何をするかと何にするかの二つから命令を鋳造します。オペレーターと移動を選び、短い表現の意味を見つけましょう。

課題 40 / 動詞と道

cの単語変更、ccの行変更、Cの行末変更を比べ、warm sunとfixedを作りましょう。 必要なコマンド: c・cc・C.

開始位置: 行 1, 列 1

開始時のテキスト

```
1     cold moss
2     broken
```

目標のテキスト

```
1     warm sun
2     fixed
```

練習するコマンド: c, cc, C

ヒント: 範囲を削除して挿入を始める変更オペレーターです。 現在の行全体を変更し、挿入を始めます。

解き方の一例

`cewarm → <Esc> → jccfixed → <Esc> → ggwCsun → <Esc>`

課題 41 / 残して貼る

yで単語をコピーしてPでカーソルの前に貼り、Yと削除も試しながらnorthを残しましょう。 必要なコマンド: y・Y・P・d.

開始位置: 行 1, 列 1

開始時のテキスト

```
1     red blue
2     north
```

目標のテキスト

```
1     red red blue
2     north
```

練習するコマンド: y, Y, P, d

ヒント: コピーのオペレーターです。移動やテキストオブジェクトと組み合わせます。 現在の行全体をコピーします。このゲームではyyと同じです。

解き方の一例

`ywwPYpdd`

課題 42 / もう一度

dwで最初のbadを削除し、残り二つは編集を繰り返してlightだけにしましょう。 必要なコマンド: dw・..

開始位置: 行 1, 列 1

開始時のテキスト

```
1     bad bad bad light
```

目標のテキスト

```
1     light
```

練習するコマンド: dw, .

ヒント: カーソルから次の単語の先頭の手前まで削除します。 入力した文章を含め、直前の編集を繰り返します。

解き方の一例

`dw..`

15 / 言葉の聖域

距離ではなく、意味を選ぶ

エリは文字数で文章を測りません。ここでは単語、文、段落が、意図を入れるための用意された器になります。

課題 43 / 単語の中心

iwとawを比べ、mossの内部、次にblueと隣の空白を削除してredを残しましょう。 必要なコマンド: iw · aw.

開始位置: 行 1, 列 6

開始時のテキスト

```
1      red moss blue
```

目標のテキスト

```
1      red
```

練習するコマンド: iw, aw

ヒント: 周囲の空白を含めない単語内部です。d、y、cと組み合わせます。 隣接する空白を含む単語の対象範囲です。

解き方の一例

`diwwdaw`

課題 44 / 大きな単語

moss.greenを一つのWORDとして扱い、iWとaWを試してredだけを残しましょう。 必要なコマンド: iW · aW.

開始位置: 行 1, 列 7

開始時のテキスト

```
1      red moss.green blue
```

目標のテキスト

```
1      red
```

練習するコマンド: iW, aW

ヒント: 句読点を含む、空白で区切るWORDの内部です。 周囲の空白を含むWORDです。

解き方の一例

`diWwdaW`

課題 45 / 文と段落

文の内側と全体、段落の内側をコピーし、文とkeepを残してalpha/betaの段落と区切りを削除しましょう。 必要なコマンド: is・as・ip・ap.

開始位置: 行 1, 列 1

開始時のテキスト

```
1     One. Two.
2     [空行]
3     alpha
4     beta
5     [空行]
6     keep
```

目標のテキスト

```
1     One. Two.
2     [空行]
3     keep
```

練習するコマンド: is, as, ip, ap

ヒント: 区切りの空白を含めない文の内部です。 区切りの空白を含む文です。

解き方の一例

```
yisyasjjyipdap
```

16 / 区切りの宝物庫

内側と、それを囲む枠

センは箱の中の箱を守ります。中身だけか枠全体かを選びましょう。名前の違う鍵で同じ錠が開くこともあります。

課題 46 / 四角と丸

示した全別名で[]と()の内部と枠をコピーし、最後は名前なしレジスターに(glow)全体を残しましょう。 必要なコマンド: i[· a[· i] · a] · i(· a(· i) · a) · ib · ab.

開始位置: 行 1, 列 3

カーソルの目的地: 行 1, 列 8.

```
1     [moss] (glow)
```

レジスター ""

```
1     (glow)
```

練習するコマンド: i[, a[, i], a], i(, a(, i), a), ib, ab

ヒント: 対応する[]の内側です。括弧を除き、両方の別名は同じです。 括弧を含む[]全体です。

解き方の一例

```
yi[ya[yi]ya]f(yi(ya(yi)ya)yibyab
```

課題 47 / 角と別名

{}と<>の全別名を比べ、最後に枠を含む<dawn>全体をレジスターに残しましょう。 必要なコマンド: i{ · a{ · i} · a} · iB · aB · i< · a< · i> · a>.

開始位置: 行 1, 列 3

カーソルの目的地: 行 1, 列 9.

```
1     {ember} <dawn>
```

レジスター ""

```
1     <dawn>
```

練習するコマンド: i{, a{, i}, a}, iB, aB, i<, a<, i>, a>

ヒント: 対応する{}の内側です。括弧を除き、iBも同じ意味です。 括弧を含む{}全体です。aBも同じ意味です。

解き方の一例

```
yi{ya{yi}ya}yiByaBf<yi<ya<yi>ya>
```

課題 48 / 枠のすべて

(seed)の内側だけをbloomに変えて括弧を残し、[noise]全体を削除しましょう。 必要なコマンド: i(· a[.

開始位置: 行 1, 列 11

開始時のテキスト

```
1     [noise] (seed) {ash} <sky>
```

目標のテキスト

```
1      (bloom) {ash} <sky>
```

練習するコマンド: i(, a[

ヒント: 対応する()の内側です。括弧を除き、ibも同じ意味です。 括弧を含む[]全体です。

解き方の一例

`ci(bloom → <Esc> → 0da[`

17 / ささやきの礼拝堂

記号の間に耳を澄ませる

ユナは引用符やタグの中に伝言を残します。内容を聞くには、声とそれを運ぶ枠を区別することが必要です。

課題 49 / 二つの小さな声

二重・一重引用符の内側と全体を比べ、最後は一重引用符を含むglow全体をコピーしましょう。 必要なコマンド: i" · a" · i' · a'.

開始位置: 行 1, 列 3

カーソルの目的地: 行 1, 列 8.

```
1     "moss" 'glow'
```

レジスター ""

```
1     'glow'
```

練習するコマンド: i", a", i', a'

ヒント: 二重引用符の内側の文章です。引用符を除きます。 周囲の二重引用符を含む文章です。

解き方の一例

```
yi"ya"f'yi'ya'
```

課題 50 / ささやきとタグ

バッククォートの内側と全体、次にHTML要素をコピーし、glow全体を残しましょう。 必要なコマンド: i` · a` · it · at.

開始位置: 行 1, 列 3

カーソルの目的地: 行 1, 列 8.

```
1     `seed` <b>glow</b>
```

レジスター ""

```
1     <b>glow</b>
```

練習するコマンド: i`, a`, it, at

ヒント: バッククォートの内側です。区切り記号を除きます。 周囲のバッククォートを含む文章です。

解き方の一例

```
yi`ya`f<yityat
```

課題 51 / 入れ子の伝言

pタグを残して内容だけnewに変え、引用符を含むnoise全体を削除します。末尾の空白は残しましょう。 必要なコマンド: it · a".

開始位置: 行 1, 列 5

開始時のテキスト

```
1     <p>old</p> "noise"
```

目標のテキスト

```
1     <p>new</p>
```

練習するコマンド: it, a"

ヒント: 対応するHTMLタグの内側です。タグ自体を除きます。 周囲の二重引用符を含む文章です。

解き方の一例

`citnew → <Esc> → f"da"`

18 / 地図職人の野営地

戻れる場所に印を残そう

アロは風にも散らない地図を描きます。位置を覚え、行や正確な列へ戻り、役目を終えた目印を消しましょう。

課題 52 / 目印を置く

二行目にマークaを置き、行へのジャンプと正確な列へのジャンプを比べて記録位置へ戻りましょう。 必要なコマンド: m · ' · `.

開始位置: 行 2, 列 5

カーソルの目的地: 行 2, 列 5.

```
1       north
2       south
3       dawn
```

マーク a: 行 2, 列 5

練習するコマンド: m, ', `

ヒント: mと文字で現在位置を記録します。例：ma。 一重引用符とマーク名で、その行の最初の非空白文字へ飛びます。

解き方の一例

```
magg'agg`a
```

課題 53 / 地図を読む

隣の列にaとbを置き、マーク一覧を開き、:delmでaだけを削除しましょう。 必要なコマンド: :marks · :delm.

開始位置: 行 1, 列 1

カーソルの目的地: 行 1, 列 2.

```
1     path
```

マーク b: 行 1, 列 2

削除したマーク: a

開いている一覧: マーク

練習するコマンド: :marks, :delm

ヒント: 記録済みのマークを表示します。 指定マークを削除します。例：:delmarks a。:delmは省略形です。

解き方の一例

```
malmb → :marks → <Enter> → :delm<Space>a → <Enter>
```

課題 54 / 古い道を消す

マーク範囲の削除、:delmarks!、:delm!を練習し、a、b、cをすべて削除しましょう。 必要なコマンド: : delmarks・:delmarks!・:delm!.

開始位置: 行 1, 列 1

カーソルの目的地: 行 1, 列 2.

```
1     path
```

削除したマーク: a, b, c

練習するコマンド: :delmarks, :delmarks!, :delm!

ヒント: 指定マークを削除します。例：:delmarks a。:delmは省略形です。 ローカルマークをすべて削除します。

解き方の一例

`malmb → :delmarks<Space>a-b → <Enter> → ma → :delmarks! → <Enter> → mc → :delm! → <Enter>`

19 / こだまの戸棚

クリップボードにも名前を

エスメはこだまを別々の引き出しに保管します。ひとつを残し、別のひとつに足し、大切な音を消さずに雑音を手放しましょう。

課題 55 / 名前のないこだま

aを削除して小削除レジスターから戻し、名前なしレジスターを明示してab全体をコピーしましょう。 必要なコマンド: "- ・ "" ・ x ・ yy ・ P.

開始位置: 行 1, 列 1

目標のテキスト

```
1  ab
```

レジスター ""

```
1  ab
```

レジスター "-

```
1  a
```

練習するコマンド: "-, "", x, yy, P

ヒント: 小削除レジスターは一行未満の削除を保持します。 名前なしレジスターは直近にコピー・削除した文章を保持します。

解き方の一例

```
x"-P""yy
```

課題 56 / 上書きせずに追加

aにmossを保存し、Aでglowを追加して両方の行を貼り、aの内容を確かめましょう。 必要なコマンド: "a ・ "A ・ p.

開始位置: 行 1, 列 1

開始時のテキスト

```
1  moss
2  glow
```

目標のテキスト

```
1  moss
2  moss
3  glow
4  glow
```

レジスター "a

```
1  moss
2  glow
```

練習するコマンド: "a, "A, p

ヒント: 名前付きa-zレジスターです。オペレーターの前に文字で選びます。 A-Zは対応するa-zを置き換えず、後ろへ追加します。

解き方の一例

```
"ayyj"Ayygg"ap
```

課題 57 / 静けさがコピーを守る

aにkeepを残し、trashをブラックホールへ削除して、保存したコピーを現在の行の上に置きましょう。 必要なコマンド: "_ · "a · dd · P.

開始位置: 行 1, 列 1

開始時のテキスト

```
1  keep
2  trash
```

目標のテキスト

```
1  keep
2  keep
```

レジスター "a

```
1  keep
```

レジスター ""

```
1  keep
```

練習するコマンド: "_, "a, dd, P

ヒント: ブラックホールです。ほかのレジスターを上書きせず削除します。 名前付きa-zレジスターです。オペレーターの前に文字で選びます。

解き方の一例

`"ayyj"_dd"aP`

20 / 九つの記憶の部屋

記憶には順番がある

ノナが覚えているのは最後に消えたものだけではありません。履歴の引き出しを開けば、削除が必ずしも別れではないと分かります。

課題 58 / 最後のコピー

lightをコピーしてdustを削除し、0からコピーを取り戻します。1にはdustを残しましょう。 必要なコマンド: "0・dd.

開始位置: 行 1, 列 1

開始時のテキスト

```
1     light
2     dust
```

目標のテキスト

```
1     light
2     light
```

レジスター "0

```
1     light
```

レジスター "1

```
1     dust
```

練習するコマンド: "0, dd

ヒント: 0は、そのあと削除しても直前のコピーを保持します。 現在の行を切り取り、レジスターに保存します。

解き方の一例

```
yyjdd"0p
```

課題 59 / 続いていく別れ

九行を削除して履歴1から9を調べ、最も古いoneをkeepの上に戻しましょう。 必要なコマンド: "1 · "2 · "3 · "4 · "5 · "6 · "7 · "8 · "9.

開始位置: 行 1, 列 1

開始時のテキスト

```
1	one
2	two
3	three
4	four
5	five
6	six
7	seven
8	eight
9	nine
10	keep
```

目標のテキスト

```
1	one
2	keep
```

レジスター "1

```
1	nine
```

レジスター "2

```
1	eight
```

レジスター "3

```
1	seven
```

レジスター "4

```
1	six
```

レジスター "5

```
1	five
```

レジスター "6

```
1	four
```

レジスター "7

```
1	three
```

レジスター "8

```
1	two
```

レジスター "9

```
1	one
```

練習するコマンド: "1, "2, "3, "4, "5, "6, "7, "8, "9

ヒント: 大きな削除の履歴です。1が最新、9が保持する中で最も古いものです。 大きな削除の履歴です。1が最新、9が保持する中で最も古いものです。

解き方の一例

`dd........ → "1pu → "2pu → "3pu → "4pu → "5pu → "6pu → "7pu → "8pu → "9P`

課題 60 / 引き出しを読む

oneをaにコピーし、レジスター一覧を開いてaと0に同じ行があることを確かめましょう。 必要なコマンド: : reg.

開始位置: 行 1, 列 1

カーソルの目的地: 行 1, 列 1.

```
1       one
2       two
```

レジスター "0

```
1       one
```

レジスター "a

```
1       one
```

開いている一覧: レジスター

練習するコマンド: :reg

ヒント: レジスターを表示します。:reg a0のように名前も指定できます。 まず一度、意図した操作をしてカーソルや文章を確認します。一回の動きが分かってから繰り返しを加えましょう。

解き方の一例

`"ayy → :reg → <Enter>`

21 / 書記たちの港

多くのページ、ひとつの旅

テロは仮想の港の間で文書を運びます。残したいものを保存し、名前を見失わずに開いたページを切り替えましょう。

課題 61 / 覚えたページ

notes.txtにdawn risesを保存し、echo.txtをawakeに直して保存し、変更を失わずnotes.txtに戻りましょう。 必要なコマンド: :w・:e・c.

開始位置: 行 1, 列 1

開始時のテキスト

```
1      dawn
```

目標のテキスト

```
1      dawn rises
```

現在のバッファー: notes.txt

保存済み仮想ファイル: notes.txt

```
1      dawn rises
```

保存済み仮想ファイル: echo.txt

```
1      awake
```

練習するコマンド: :w, :e, c

ヒント: 仮想文書を保存します。任意の名前でゲーム内のファイルを指定できます。 指定名の仮想ファイルを開きます。

解き方の一例

```
A<Space>rises → <Esc> → :w → <Enter> → :e<Space>echo.txt → <Enter> → ccawake → <Esc> → :w → <Enter> → :e<Space>notes.txt → <Enter>
```

課題 62 / 別の港

beta.txtを開き、バッファー一覧を表示し、名前でalpha.txtへ戻りましょう。 必要なコマンド: :e・:ls・:b.

開始位置: 行 1, 列 1

カーソルの目的地: 行 1, 列 1.

```
1      north
```

現在のバッファー: alpha.txt

開いている一覧: バッファー

練習するコマンド: :e, :ls, :b

ヒント: 指定名の仮想ファイルを開きます。 開いているバッファーと番号を表示します。

解き方の一例

```
:e<Space>beta.txt → <Enter> → :ls → <Enter> → :b<Space>alpha.txt → <Enter>
```

課題 63 / バッファーを渡る

final.txtとしてコピーを保存して開き、番号での切り替えを比べ、final.txtで終えましょう。 必要なコマンド: : w・:e・:b.

開始位置: 行 1, 列 1

目標のテキスト

```
1     light
```

現在のバッファー: final.txt

保存済み仮想ファイル: final.txt

```
1     light
```

練習するコマンド: :w, :e, :b

ヒント: 仮想文書を保存します。任意の名前でゲーム内のファイルを指定できます。 指定名の仮想ファイルを開きます。

解き方の一例

`:w<Space>final.txt → <Enter> → :e<Space>final.txt → <Enter> → :b<Space>1 → <Enter> → :b<Space>2 → <Enter>`

22 / からくり果樹園

よい一手に、多くの実りを

オミは果樹園の繊細な機構を世話しています。確実な一巡を記録し、止まる位置を確かめ、次の枝にも働いてもらいましょう。

課題 64 / 小さな機械

aに!の追加と下移動を記録し、一度再生して最初の二つのdawnを変えましょう。 必要なコマンド: q · @.

開始位置: 行 1, 列 1

開始時のテキスト

```
1	dawn
2	dawn
3	dawn
```

目標のテキスト

```
1	dawn!
2	dawn!
3	dawn
```

記録したマクロ: a

練習するコマンド: q, @

ヒント: qと文字で記録開始、もう一度qで終了します。 @aのようにレジスターのマクロを再生します。回数も指定できます。

解き方の一例

`qaA! → <Esc> → jq@a`

課題 65 / 果樹園は繰り返す

行頭への+追加と下移動を記録し、@aと@@を使い、四つ目のseedはそのままにしましょう。 必要なコマンド: q・@・@@.

開始位置: 行 1, 列 1

開始時のテキスト

```
1	seed
2	seed
3	seed
4	seed
```

目標のテキスト

```
1	+seed
2	+seed
3	+seed
4	seed
```

記録したマクロ: a

練習するコマンド: q, @, @@

ヒント: qと文字で記録開始、もう一度qで終了します。 @aのようにレジスターのマクロを再生します。回数も指定できます。

解き方の一例

`qaI+ → <Esc> → jq@a@@`

課題 66 / もう一度、正確に

xを一つ消す操作を記録して三回再生し、lightだけを残しましょう。 必要なコマンド: q・@・回数.

開始位置: 行 1, 列 1

開始時のテキスト

```
1	xxxxlight
```

目標のテキスト

```
1	light
```

記録したマクロ: a

練習するコマンド: q, @, count

ヒント: qと文字で記録開始、もう一度qで終了します。 @aのようにレジスターのマクロを再生します。回数も指定できます。

解き方の一例

`qaxq3@a`

23 / 織り手の試練

覚えた模様を組み合わせる

ヴェルが長い模様を広げます。回数、編集の繰り返し、マクロは別の道具です。意図した作業を表すものを選びましょう。

課題 67 / 二つの回数

オペレーターと移動の回数を組み合わせ、2d3wで六単語を消し、keepを残しましょう。 必要なコマンド: 回数・dw.

開始位置: 行 1, 列 1

開始時のテキスト

```
1  one two three four five six keep
```

目標のテキスト

```
1  keep
```

練習するコマンド: count, dw

ヒント: オペレーターや移動の前の数字は回数です。2d3wは両方の回数を組み合わせます。 カーソルから次の単語の先頭の手前まで削除します。

解き方の一例

`2d3w`

課題 68 / 同じ織り目

xを消し、:undoと:redoで元に戻してやり直し、点で削除を繰り返しましょう。 必要なコマンド: :undo・:redo・..

開始位置: 行 1, 列 1

開始時のテキスト

```
1  xxdawn
```

目標のテキスト

```
1  dawn
```

練習するコマンド: :undo, :redo, .

ヒント: 最後の編集を元に戻します。uのコマンドライン形式です。 元に戻した編集をやり直します。Ctrl-rのコマンドライン形式です。

解き方の一例

`x → :undo → <Enter> → :redo → <Enter> → .`

課題 69 / 確かな模様

ashの削除と下移動を記録して二回繰り返し、keepで0を試して仮のコピーを元に戻しましょう。 必要なコマンド: q・@・回数・"0.

開始位置: 行 1, 列 1

開始時のテキスト

```
1      ash dawn
2      ash dawn
3      ash dawn
4      keep
```

目標のテキスト

```
1      dawn
2      dawn
3      dawn
4      keep
```

レジスター "0

```
1      keep
```

記録したマクロ: a

練習するコマンド: q, @, count, "0

ヒント: qと文字で記録開始、もう一度qで終了します。 @aのようにレジスターのマクロを再生します。回数も指定できます。

解き方の一例

```
qadwjq2@ayy"0Pu
```

24 / 始まりの頂

ひとつの記録に、群島のすべてを

すべての道が出会う場所でルマが待っています。新しい力ではなく、記憶、正確さ、想像を合わせる最後の仕事を託します。

課題 70 / 雑音のない記録

タグの内容をlightにし、[noise]全体とその空行を削除して保存しましょう。 必要なコマンド: it · a[· dd · :w.

開始位置: 行 1, 列 5

開始時のテキスト

```
<p>old</p>
[noise]
keep
```

目標のテキスト

```
<p>light</p>
keep
```

練習するコマンド: it, a[, dd, :w

ヒント: 対応するHTMLタグの内側です。タグ自体を除きます。 括弧を含む[]全体です。

解き方の一例

`citlight → <Esc> → jda[dd → :w → <Enter>`

課題 71 / 正確な帰り道

AとaでⅡつの位置を記録し、正確なジャンプと両方向の検索を合わせてdawnに戻りましょう。 必要なコマンド: m · ` · / · ? · n.

開始位置: 行 1, 列 6

カーソルの目的地: 行 3, 列 6.

```
star north
star south
star dawn
```

マーク A: 行 1, 列 6

マーク a: 行 3, 列 6

練習するコマンド: m, `, /, ?, n

ヒント: mと文字で現在位置を記録します。例：ma。 バッククォートとマーク名で、記録した正確な位置へ飛びます。

解き方の一例

`mA/dawn → <Enter> → ma`A?star → <Enter> → n`a`

課題 72 / すべての光、ひとつの空

sleepをwakeにする操作を三行に適用し、message.txtの最後の行をdawn.txtへコピーして保存しましょう。
必要なコマンド: q・@・@@・:w・:e・yy・"0・p.

開始位置: 行 1, 列 1

開始時のテキスト

```
1  sleep
2  sleep
3  sleep
```

目標のテキスト

```
1  wake
2  wake
3  wake
4  the dawn returns
```

レジスター "0

```
1  the dawn returns
```

現在のバッファー: dawn.txt

保存済み仮想ファイル: dawn.txt

```
1  wake
2  wake
3  wake
4  the dawn returns
```

保存済み仮想ファイル: message.txt

```
1  the dawn returns
```

記録したマクロ: a

練習するコマンド: q, @, @@, :w, :e, yy, "0, p

ヒント: qと文字で記録開始、もう一度qで終了します。 @aのようにレジスターのマクロを再生します。回数も指定できます。

解き方の一例

```
qaccwake → <Esc> → jq@a@@ → :w → <Enter> → :e<Space>message.txt → <Enter> → yy → :e<Space>dawn.txt → <Enter> → "0p → :w → <Enter>
```

さらに学ぶ

本書はゲームが対応する教育用の一部機能を説明します。仮想ファイルはブラウザー内に保存され、端末のファイルシステムにはアクセスしません。

Vim documentation

Motions and text objects
https://vimhelp.org/motion.txt.html

Editing and registers
https://vimhelp.org/change.txt.html

Macros
https://vimhelp.org/usr_10.txt.html

Files and buffers
https://vimhelp.org/usr_22.txt.html