wd

build passing

`wd` (*warp directory*) lets you jump to custom directories in zsh, without using `cd`. Why? Because `cd` seems inefficient when the folder is frequently visited or has a long path.

Setup

[oh-my-zsh]

`wd` comes bundled with oh-my-zsh!

Just add the plugin in your `.zshrc` file:

```
plugins=(... wd)
```

[Antigen]

In your `.zshrc`:

```
antigen bundle mfaerevaag/wd
```

[Antibody]

In your `.zshrc`:

```
antibody bundle mfaerevaag/wd
```

Arch (AUR)

1. Install from the AUR

```
yay -S zsh-plugin-wd-git
# or use any other AUR helper
```

2. Then add to your `.zshrc`:

```
wd() {
    . /usr/share/wd/wd.sh
}
```

zplug

```
zplug "mfaerevaag/wd", as:command, use:"wd.sh", hook-load:"wd() { .
$ZPLUG_REPOS/mfaerevaag/wd/wd.sh }"
```

Automatic

Note: automatic install does not provide the manpage. It is also poor security practice to run remote code without first reviewing it, so you ought to look here

Run either command in your terminal:

```
curl -L https://github.com/mfaerevaag/wd/raw/master/install.sh | sh
```

or

```
wget --no-check-certificate https://github.com/mfaerevaag/wd/raw/master/install.sh -
O - | sh
```

Manual

1. Clone this repository on your local machine in a sensible location (if you know what you're doing of course all of this is up to you):

```
git clone git@github.com:mfaerevaag/wd.git ~/.local/wd --depth 1
```

2. Add `wd` function to `.zshrc` (or `.profile` etc.):

```
wd() {
    . ~/.local/wd/wd.sh
}
```

3. Install manpage (optional):

```
sudo cp ~/.local/wd/wd.1 /usr/share/man/man1/wd.1
sudo chmod 644 /usr/share/man/man1/wd.1
```

Note: when pulling and updating `wd`, you'll need to repeat step 3 should the manpage change

Completion

If you're NOT using oh-my-zsh and you want to utilize the zsh-completion feature, you will also need to add the path to your `wd` installation (`~/bin/wd` if you used the automatic installer) to your `fpath`. E.g. in your `~/.zshrc`:

```
fpath=(~/path/to/wd $fpath)
```

Also, you may have to force a rebuild of `zcompdump` by running:

```
rm -f ~/.zcompdump; compinit
```

Usage

- Add warp point to current working directory:

```
wd add foo
```

If a warp point with the same name exists, use `wd add foo --force` to overwrite it.

Note: a warp point cannot contain colons, or consist of only spaces and dots. The first will conflict in how `wd` stores the warp points, and the second will conflict with other features, as below.

You can omit point name to automatically use the current directory's name instead.

- From any directory, warp to `foo` with:

```
wd foo
```

- You can also warp to a directory within `foo`, with autocompletion:

```
wd foo some/inner/path
```

- You can warp back to previous directory and higher, with this dot syntax:

```
wd ..
wd ...
```

This is a wrapper for the zsh's `dirs` function.
You might need to add `setopt AUTO_PUSHD` to your `.zshrc` if you are not using oh-my-zsh.

- Remove warp point:

```
wd rm foo
```

You can omit point name to use the current directory's name instead.

- List all warp points (stored in `~/.warprc` by default):

```
wd list
```

- List files in given warp point:

```
wd ls foo
```

- Show path of given warp point:

```
wd path foo
```

- List warp points to current directory, or optionally, path to given warp point:

```
wd show
```

- Remove warp points to non-existent directories.

```
wd clean
```

Use `wd clean --force` to not be prompted with confirmation.

- Print usage info:

```
wd help
```

The usage will be printed also if you call `wd` with no command

- Print the running version of `wd`:

```
wd --version
```

- Specifically set the config file (default being `~/.warprc`), which is useful for testing:

```
wd --config ./file <command>
```

- Force `exit` with return code after running. This is not default, as it will *exit your terminal*, though required for testing/debugging.

```
wd --debug <command>
```

- Silence all output:

```
wd --quiet <command>
```

Configuration

You can configure `wd` with the following environment variables:

`WD_CONFIG`

Defines the path where warp points get stored. Defaults to `$HOME/.warprc`.

Testing

`wd` comes with a small test suite, run with shunit2. This can be used to confirm that things are working as they should on your setup, or to demonstrate an issue.

To run, simply `cd` into the `test` directory and run the `tests.sh`.

```
cd ./test
./tests.sh
```

Maintainers

Following @mfaerevaag stepping away from active maintainership of this repository, the following users now are also maintainers of the repo:

- @alpha-tango-kilo
- @MattLewin

Anyone else contributing is greatly appreciated and will be mentioned in the release notes!

Credit to altschuler for an awesome idea.

Hope you enjoy!