

NAME

fish – the friendly interactive shell

SYNOPSIS

fish [*OPTIONS*] [*FILE* [*ARG* ...]]

fish [*OPTIONS*] [-c *COMMAND* [*ARG* ...]]

DESCRIPTION

fish is a command-line shell written mainly with interactive use in mind. This page briefly describes the options for invoking **fish**. The full manual <#intro> is available in HTML by using the **help** command from inside fish, and in the *fish-doc(1)* man page. The tutorial <> is available as HTML via **help tutorial** or in *man fish-tutorial*.

The following options are available:

-c or **--command=COMMAND**

Evaluate the specified commands instead of reading from the commandline. Any additional positional arguments are used as **\$argv**.

-C or **--init-command=COMMANDS**

Evaluate specified commands after reading the configuration but before executing command specified by **-c** or reading interactive input.

-d or **--debug=DEBUG_CATEGORIES**

Enables debug output and specify a pattern for matching debug categories. See Debugging below for details.

-o or **--debug-output=DEBUG_FILE**

Specifies a file path to receive the debug output, including categories and **fish_trace** <#envvar-fish_trace>. The default is standard error.

-i or **--interactive**

The shell is interactive.

-l or **--login**

Act as if invoked as a login shell.

-N or **--no-config**

Do not read configuration files.

-n or **--no-execute**

Do not execute any commands, only perform syntax checking.

-p or **--profile=PROFILE_FILE**

when **fish** exits, output timing information on all executed commands to the specified file. This excludes time spent starting up and reading the configuration.

--profile-startup=PROFILE_FILE

Will write timing for **fish** startup to specified file.

-P or **--private**

Enables private mode <#private-mode>: **fish** will not access old or store new history.

--print-rusage-self

When **fish** exits, output stats from getrusage.

--print-debug-categories

Print all debug categories, and then exit.

-v or **--version**

Print version and exit.

-f or --features=FEATURES

Enables one or more comma-separated feature flags <#featureflags>.

The **fish** exit status is generally the exit status of the last foreground command <#variables-status>.

DEBUGGING

While fish provides extensive support for debugging fish scripts <#debugging>, it is also possible to debug and instrument its internals. Debugging can be enabled by passing the **--debug** option. For example, the following command turns on debugging for background IO thread events, in addition to the default categories, i.e. *debug*, *error*, *warning*, and *warning-path*:

```
> fish --debug=iiothread
```

Available categories are listed by **fish --print-debug-categories**. The **--debug** option accepts a comma-separated list of categories, and supports glob syntax. The following command turns on debugging for *complete*, *history*, *history-file*, and *profile-history*, as well as the default categories:

```
> fish --debug='complete, *history*'
```

Debug messages output to stderr by default. Note that if **fish_trace** <#envvar-fish_trace> is set, execution tracing also outputs to stderr by default. You can output to a file using the **--debug-output** option:

```
> fish --debug='complete, *history*' --debug-output=/tmp/fish.log --init-command
```

These options can also be changed via the **FISH_DEBUG** <#envvar-FISH_DEBUG> and **FISH_DEBUG_OUTPUT** <#envvar-FISH_DEBUG_OUTPUT> variables. The categories enabled via **--debug** are *added* to the ones enabled by \$FISH_DEBUG, so they can be disabled by prefixing them with **-** (**reader-***, **-ast*** enables reader debugging and disables ast debugging).

The file given in **--debug-output** takes precedence over the file in **FISH_DEBUG_OUTPUT** <#envvar-FISH_DEBUG_OUTPUT>.

EXAMPLES

To just start fish:

```
fish
```

To run a file with fish:

```
fish /path/to/script.fish
```

To run some commands with fish:

```
fish -c 'echo Hi there!'
```

You can also pass arguments to those commands:

```
> fish -c 'printf %s\n $argv' "first line" "second line"
first line
second line
```

To run a script, except read this other file first:

```
fish --init-cmd "source otherfile" script.fish
```

To profile <#profiling> fish's startup and find what takes the most time in your configuration:

```
fish --profile-startup /tmp/start.prof -ic exit  
sort -nk2 /tmp/start.prof
```

Copyright

fish-shell developers