

Basics & Navigation

Shell Navigation

|                    |   |
|--------------------|---|
| <code>cd</code>    | Change directory to <code>directory_name</code> .               |
| <code>cd ..</code> | Move one directory up.  |
| <code>cd -</code>  | Return to the previous directory.                               |
| <code>pwd</code>   | Print the current working directory.                            |
| <code>ls</code>    | List files and directories in the current directory.            |
| <code>ls -l</code> | List files with detailed information (permissions, size, etc.). |
| <code>ls -a</code> | List all files, including hidden files.                         |

Basic Commands

|                      |   |
|----------------------|---|
| <code>echo</code>    | Prints the specified text to the terminal.                          |
| <code>printf</code>  | Prints the specified text with a trailing newline.                  |
| <code>read</code>    | Reads input from stdin and assigns it to <code>my_variable</code> . |
| <code>clear</code>   | Clears the terminal screen.   |
| <code>history</code> | Displays the command history.                                       |

Keybindings

|                             |  |
|-----------------------------|--|
| <code>Ctrl+A</code>         | Move cursor to the beginning of the line.        |
| <code>Ctrl+E</code>         | Move cursor to the end of the line.              |
| <code>Alt+Left</code>       | Jump to the previous word.                       |
| <code>Alt+Right</code>      | Jump to the next word.                           |
| <code>Up/Down Arrows</code> | Switch to the previous/next command in history.  |
| <code>Alt+Up/Down</code>    | Switch to the previous/next arguments.           |
| <code>Ctrl+U</code>         | Delete from cursor to the beginning of the line. |
| <code>Ctrl+C</code>         | Cancel the current command.                      |

Variables & Loops

Variable Management

|                          |  |
|--------------------------|--|
| <code>set</code>         | Sets a variable named <code>my_variable</code> to 'value'.       |
| <code>set -g</code>      | Sets a global variable accessible in all scopes.                 |
| <code>set -l</code>      | Sets a local variable, only accessible within the current scope. |
| <code>set --erase</code> | Removes the variable <code>my_variable</code> .                  |
| <code>echo</code>        | Prints the value of <code>\$my_variable</code> .                 |
| <code>echo</code>        | Prints a slice of the variable (characters 1 to 3).              |

Loop Structures

|   |  |
|---|--|
| <code>for i in item1 item2 item3; ...; end</code> | Iterates over a list of items.                             |
| <code>for i in (seq 1 5); ...; end</code>         | Iterates over a sequence of numbers from 1 to 5.           |
| <code>while</code>                                | Executes a block of code as long as the condition is true. |
| <code>break</code>                                | Exits the current loop.                                    |
| <code>continue</code>                             | Skips the current iteration and continues with the next.   |

Arithmetic Operations

|                          |  |
|--------------------------|--|
| <code>math 1 + 2</code>  | Performs addition.                               |
| <code>math 5 - 3</code>  | Performs subtraction.                            |
| <code>math 4 \* 6</code> | Performs multiplication.                         |
| <code>math 8 / 2</code>  | Performs division.                               |
| <code>math 7 % 3</code>  | Calculates the modulo (remainder).               |
| <code>math 2 ^ 3</code>  | Performs exponentiation.                         |
| <code>set counter</code> | Increments a variable <code>counter</code> by 1. |

Conditionals & Strings

Conditional Statements

|  |  |
|--|--|
| <code>if</code><br><code>condition;</code><br><code>...; end</code>      | Executes a block of code if the condition is true.                           |
| <code>else</code>  | Executes a block of code if the previous <code>if</code> condition is false. |
| <code>else if</code><br><code>condition;</code><br><code>...; end</code> | Chains another condition to check if the initial <code>if</code> is false.   |
| <code>test</code><br><code>condition</code>                              | Evaluates a condition; returns 0 if true, 1 if false.                        |
| <code>and</code>   | Logical AND operator for combining conditions.                               |
| <code>or</code>  | Logical OR operator for combining conditions.                                |
| <code>not</code>   | Logical NOT operator to negate a condition.                                  |

String Manipulation

|  |  |
|--|--|
| <code>string length</code><br><code>'text'</code>  | Returns the length of the string 'text'.                                     |
| <code>string sub --</code><br><code>start 2 --</code><br><code>length 3</code><br><code>'example'</code> | Extracts a substring of length 3 starting from index 2.                      |
| <code>string match</code><br><code>--regex</code><br><code>'pattern'</code><br><code>'string'</code>     | Matches a string against a regular expression; returns the match or nothing. |
| <code>string</code><br><code>replace --all</code><br><code>'old' 'new'</code><br><code>'string'</code>   | Replaces all occurrences of 'old' with 'new' in 'string'.                    |
| <code>string join</code><br><code>'separator'</code><br><code>item1 item2</code><br><code>...</code>     | Joins items together with the specified separator.                           |
| <code>string split</code><br><code>'separator'</code><br><code>'string'</code>                           | Splits a string into an array based on the separator.                        |

String Matching Patterns

|                                       |   |
|---------------------------------------|---|
| <code>x?</code>                       | Matches zero or one occurrences of <code>x</code> .                               |
| <code>x*</code>                       | Matches zero or more occurrences of <code>x</code> .                              |
| <code>x+</code>                       | Matches one or more occurrences of <code>x</code> .                               |
| <code>x{n}</code>                     | Matches exactly <code>n</code> occurrences of <code>x</code> .                    |
| <code>x{n,m}</code><br><code>}</code> | Matches between <code>n</code> and <code>m</code> occurrences of <code>x</code> . |
| <code>[xy]</code>                     | Matches either <code>x</code> or <code>y</code> .                                 |
| <code>[^xy]</code><br><code>]</code>  | Matches any character that is not <code>x</code> or <code>y</code> .              |

Functions & Events

Function Definition

|  |   |
|--|---|
| <code>function</code><br><code>my_function; ...;</code><br><code>end</code>  | Defines a new function named <code>my_function</code> . |
| <code>function</code><br><code>my_function -a</code><br><code>argument1</code><br><code>argument2; ...;</code><br><code>end</code> | Defines a function accepting specific arguments.        |
| <code>function</code><br><code>my_function -d</code><br><code>'description'; ...;</code><br><code>end</code>                       | Adds a description to the function.                     |
| <code>functions --erase</code><br><code>my_function</code>   | Removes the function <code>my_function</code> .         |
| <code>return</code>  | Exits the function, optionally returning a value.       |
| <code>\$argv</code>  | Array containing the arguments passed to the function.  |

Event Handling

|  |   |
|--|---|
| <code>emit my_event</code>   | Emits an event named <code>my_event</code> .  |
| <code>function</code><br><code>hook_function --</code><br><code>on-event</code><br><code>my_event; ...;</code><br><code>end</code>       | Defines a hook function that runs when <code>my_event</code> is emitted.                          |
| <code>function</code><br><code>hook_function --</code><br><code>on-variable</code><br><code>my_variable;</code><br><code>...; end</code> | Defines a hook function that runs when <code>my_variable</code> is changed.                       |
| <code>function</code><br><code>hook_function --</code><br><code>on-signal</code><br><code>SIGINT; ...;</code><br><code>end</code>        | Defines a hook function that runs when the <code>SIGINT</code> signal is received (e.g., Ctrl+C). |

Process Communication

|  |  |
|--|--|
| <code>&gt; file</code>                     | Redirects the output to a file, overwriting its contents.                      |
| <code>&gt;&gt;</code><br><code>file</code> | Appends the output to a file.  |
| <code> </code><br><code>command</code>     | Pipes the output to another command.   |
| <code>(command)</code>                     | Command substitution; replaces the command with its output.                    |
| <code>(command   psub)</code>              | Process substitution; creates a temporary file with the output of the command. |

# Abbreviations & Completions

## Abbreviations

|   |   |
|---|---|
| <code>abbr --add</code><br><code>short_cmd</code><br>'long<br>command'                      | Adds an abbreviation; typing <code>short_cmd</code> will execute <code>long command</code> .            |
| <code>abbr --erase</code><br><code>short_cmd</code>   | Removes the abbreviation <code>short_cmd</code> .   |
| <code>abbr</code>   | Lists all defined abbreviations.  |
| Example:<br><br><code>abbr --add</code><br><code>gcm 'git</code><br><code>commit -m'</code> | Creates an abbreviation for <code>git commit -m</code> . Use: <code>gcm</code><br>'Your commit message' |

## Command Completions

|   |   |
|---|---|
| <code>complete --</code><br><code>command cmd --</code><br><code>arguments arg1</code><br><code>arg2</code> | Adds completions for the command <code>cmd</code> with specified arguments. |
| <code>complete --</code><br><code>command cmd --</code><br><code>erase</code>                               | Removes all completions for the command <code>cmd</code> .                  |
| <code>complete --</code><br><code>command cmd --</code><br><code>no-files</code>                            | Disables file completion for command <code>cmd</code> .                     |
| <code>complete --</code><br><code>command cmd --</code><br><code>force-files</code>                         | Forces file completion for command <code>cmd</code> .                       |
| <code>complete --</code><br><code>command cmd --</code><br><code>condition</code><br><code>condition</code> | Adds completions based on a specified condition.                            |
| <code>complete --</code><br><code>command cmd --</code><br><code>description</code><br>'text'               | Adds a description for the completion.                                      |

## Useful Built-in Functions

|  |   |
|--|---|
| <code>__fish_seen_argument</code>        | Checks whether a specific argument has been used. |
| <code>__fish_seen_subcommand_from</code> | Checks if a specific subcommand has been used.    |
| <code>__fish_use_subcommand</code>       | Checks if any subcommand is used.                 |
| <code>__fish_complete_directories</code> | Completes directories.                            |
| <code>__fish_complete_suffix</code>      | Completes files with a specific suffix.           |
| <code>__fish_complete_users</code>       | Lists all users for completion.                   |