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