# 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