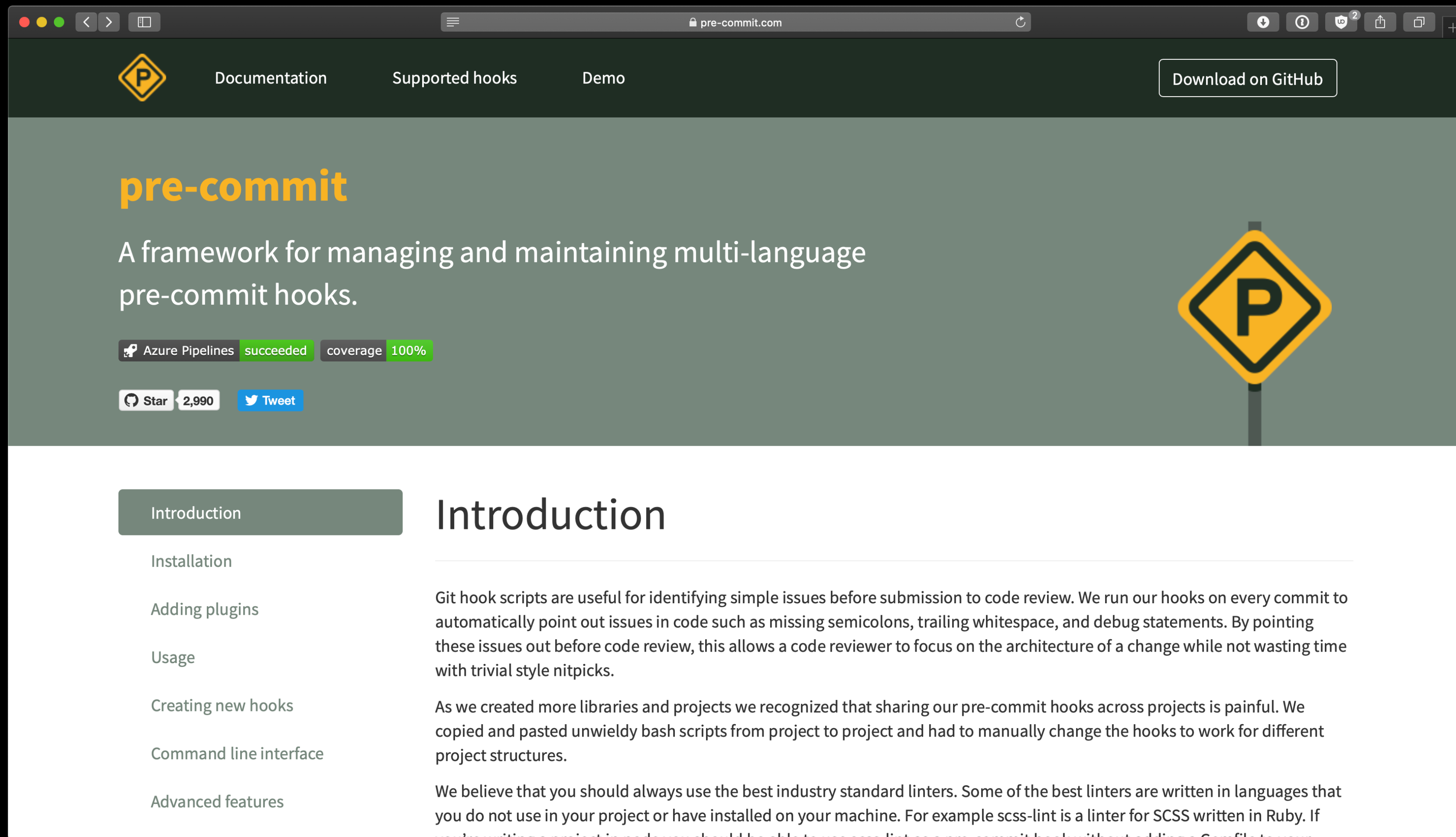


Intro to Pre-Commit

Elliot Jordan

What is Pre-Commit?



<https://pre-commit.com>

Installation

```
pip install pre-commit
```

– or –

```
brew install pre-commit
```

Configuration

```
cd ~/path/to/git_repo
```

```
touch .pre-commit-config.yaml
```

Configuration

`open -a "Atom" .pre-commit-config.yaml`



.pre-commit-config.yaml

repos:

- repo: <https://github.com/pre-commit/pre-commit-hooks>

rev: v2.3.0

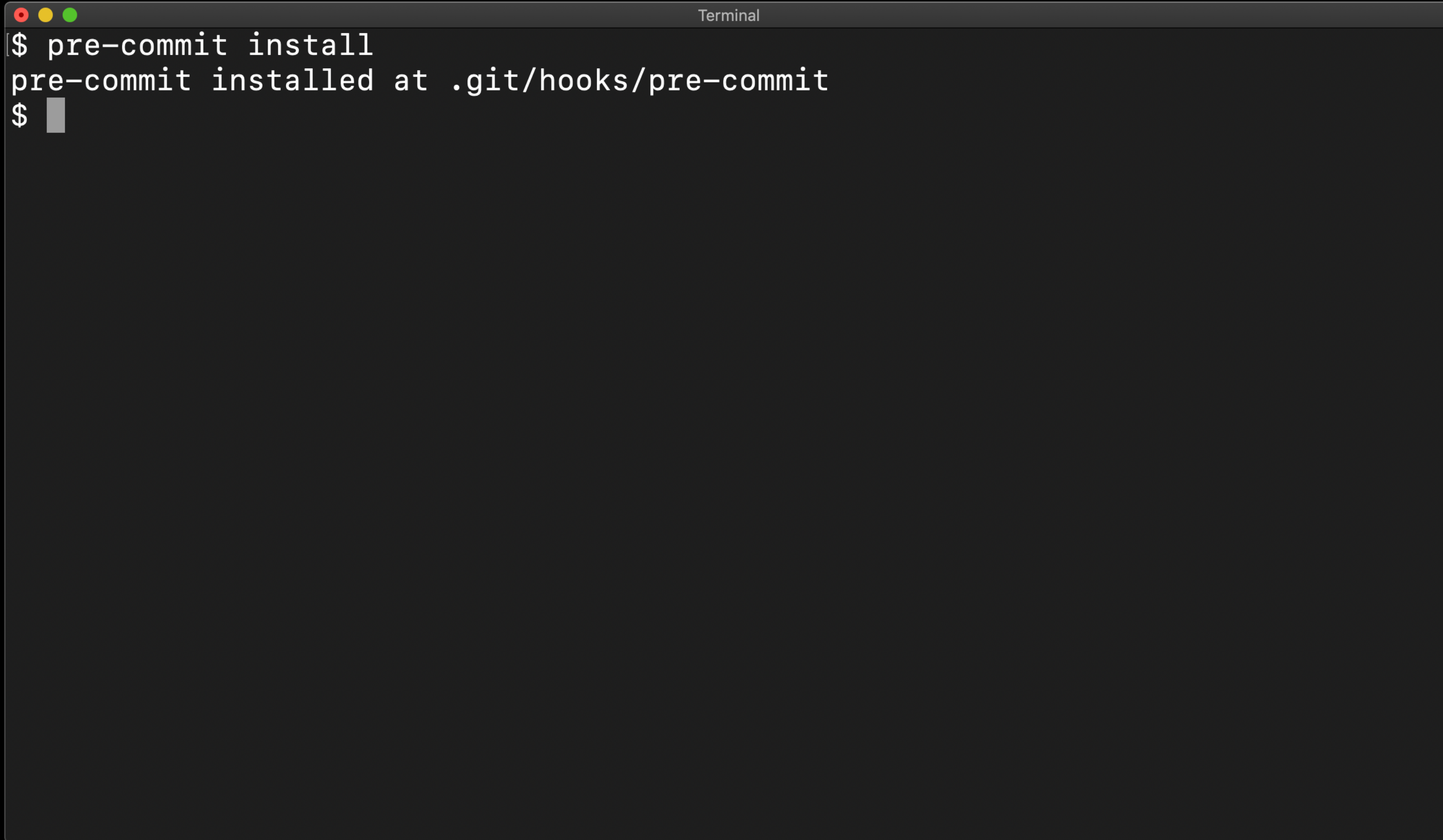
hooks:

- id: `no-commit-to-branch`

Configuration

```
pre-commit install
```

Configuration

A terminal window with a dark gray background and a title bar that says "Terminal". The window has three colored window control buttons (red, yellow, green) in the top-left corner. The terminal shows the command `pre-commit install` being executed, followed by the output `pre-commit installed at .git/hooks/pre-commit`. A new prompt `$` is shown on the next line with a small gray cursor block.

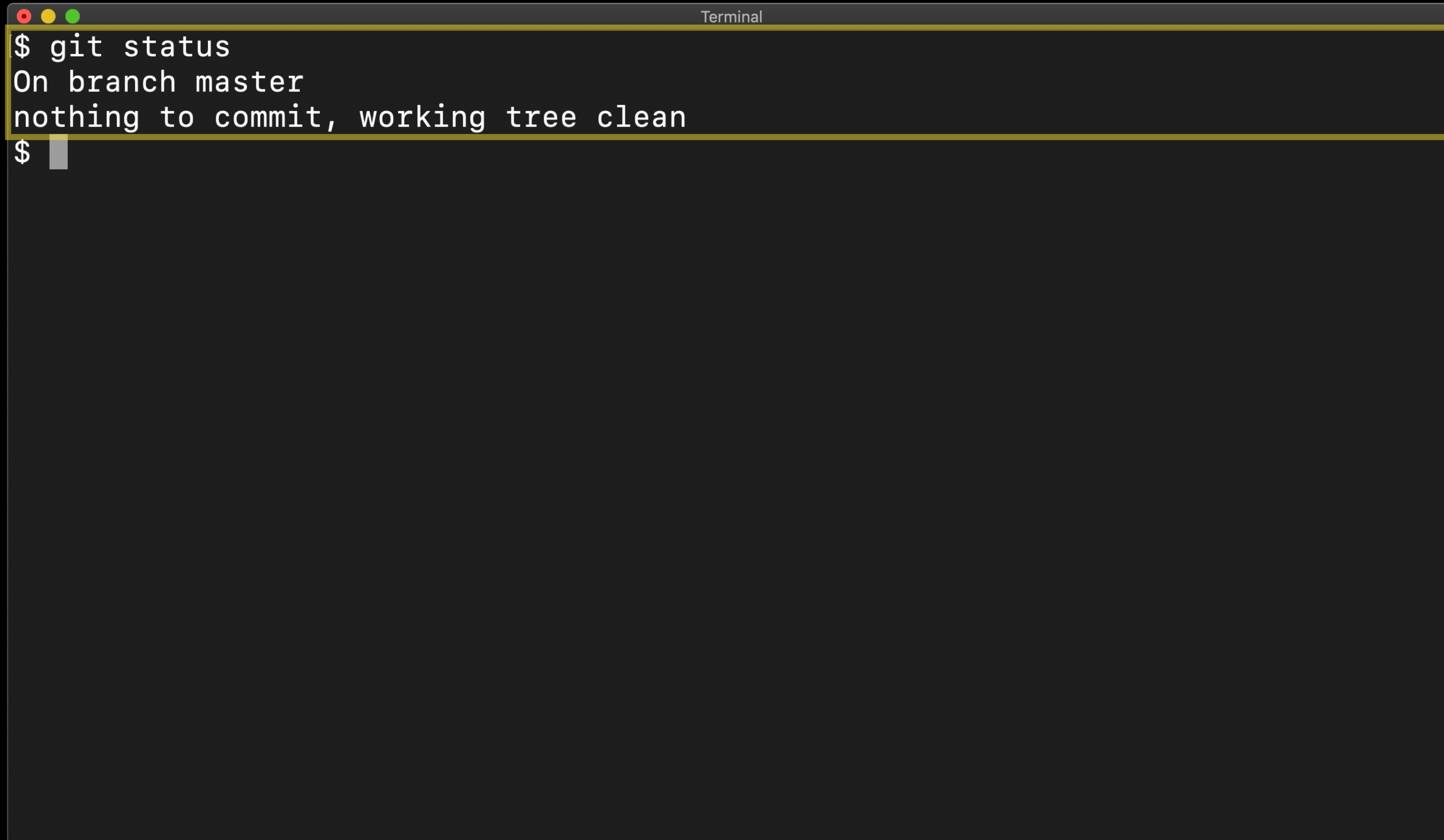
```
$ pre-commit install
pre-commit installed at .git/hooks/pre-commit
$
```

Testing the Hooks

Manually trigger all configured hooks with:

```
pre-commit run --all-files
```

Testing the Hooks

A terminal window with a dark gray background and a light gray title bar. The title bar contains three colored window control buttons (red, yellow, green) on the left and the word "Terminal" in the center. The terminal content shows the command "\$ git status" followed by its output: "On branch master", "nothing to commit, working tree clean", and a new prompt "\$" with a cursor. The text is in a monospaced font.

```
Terminal
$ git status
On branch master
nothing to commit, working tree clean
$
```

Testing the Hooks

```
Terminal
$ git status
On branch master
nothing to commit, working tree clean
$ pre-commit run --all-files
Don't commit to branch.....Failed
$
```

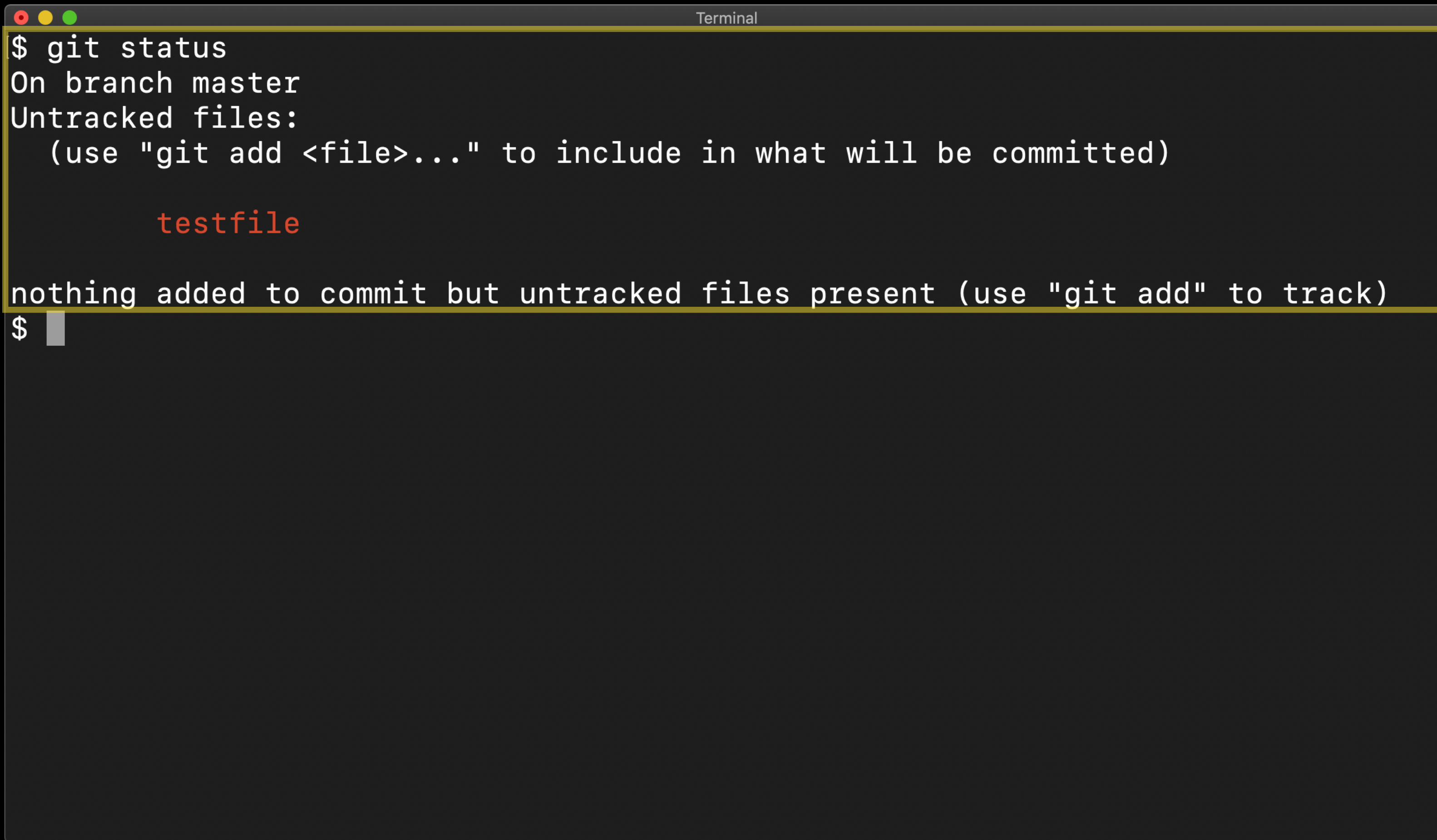
Testing the Hooks

```
Terminal
$ git status
On branch master
nothing to commit, working tree clean
$ pre-commit run --all-files
Don't commit to branch.....Failed
$ git checkout -b new-branch
Switched to a new branch 'new-branch'
$
```

Testing the Hooks

```
Terminal
$ git status
On branch master
nothing to commit, working tree clean
$ pre-commit run --all-files
Don't commit to branch.....Failed
$ git checkout -b new-branch
Switched to a new branch 'new-branch'
$ pre-commit run --all-files
Don't commit to branch.....Passed
$
```

Usage

A terminal window with a dark gray background and a yellow border. The title bar at the top shows three colored window control buttons (red, yellow, green) on the left and the word "Terminal" in the center. The terminal content shows the output of the 'git status' command. The text is as follows:

```
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    testfile

nothing added to commit but untracked files present (use "git add" to track)
$
```

The file name 'testfile' is highlighted in red. A gray cursor block is visible on the line following the last prompt '\$'.

Usage

```
Terminal
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    testfile

nothing added to commit but untracked files present (use "git add" to track)
$ git add .
$
```

Usage

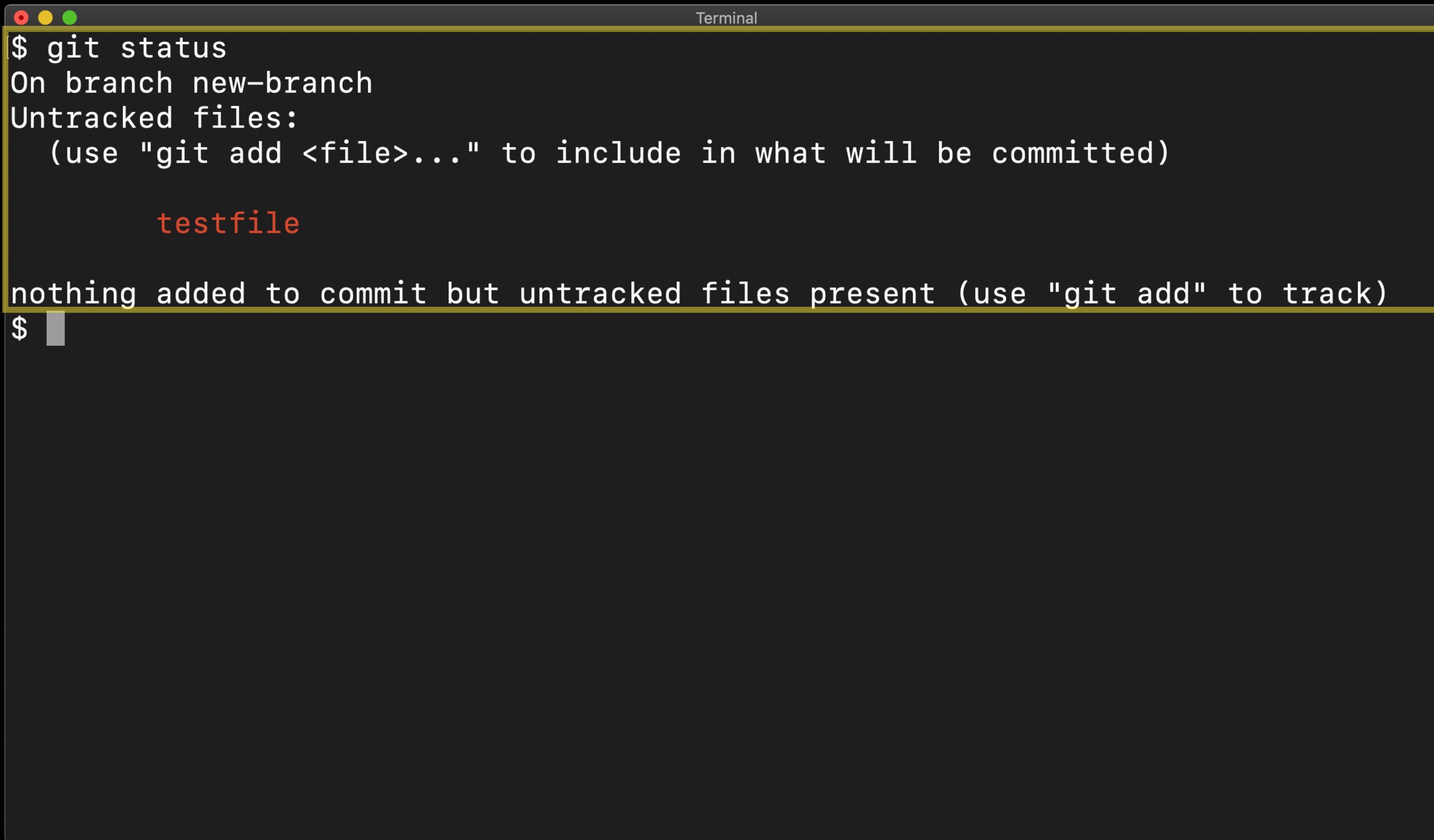
```
Terminal
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    testfile

nothing added to commit but untracked files present (use "git add" to track)
$ git add .
$ git commit -am "Create test file"
Don't commit to branch.....Failed
$
```

 Commit NOT saved.
Fix the failures and try again.

Usage

A terminal window with a dark gray background and a yellow border. The title bar at the top shows three colored window control buttons (red, yellow, green) on the left and the word "Terminal" in the center. The terminal content shows the output of the 'git status' command. The first line is the prompt '\$' followed by 'git status'. The subsequent lines are 'On branch new-branch', 'Untracked files:', and an indented line '(use "git add <file>..." to include in what will be committed)'. Below this, the filename 'testfile' is listed in red text. The next line is 'nothing added to commit but untracked files present (use "git add" to track)'. The final line shows the prompt '\$' followed by a gray cursor block.

```
$ git status
On branch new-branch
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    testfile

nothing added to commit but untracked files present (use "git add" to track)
$ █
```

Usage

```
Terminal
$ git status
On branch new-branch
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    testfile

nothing added to commit but untracked files present (use "git add" to track)
$ git add .
$
```

Usage

```
Terminal
$ git status
On branch new-branch
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    testfile

nothing added to commit but untracked files present (use "git add" to track)
$ git add .
$ git commit -am "Create test file"
Don't commit to branch.....Passed
[new-branch 83e02ad] Create test file
1 file changed, 1 insertion(+)
create mode 100644 testfile
$
```

✓ Commit saved.

Adding More Hooks



.pre-commit-config.yaml

```
repos:
- repo: https://github.com/pre-commit/pre-commit-hooks
  rev: v2.3.0
  hooks:
  - id: no-commit-to-branch
  - id: another-hook
  - id: and-so-on
```

More hooks from the same repo.

repos:

- repo: <https://github.com/pre-commit/pre-commit-hooks>
rev: v2.3.0
hooks:
 - id: no-commit-to-branch
 - id: another-hook
 - id: and-so-on
- repo: <https://github.com/another/repo>
rev: v1.2.3
hooks:
 - id: hook-from-another-repo
 - id: and-another-hook
 - id: etc-hook

Hooks from another repo.

repos:

- repo: <https://github.com/pre-commit/pre-commit-hooks>

rev: v2.3.0

hooks:

- id: **check-added-large-files**

args: [--maxkb=200]

- id: check-byte-order-marker

- id: check-case-conflict

- id: check-docstring-first

- id: **check-merge-conflict**

- id: check-symlinks

- id: **check-xml**

- id: **check-yaml**

- id: fix-encoding-pragma

- id: mixed-line-ending

- id: no-commit-to-branch

- id: sort-simple-yaml

- id: trailing-whitespace

exclude: '\.(app|framework)/'

args: [--markdown-linebreak-ext=md]

- repo: <https://github.com/python/black>

rev: 19.3b0

hooks:

- id: **black**

language_version: python3.7

- repo: <https://github.com/asottile/blacken-docs>

rev: v1.3.0

hooks:

- id: **blacken-docs**

additional_dependencies: [black==19.3b0]

Pre-Commit Hooks for Mac Admins

The screenshot shows a web browser window displaying the GitHub repository page for `homebysix/pre-commit-macadmin`. The browser's address bar shows the URL `github.com/homebysix/pre-commit-macadmin/blob/master/README.md`. The GitHub interface includes a top navigation bar with the GitHub logo, a search bar, and links for Pull requests, Issues, Marketplace, and Explore. The repository header shows the repository name, a file icon, and buttons for Unwatch (2), Star (20), and Fork (3). Below the header, there are tabs for Code, Issues (0), Pull requests (0), Actions, Projects (0), Wiki, Security, Insights, and Settings. The main content area shows the file `pre-commit-macadmin / README.md` with a branch selector set to `master`. A commit history section shows a commit by `homebysix` titled "Version bump to 1.5.0" with commit hash `771bdc9` and a timestamp of "13 hours ago". Below the commit history, the file statistics show "166 lines (109 sloc) | 5.91 KB". The README content includes the title "Pre-Commit Hooks for Mac Admins", a description of the repository's purpose, a "Requirements" section, and the beginning of an "Adding hooks to your pre-commit config" section.

homebysix / pre-commit-macadmin

Unwatch 2 Star 20 Fork 3

<> Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security Insights Settings

Branch: master pre-commit-macadmin / README.md Find file Copy path

homebysix Version bump to 1.5.0 771bdc9 13 hours ago

1 contributor

166 lines (109 sloc) | 5.91 KB Raw Blame History

Pre-Commit Hooks for Mac Admins

This repository contains hooks for [pre-commit](#) that may be useful to Mac admins, client engineers, and other Apple-focused IT professionals.

Requirements

To use these hooks, you first need to install pre-commit using the instructions here: <https://pre-commit.com/#install>

Adding hooks to your pre-commit config

Pre-Commit Hooks for Mac Admins

```
.pre-commit-config.yaml

repos:
- repo: https://github.com/homebysix/pre-commit-macadmin
  rev: v1.5.0
  hooks:
  - id: check-autopkg-recipes
    args ['--strict', '--recipe-prefix', 'com.github.homebysix.']
    exclude: Kitematic
  - id: forbid-autopkg-overrides
  - id: forbid-autopkg-trust-info
```

Public AutoPkg recipe repo.

Pre-Commit Hooks for Mac Admins



.pre-commit-config.yaml

```
repos:
- repo: https://github.com/homebysix/pre-commit-macadmin
  rev: v1.5.0
  hooks:
  - id: check-autopkg-recipe-list
  - id: check-autopkg-recipes
    args [
      '--recipe-prefix', 'com.example.autopkg.'
      '--override-prefix=local.'
  - id: check-plists
```

AutoPkg recipes & overrides.

Pre-Commit Hooks for Mac Admins



.pre-commit-config.yaml

```
repos:
- repo: https://github.com/homebysix/pre-commit-macadmin
  rev: v1.5.0
  hooks:
  - id: check-munkipkg-buildinfo
  - id: check-outset-scripts
  - id: check-plists
```

MunkiPkg sources, including Outset scripts.

Pre-Commit Hooks for Mac Admins

```


```

 .pre-commit-config.yaml

repos:
- repo: https://github.com/homebysix/pre-commit-macadmin
 rev: v1.5.0
 hooks:
 - id: check-munki-pkginfo
 args: [
 '--required-keys', 'category', 'description', 'developer', 'name', 'version',
 '--catalogs', 'stable', 'testing',
 '--categories', 'Communication', 'Creative', 'Media Viewers', 'Productivity', 'Web Browsers',
 '--']
 - id: check-munkiadmin-scripts
 - id: check-plists

```


```

Munki repo.

Pre-Commit Hooks for Mac Admins

Bad pkginfo.

Catalogs should be an array/list.

Missing description.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>

    [...snip...]

    <key>autoremove</key>
    <false/>
    <key>catalogs</key>
    <string>testing</string>
    <key>category</key>
    <string>Web Browsers</string>
    <key>developer</key>
    <string>Brave Software</string>

    [...snip...]

</dict>
</plist>
```

Pre-Commit Hooks for Mac Admins

Pre-commit saves the day!

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<dict>
```

```
Check Munki Pkginfo Files.....Failed
hookid: check-munki-pkginfo
```

```
pkginfo/apps/Brave-74.0.64.76.plist: missing required key "description"
pkginfo/apps/Brave-74.0.64.76.plist: pkginfo key catalogs should be type <class 'list'>, not type <class 'str'>
```

```
Check MunkiAdmin Scripts.....Passed
```

```
<key>developer</key>
<string>Brave Software</string>

[...snip...]

</dict>
</plist>
```

Thank you!

Elliot Jordan

  homebysix
 elliotjordan