



# Introduction to Git

## Python Fundamentals for Engineers and Manufacturers



## git --local-branching-on-the-cheap



Search entire site...

Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is **easy to learn** and has a **tiny footprint with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient **staging areas**, and **multiple workflows**.



Learn Git in your browser for free with [Try Git](#).



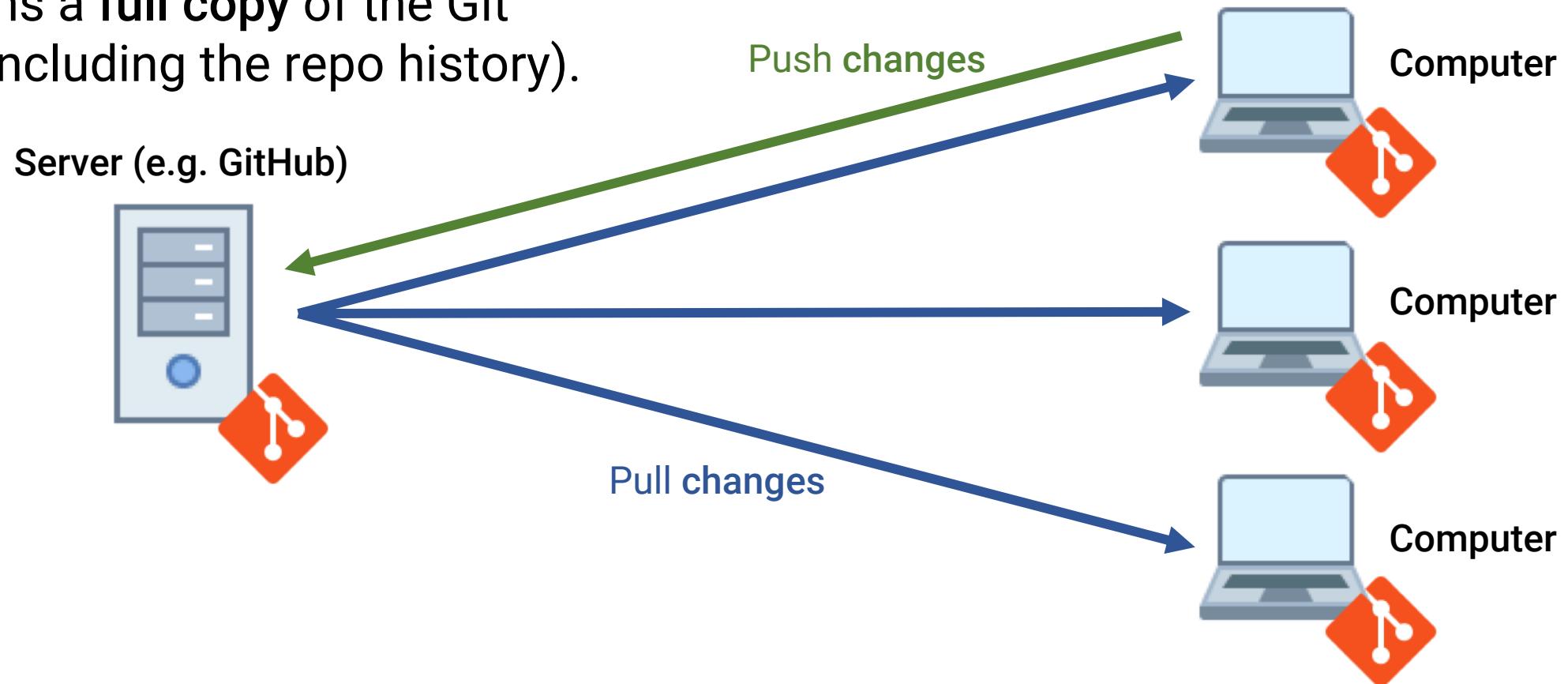
From <https://git-scm.com/>

- **No one is perfect** – you need a way to roll-back changes and explore lines of development.
- Developers can “go back in time” and see every change to a file (e.g. to understand why a change was made).
- Provides a back-up for your repo (if you are using [GitHub](#), for example).
- Supports multiple people working on a repo at the same time.

# What makes Git “distributed”?

The Server and every Computer contains a **full copy** of the Git repo (including the repo history).

Workshop repo  
<http://bit.ly/2opFlZ6>

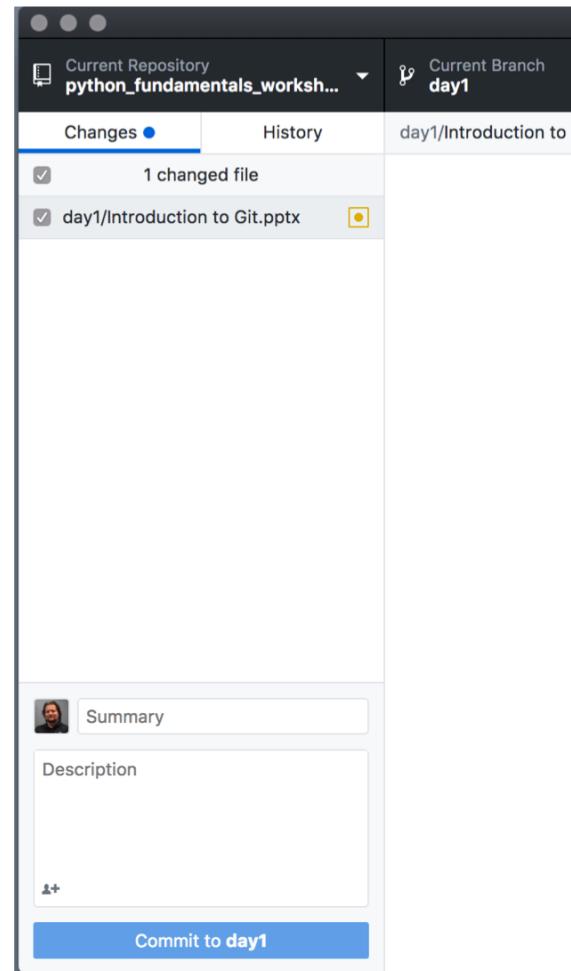


# Git “user interface options”

Workshop repo  
<http://bit.ly/2opFlZ6>



## GitHub Desktop



Current Repository  
python\_fundamentals\_workshop... ▾

Current Branch  
day1

Changes • History day1/Introduction to G

1 changed file

day1/Introduction to Git.pptx

Summary

Description

+ Commit to day1

## Git in terminal

```
adamjcook at rhazes in ~/projects/python_fundamentals_workshop on day1
$ git status
On branch day1
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   day1/Workshop Housekeeping.pdf
    modified:   day1/Workshop Housekeeping.pptx

no changes added to commit (use "git add" and/or "git commit -a")

adamjcook at rhazes in ~/projects/python_fundamentals_workshop on day1 [!]
$ git add --all

adamjcook at rhazes in ~/projects/python_fundamentals_workshop on day1 [+]
$ git status
On branch day1
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   day1/Workshop Housekeeping.pdf
    modified:   day1/Workshop Housekeeping.pptx

adamjcook at rhazes in ~/projects/python_fundamentals_workshop on day1 [+]
$ git commit
[day1 a9a0cba] slightly adjust title style on housekeeping slide deck
 2 files changed, 0 insertions(+), 0 deletions(-)

adamjcook at rhazes in ~/projects/python_fundamentals_workshop on day1
$ git push origin day1
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 126.92 KiB | 1.35 MiB/s, done.
Total 5 (delta 4), reused 0 (delta 0)
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To github.com:smevirtual/python_fundamentals_workshop.git
 b280828..a9a0cba day1 -> day1

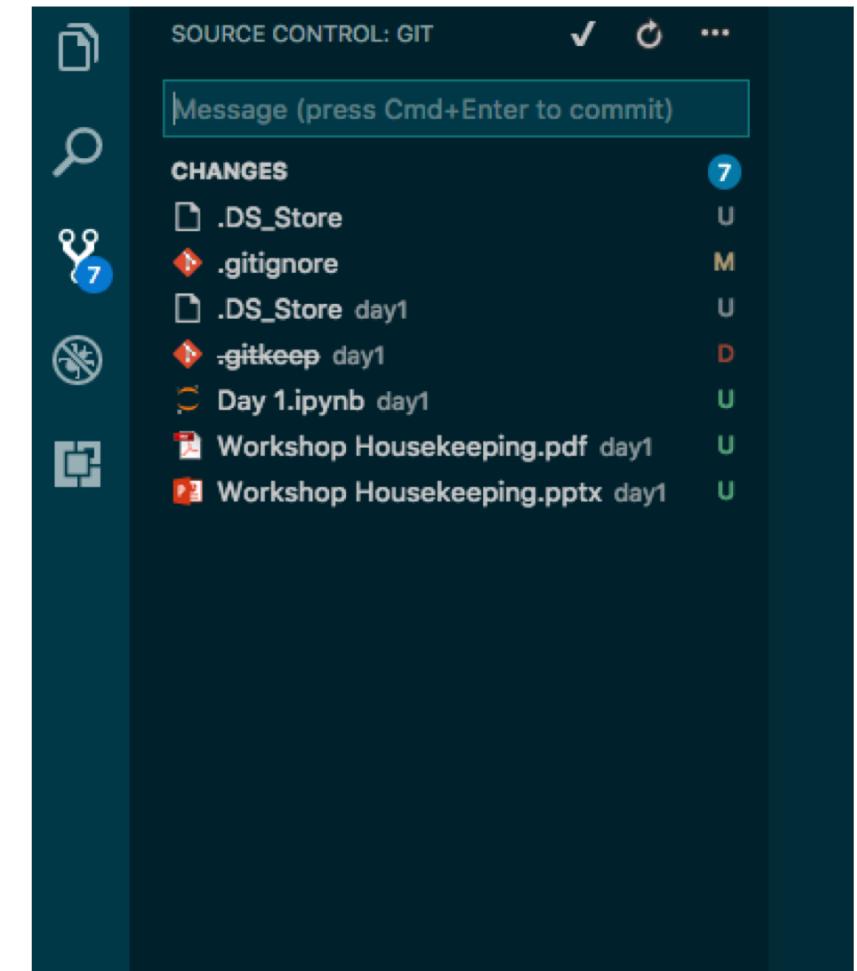
adamjcook at rhazes in ~/projects/python_fundamentals_workshop on day1
$ git status
On branch day1
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   day1/Introduction to Git.pptx

no changes added to commit (use "git add" and/or "git commit -a")

adamjcook at rhazes in ~/projects/python_fundamentals_workshop on day1 [!]
$
```

## Integrated into code editor



SOURCE CONTROL: GIT ✓ ⌂ ...

Message (press Cmd+Enter to commit)

CHANGES 7

- .DS\_Store U
- .gitignore M
- .DS\_Store day1 U
- .gitkeep day1 D
- Day 1.ipynb day1 U
- Workshop Housekeeping.pdf day1 U
- Workshop Housekeeping.pptx day1 U

# Remote Repositories (GitHub UI)

Workshop repo  
<http://bit.ly/2opFIZ6>



Username or organization

Repository (repo) name

For giving “kudos”

For creating your own copy of this repo

The screenshot shows a GitHub repository page for 'smevirtual / python\_fundamentals\_workshop'. The top navigation bar includes 'Code', 'Issues 3', 'Pull requests 0', 'Projects 0', 'Wiki', 'Insights', and 'Settings'. Below the navigation bar, the repository description reads: 'All content (slide decks and code) for the Python Fundamentals for Engineers and Manufacturers workshop.' A red arrow points from the 'Issues 3' button to the text 'For filing bugs and providing feedback'. Another red arrow points from the 'Pull requests 0' button to the text 'For reviewing your own or other people's contribution to the repo'. A red arrow points from the 'Star 0' button to the text 'For giving “kudos”'. A red arrow points from the 'Fork 0' button to the text 'For creating your own copy of this repo'. The repository stats show 4 commits, 2 branches, 0 releases, 1 contributor, and MIT license. The commit history lists three commits by user 'adamjcook': 'add several sections to the README.md' (3 days ago), 'add dummy folders for all three days and Python samples' (3 days ago), and 'add dummy folders for all three days and Python samples' (3 days ago). The bottom right corner features a green 'Clone or download' button.

For filing bugs and providing feedback

For reviewing your own or other people's contribution to the repo

# Remote Repository URL

Workshop repo  
<http://bit.ly/2opFlZ6>

A screenshot of a GitHub repository page. At the top, there are buttons for "Create new file", "Upload files", "Find file", and a green "Clone or download" button with a dropdown arrow. The dropdown menu is open, showing two main options: "Clone with SSH" and "Use HTTPS". The "Use HTTPS" option is highlighted with a red arrow. Below these are two URLs: "git@github.com:smevirtual/python\_fundament" (SSH) and "https://github.com/smevirtual/python\_fundament" (HTTPS). There are also buttons for "Open in Desktop", "Download ZIP", and a timestamp "3 days ago".

Clone with SSH ?

Use HTTPS

git@github.com:smevirtual/python\_fundament

Open in Desktop

Download ZIP

3 days ago

Repo cloning via GitHub Desktop

Mostly used if you just want a copy of the source code.

Git over HTTPS (for repo cloning via terminal)

Git over SSH (for repo cloning via terminal)

Usually, called origin

## Two types of Git URLs

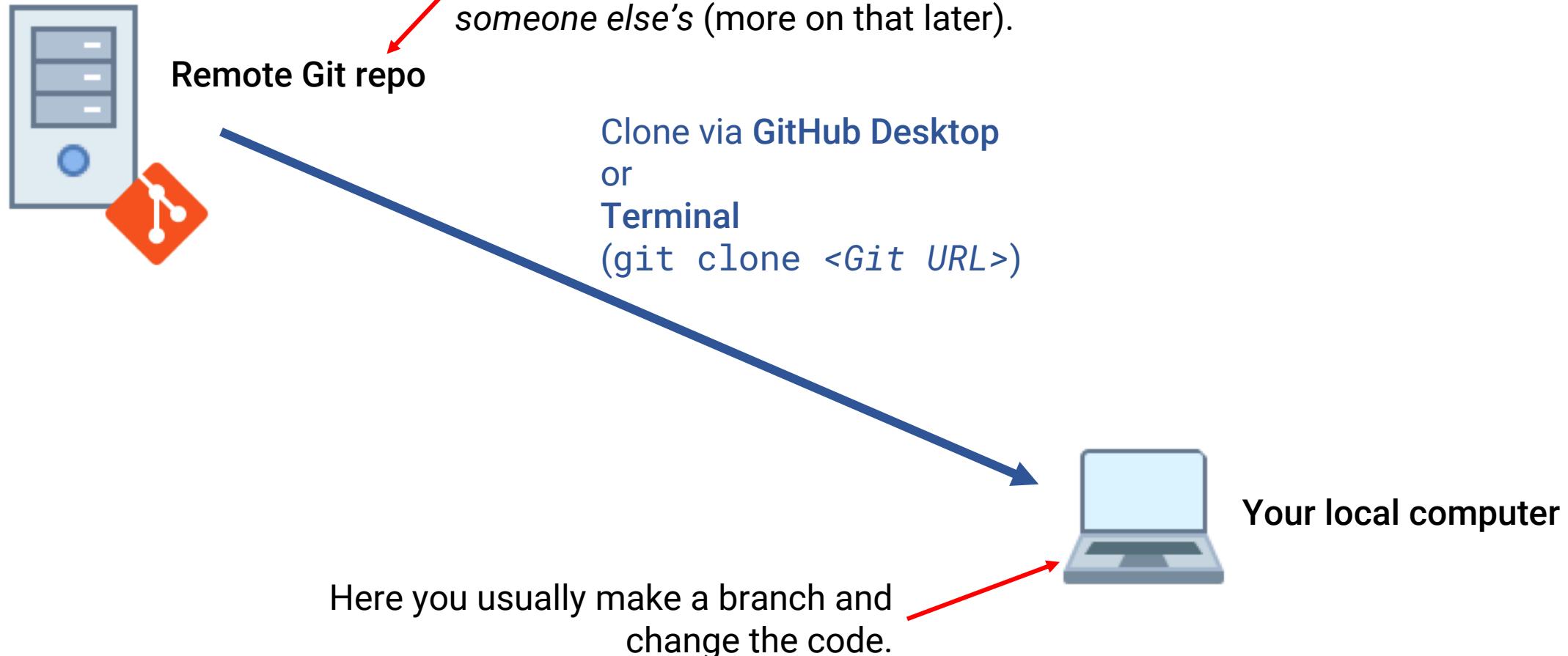
git@github.com:smevirtual/python\_fundamentals\_workshop.git

Or

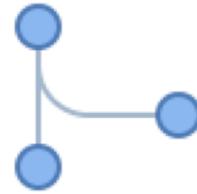
[https://github.com/smevirtual/python\\_fundamentals\\_workshop.git](https://github.com/smevirtual/python_fundamentals_workshop.git)

# Cloning

Workshop repo  
<http://bit.ly/2opFlZ6>



# Branching

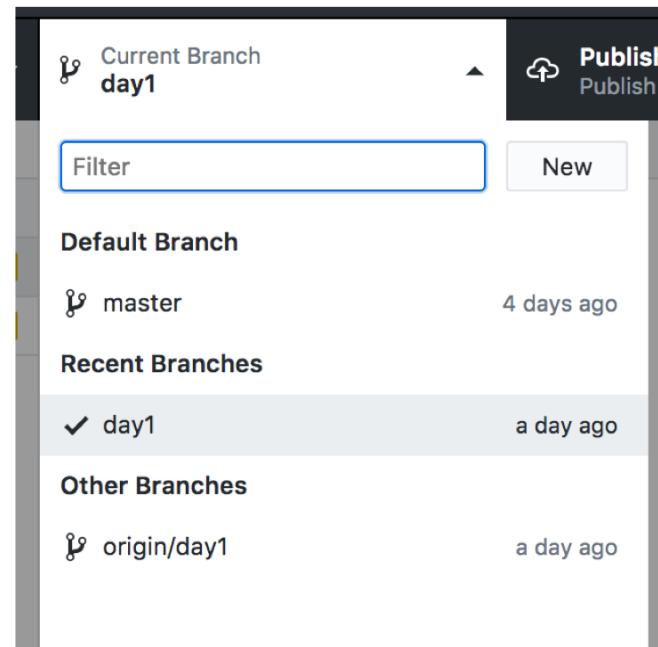


Terminal command

```
git branch <branch name>
```

Terminal command

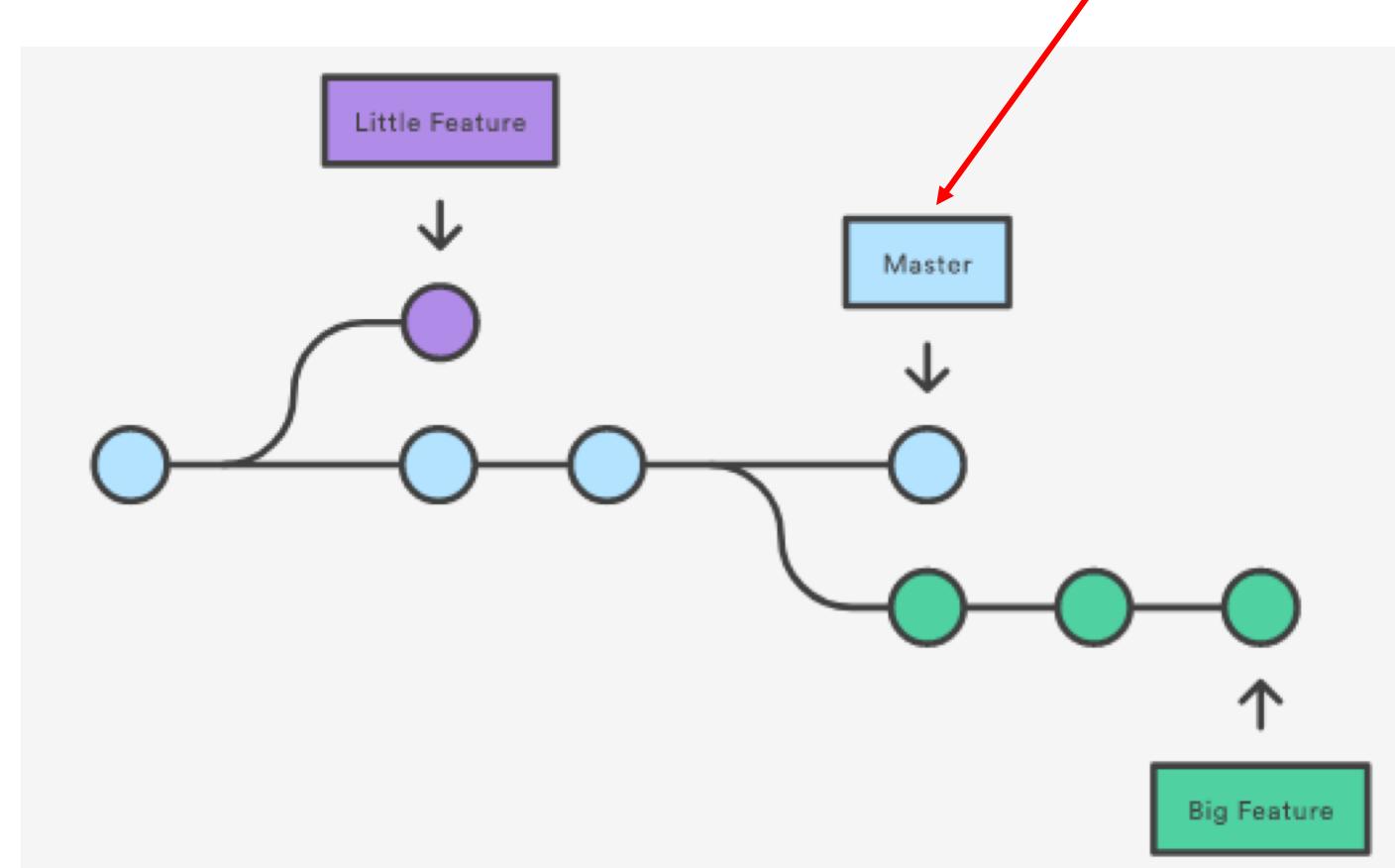
```
git checkout <branch name>
```



Workshop repo  
<http://bit.ly/2opFlZ6>



Usually, the master branch is the main development branch.



Terminal command  
git diff

Fix lint issues Browse files

develop (#2154) @blueprintjs/webpack-build-scripts@0.5.1 ... @blueprintjs/core@2.0.0-rc.2

adidahya committed 6 days ago 1 parent 78d39ee commit cefd2dd0521bd3987e064f864a4e9196c408806e

Showing 3 changed files with 18 additions and 17 deletions. Unified Split

packages/core/src/components/callout/callout.tsx

View

```
@@ -36,14 +36,15 @@ export class Callout extends React.PureComponent<ICalloutProps & React.HTMLAttri
36     const iconName = this.getIconName();
37     const classes = classNames(Classes.CALLOUT, Classes.intentClass(intent), className);
38
39 -    const maybeIcon = iconName === undefined ? undefined : (
40 -        <span className={Classes.CALLOUT_ICON}>
41 -            <Icon icon={iconName} iconSize={Icon.SIZE_LARGE} />
42 -        </span>
43 -    );
44 -    const maybeTitle = title === undefined ? undefined : (
45 -        <h5 className={Classes.CALLOUT_TITLE}>{title}</h5>
46 -    );
47
48     return (
49         <div className={classes} {...htmlProps}>
```

Removed lines Added lines

36 const iconName = this.getIconName();
37 const classes = classNames(Classes.CALLOUT, Classes.intentClass(intent), className);
38
39 + const maybeIcon =
40 + iconName === undefined ? (
41 + undefined
42 + ) : (
43 + <span className={Classes.CALLOUT\_ICON}>
44 + <Icon icon={iconName} iconSize={Icon.SIZE\_LARGE} />
45 + </span>
46 + );
47 + const maybeTitle = title === undefined ? undefined : <h5 className={Classes.CALLOUT\_TITLE}>{title}</h5>;
48
49 return (
50 <div className={classes} {...htmlProps}>

# .gitignore File

Workshop repo  
<http://bit.ly/2opFIZ6>



## gitignore.io

### gitignore.io

Create useful .gitignore files for your project

Search Operating Systems, IDEs, or Programming Languages

Create

```
# Created by https://www.gitignore.io/api/python

### Python ###
# Byte-compiled / optimized / DLL files
__pycache__/
*.py[cod]
*$py.class

# C extensions
*.so

# Distribution / packaging
.Python
build/
develop-eggs/
dist/
downloads/
eggs/
.eggs/
lib/
lib64/
parts/
sdist/
var/
wheels/
*.egg-info/
.installed.cfg
*.egg
```

Example of a Python  
.gitignore file contents.

Certain operating systems, code editors, development environments and runtimes create files that you do not want to commit (track changes).

Put a .gitignore file in your repo root.

# Adding and Committing

Workshop repo  
<http://bit.ly/2opFlZ6>



Terminal command  
`git add --all`

Terminal command  
`git commit`

Add change to staging area.

Commits the staged snapshot to the repo history  
(you can add a description of the changes made).

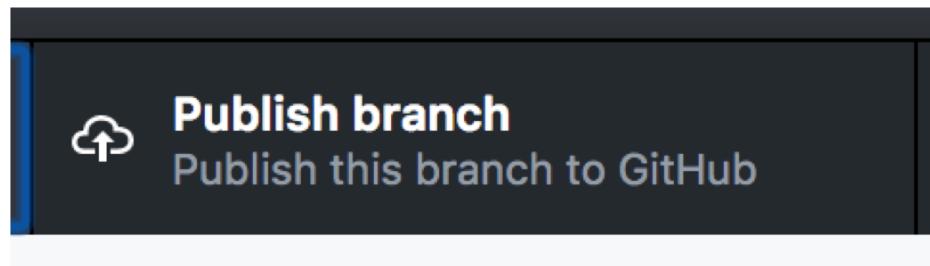
# Pushing or Publishing

Workshop repo  
<http://bit.ly/2opFlZ6>



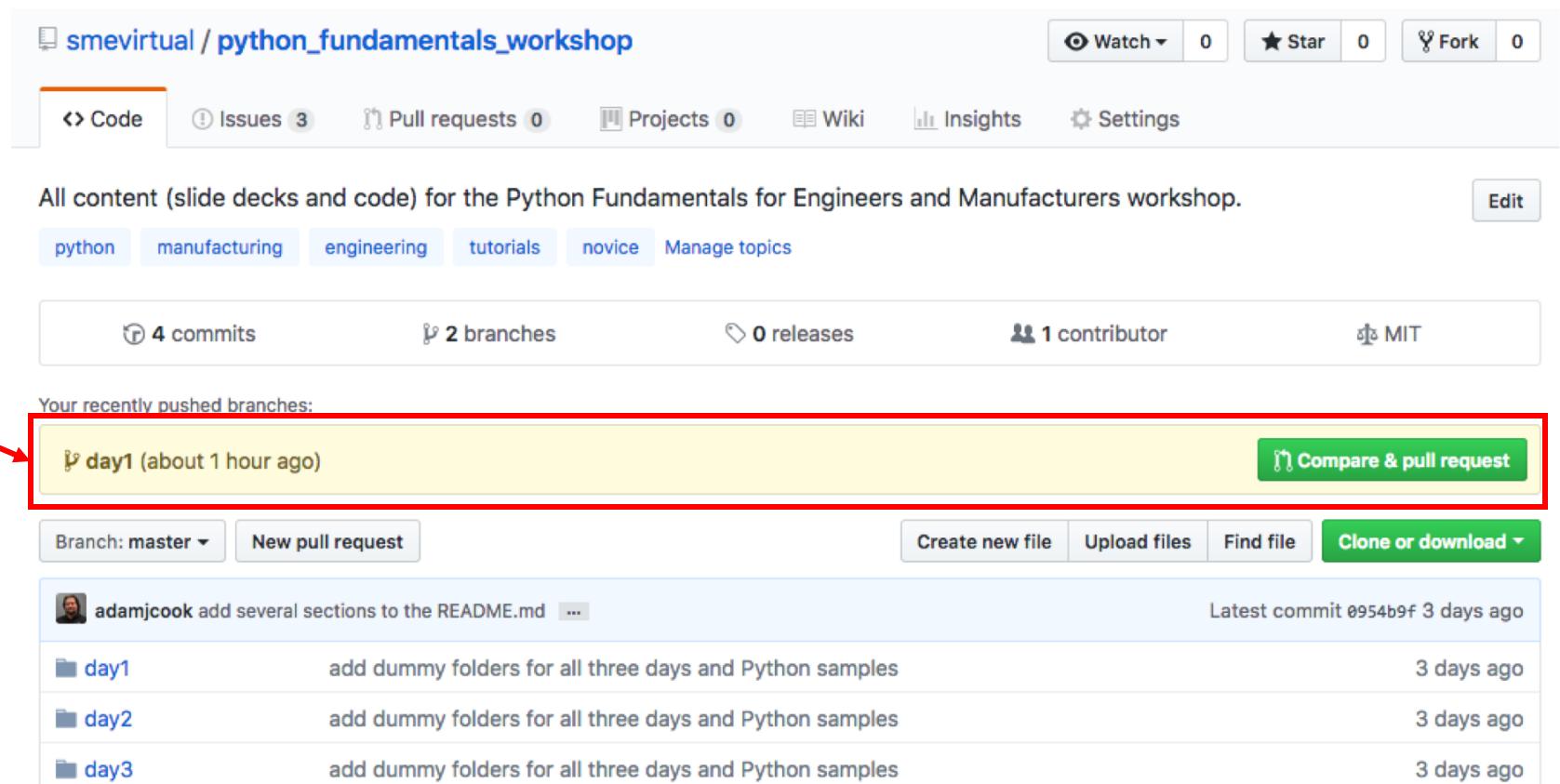
Terminal command  
`git push origin <local branch name>`

Pushes commits made on your local branch to the remote repo.



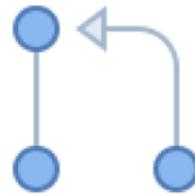
Once you push your local branch to your remote repo (e.g. forked GitHub repo), you can create a pull request.

A **pull request** gives you or the owner of the remote repo the opportunity to review the changes before accepting them officially into the repo.



The screenshot shows a GitHub repository page for 'smevirtual / python\_fundamentals\_workshop'. The repository description is 'All content (slide decks and code) for the Python Fundamentals for Engineers and Manufacturers workshop.' It has 3 issues, 0 pull requests, 0 projects, 0 wiki pages, and 0 insights. There are 4 commits, 2 branches, 0 releases, 1 contributor, and an MIT license. A red arrow points from the text above to the 'day1' branch entry in the 'Your recently pushed branches:' list. The 'Compare & pull request' button next to it is also highlighted with a red box.

Branch	Commit Message	Date
master	adamjcook add several sections to the README.md	3 days ago
day1	add dummy folders for all three days and Python samples	3 days ago
day2	add dummy folders for all three days and Python samples	3 days ago
day3	add dummy folders for all three days and Python samples	3 days ago



Terminal command

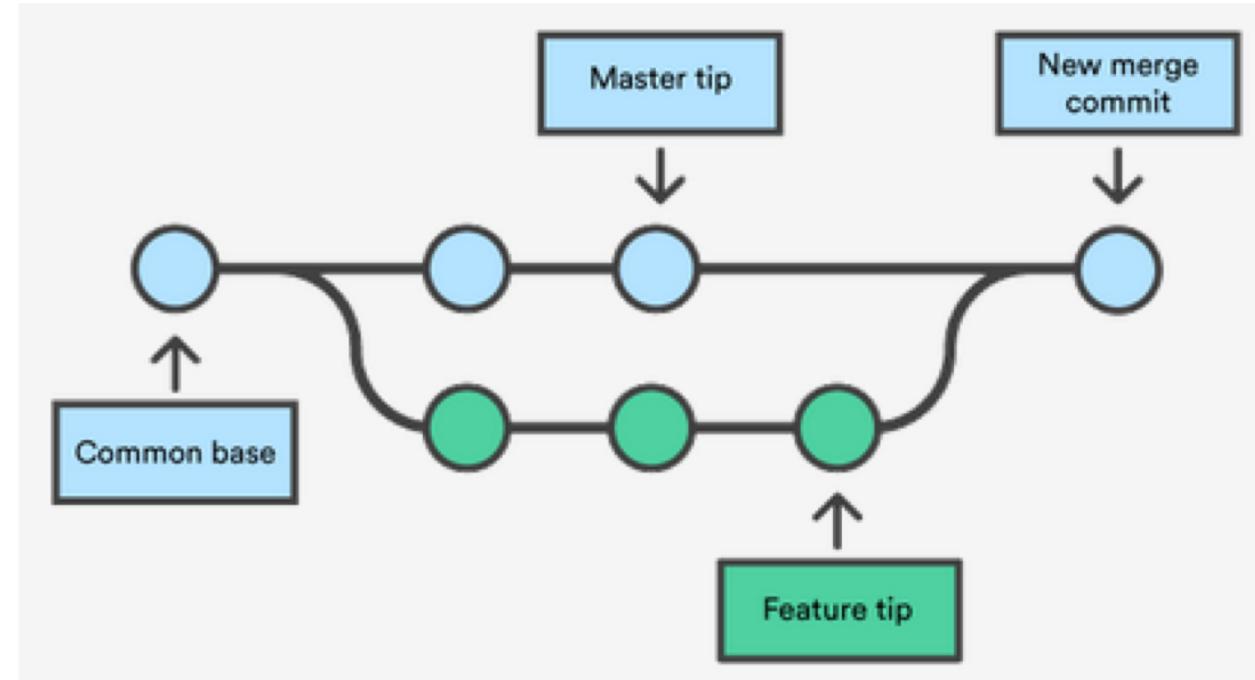
```
git pull <upstream remote branch> master
```

Terminal command

```
git checkout master
```

Terminal command

```
git merge master <upstream repo alias>/master
```



# Issues

Workshop repo  
<http://bit.ly/2opFIZ6>



Some repos will have contributing guidelines. If so, this message will appear.

Search for open and closed issues and/or issues by label.

Issue number

Issue label

The screenshot shows the GitHub Issues page for the `palantir/blueprint` repository. At the top, there are navigation links for Code, Issues (273), Pull requests (12), Projects (4), Wiki, and Insights. To the right are buttons for Watch (212), Unstar (8,400), Fork (529), and a Dismiss button for a modal message. The modal message says: "Want to submit an issue to palantir/blueprint? If you have a bug or an idea, read the [contributing guidelines](#) before opening an issue. Issues labeled `help wanted` can be good first contributions." Below the modal is a "Create a new issue" button. The main area shows a search bar with filters set to "is:issue is:open". There are buttons for Labels and Milestones. Below the search bar, it shows 273 Open and 799 Closed issues. The issues listed include:

- #2181: Card missing onDragStart, onMouseEnter, and onMouseLeave event handlers (Type: feature request)
- #2177: [Core/Menus] `pt-text-overflow-ellipsis` cuts off descenders for menu items (Domain: documentation, Package: core, Type: bug)
- #2176: Skeleton does not hide icons (Package: core, Type: bug)
- #2175: [label] missing space between text and helperText (Domain: design, Package: core, Type: bug, help wanted)
- #2173: Basic support for building on Windows (Domain: tooling, Type: task)

All icons from [Icons8](#)



1. Clone your GitHub repo to your local computer.
2. Create a new branch for your work.
3. Checkout the new branch that you created.
4. Make code changes.
5. Add and commit changes.
6. Push changes to your repo.
7. Create a pull request on GitHub.
8. Accept pull request to merge your changes.
9. Change back to your master branch and merge changes back into the master branch on your computer.



1. Fork the repo on GitHub.
2. Clone the forked repo to your local computer.
3. Create a new branch for your work.
4. Checkout the new branch that you created.
5. Make code changes.
6. Add and commit changes.
7. Push changes to your repo.
8. Create a pull request on GitHub (the repo owner will review it and merge it in or reject it).

If you use Git, you are practically guaranteed to be safe. You really have to go out of your way to lose data, lose changes or screw up your codebase.

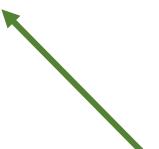
# GitHub

[github.com](https://github.com)



[gitlab.com](https://gitlab.com)

# GitLab



Good choice if you want to host your own Git server.



Atlassian

# Bitbucket

[bitbucket.org](https://bitbucket.org)



Most permissive ↑

<a href="#">Public Domain (Unlicense)</a>
<a href="#">MIT License</a>
<a href="#">BSD Licenses</a>
<a href="#">Apache 2.0</a>
<a href="#">Mozilla Public License 2.0</a>
GNU LGPLv2
GNU GPLv2
<a href="#">GNU GPLv3</a>
<a href="#">GNU GPLv3</a>
<a href="#">No license or proprietary license</a>

Least permissive ↓

All icons from [Icons8](#)

There are other “open-source” **licenses** for non-software projects too.

If the license has been altered or it is not a standard open-source license or you do not understand the license, **contact a lawyer before using the code!**

- **Warning.** Some open-source communities are more inviting than others – particularly with beginners.
- Look for repos with a Code of Conduct and/or repos that support `good-beginner-issue` labels. See [here](#) for good “beginner friendly” repos.
- **Always** check for contributing guidelines in the repo (there should be a `CONTRIBUTING.md` file in the repo root).
- **Also** check for the repo guidelines on code convention, issue formats and pull request procedures.
- **Note.** SME Virtual Network will **always** be inviting and supportive of programming beginners.

## SME Virtual Network Code of Conduct

Contact: [chapters@sme.org](mailto:chapters@sme.org)

As leaders of the SME Virtual Network, we pledge to respect everyone who contributes by providing feedback, helping others, posting issues, updating documentation, submitting pull requests and any other activities.

Communication through any of SME Virtual Networks's channels (Facebook, LinkedIn, YouTube, GitHub, Slack, etc.) must be constructive and never resort to personal attacks, trolling, public or private harassment, insults or other unprofessional conduct.

We promise to extend courtesy and respect to everyone involved in this community regardless of gender, gender identity, sexual orientation, disability, age, race, ethnicity, religion or level of experience. We expect anyone participating in this community to do the same.

If any member of the community violates this code of conduct (including any SME Virtual Network leaders), the SME Virtual Network leadership may take action - up to and including permanently banning said member from all SME Virtual Network communication channels and accounts.

If you are subject to or are a witness to unacceptable behavior (even by the SME Virtual Network leadership), or have any other concerns, please email us at [chapters@sme.org](mailto:chapters@sme.org).

*Adapted from the [Google code of conduct](#).*

Live document: <http://bit.ly/2ov3YD0>

Want to learn more about Git?

Workshop repo  
<http://bit.ly/2opFlZ6>



Two very good resources are available

<http://bit.ly/1fUdUiY>

For terminal beginners

and

<http://bit.ly/1MRoX7u>

For advanced users