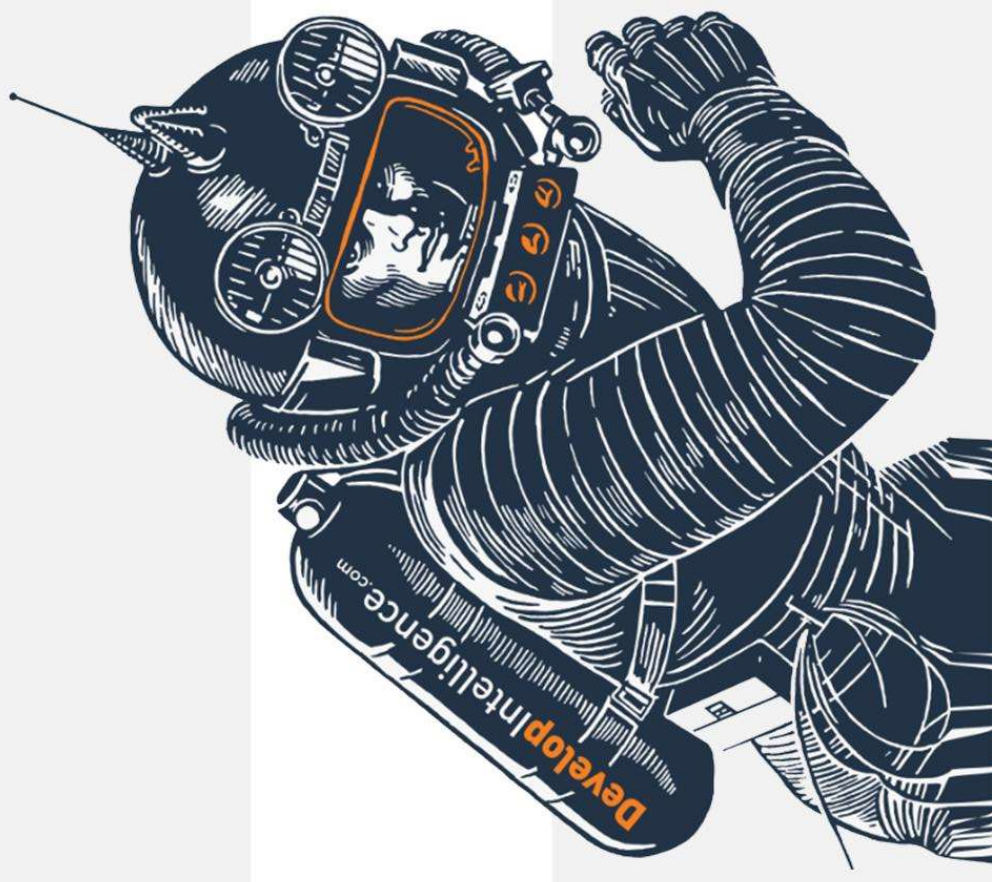


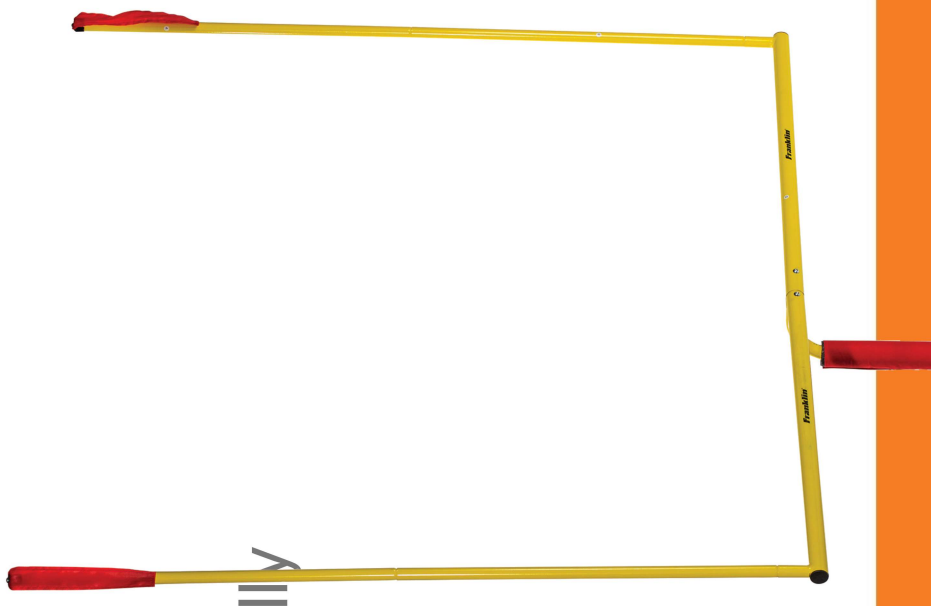
Repositories





Goals

1. Define the term repository
2. Explain how to grab a public repo off github
3. Explain how to create a new repository locally

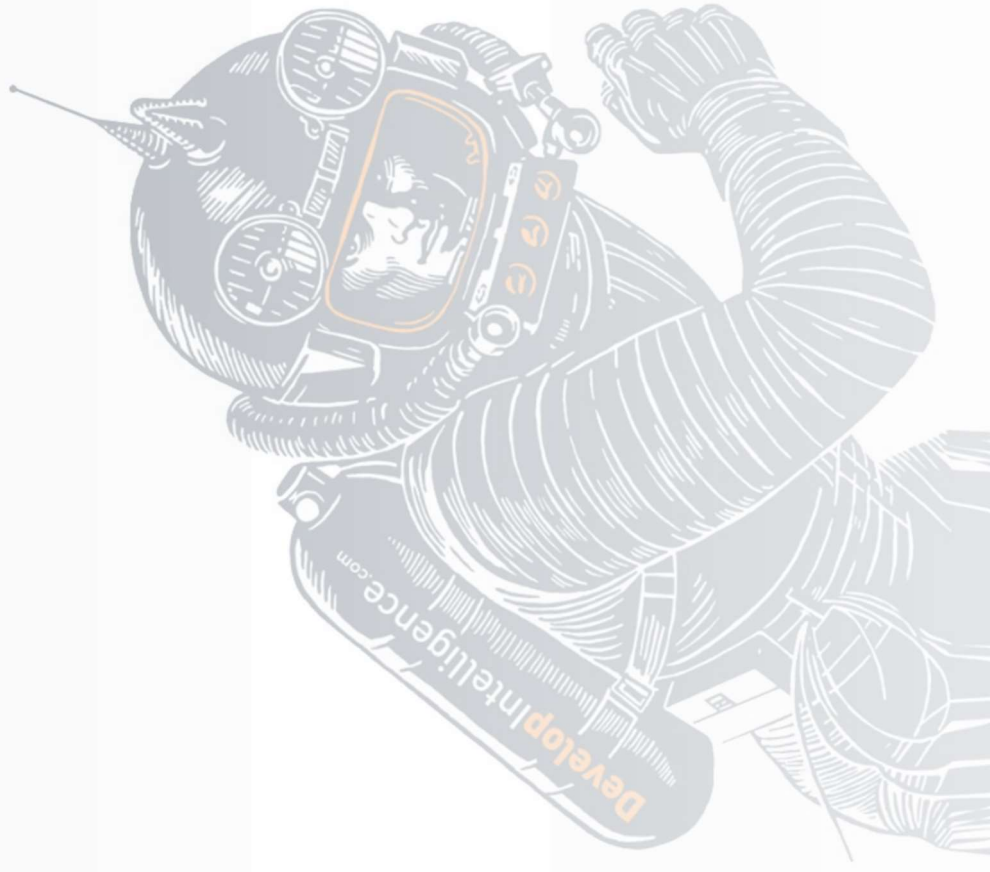




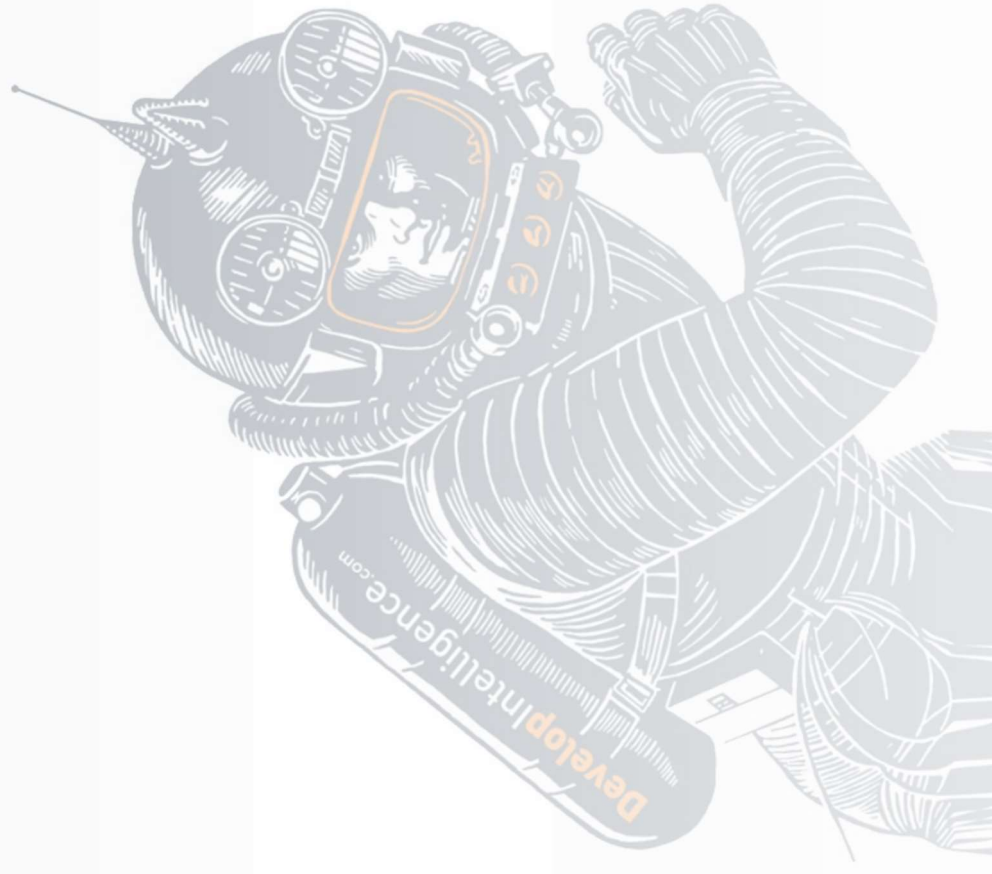
Roadmap

1. Background
2. Cloning
3. Local Repositories





Background





*A **Git repository** is a virtual storage of your project. It allows you to save versions of your code, which you can access when needed.*



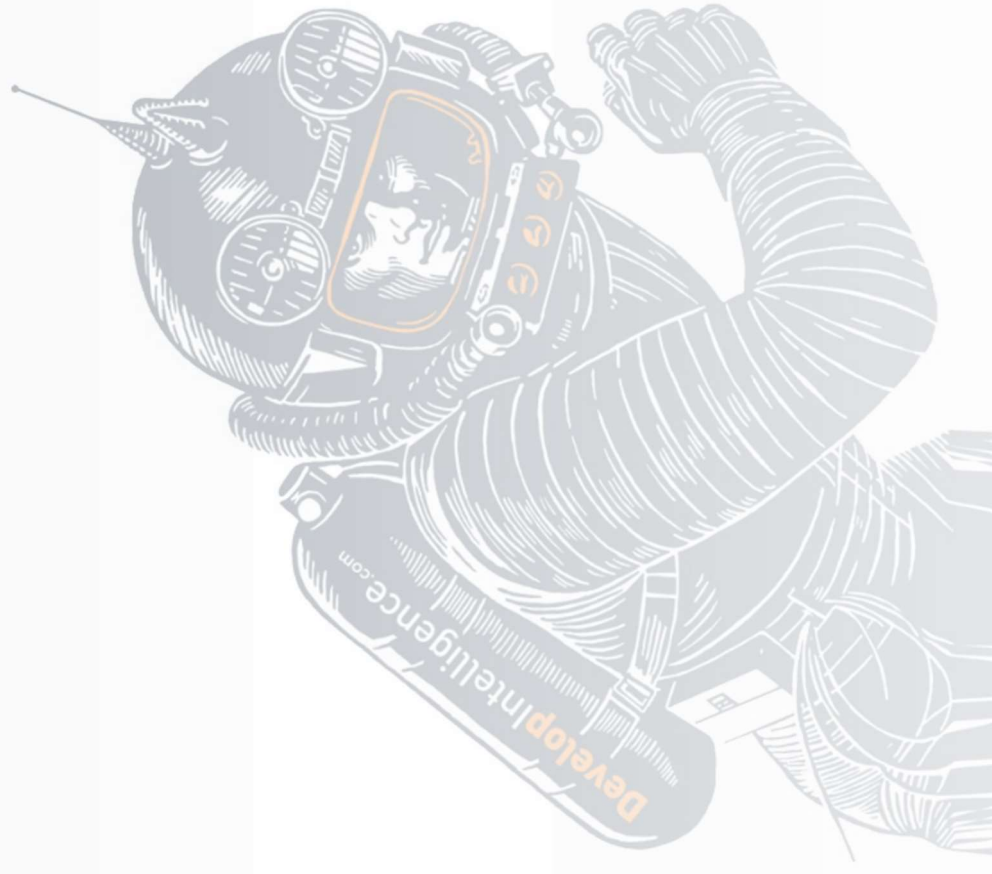
*A **repository** is a data structure that stores metadata for a set of files or directory structure... Some of the metadata that a repository contains includes a historical record of changes in the repository, a set of commit objects, and a set of references to commit objects, called heads.*



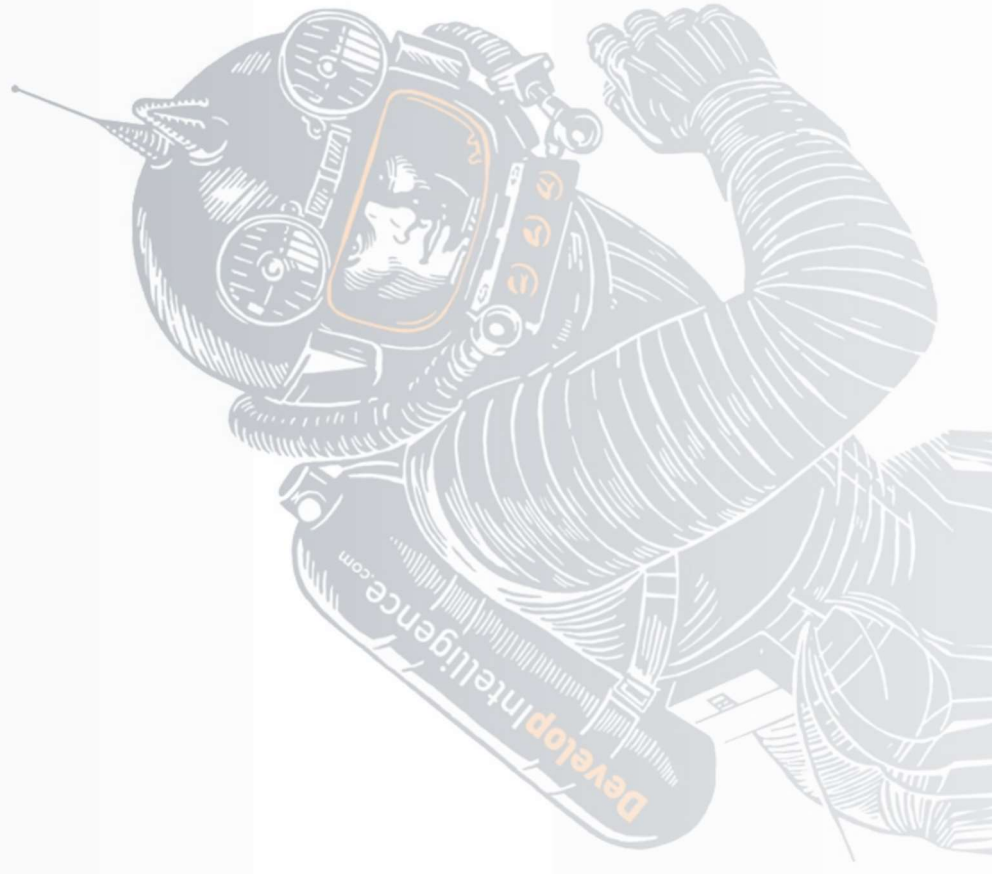
Metadata

- Commits (changes)
- Branches (alternative versions)
- Tags





Cloning





Overview

- The easiest way to get started is to grab someone else's repository
- Decide on a place in your file system to keep repositories
- Something like--
 - c:/repos
 - ~/workspace



Clone

git clone is primarily used to point to an existing repo and make a clone or copy of that repo at in a new directory, at another location. The original repository can be located on the local filesystem or on remote machine accessible supported protocols. **The git clone command copies an existing Git repository.**



Examples

- git clone `https://github.com/bathcat/right-mug`
- git clone `https://github.com/bathcat/gymnast`
- git clone `https://github.com/dictcp/awesome-git-my-awesome-git`



Things to Notice

- Directory: `.git`
- You can see the status

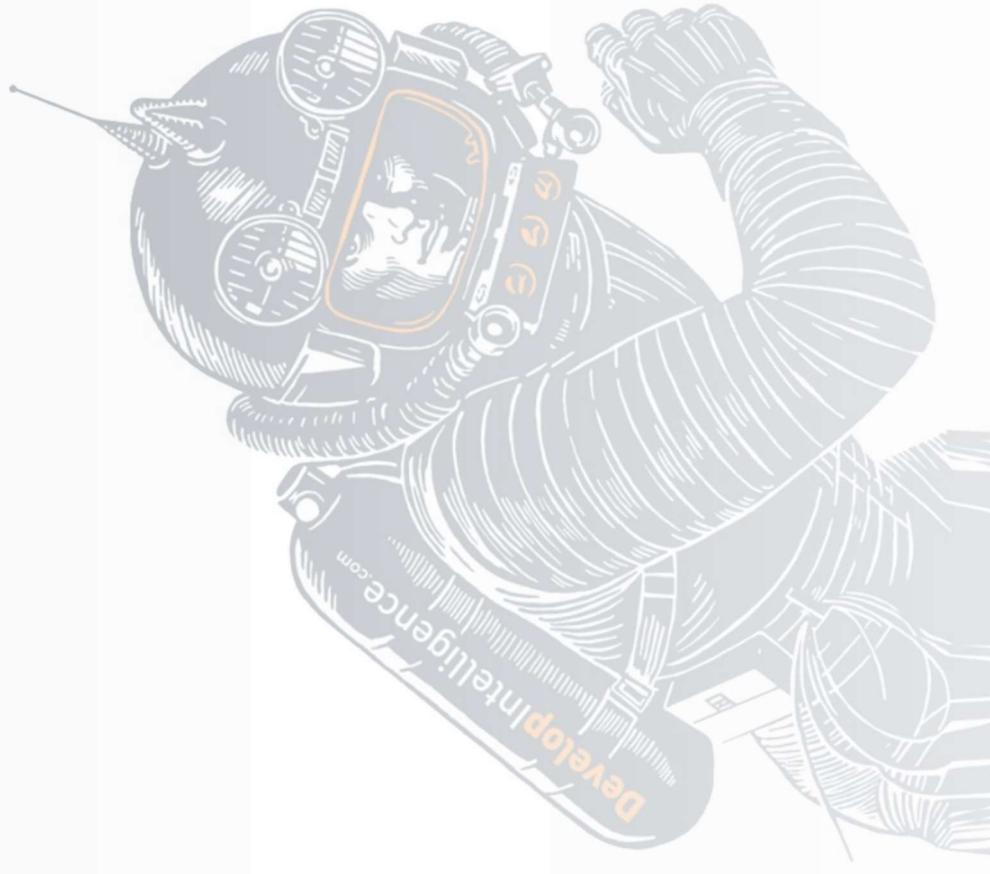




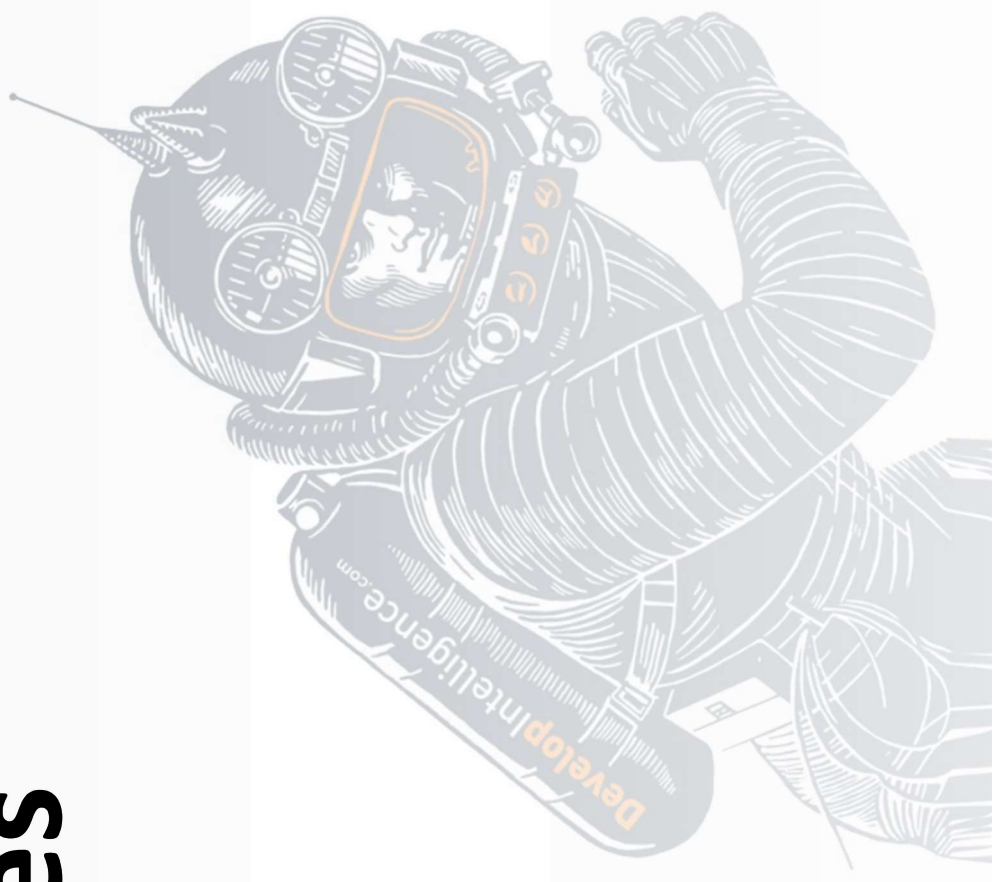
Key Commands

- Bring latest changes from the server into your working Branches

```
1 | git pull
```



Local Repositories





Workspaces

- Decide on a spot where you'll keep source code
- Examples:
 - `c:/workspace`
 - `~/workspace`
 - `~/Documents/Repos`
- Consider a specific location for 3rd-party repositories you're looking at, not working on:
 - e.g. `~/workspace/throwaways`



- At some point, git asks you to do this:

```
1 git config --global user.name "Your Name"  
2 git config --global user.email "your_email@whatever.com"  
3 git config --global core.editor notepad++.exe
```

- Note: You may need the whole path of notepad++.exe



More Setup

- If you're working with non-windows developers, this may be important:

```
1 git config --global core.autocrlf true
2 git config --global core.safecrlf warn
```



New Project

- Make a new directory and put something in it:

```
1 mkdir mom-n-pop
2 cd mom-n-pop
3 '<h1>Hello World</h1>' >> index.html
```



Initialization

- To add version control to your project:

```
1 | git init
```

- Check the status:

```
1 | git status
```





Add to Staging

- To add your changes to the staging area:

```
1 | git add index.html
```

- Check the status:

```
1 | git status
```



Commit Changes

- To move your staged changes to the local repository:

```
1 | git commit -m'Initial commit.'
```

- Check the status:

```
1 | git status
```





Make an Update

- Edit index.html
- Then:

```
1 git status
2 git add .
3 git commit -m'Make HTML legal.'
```



- For all changes on your current branch:

```
1 | git log
```

- Or:

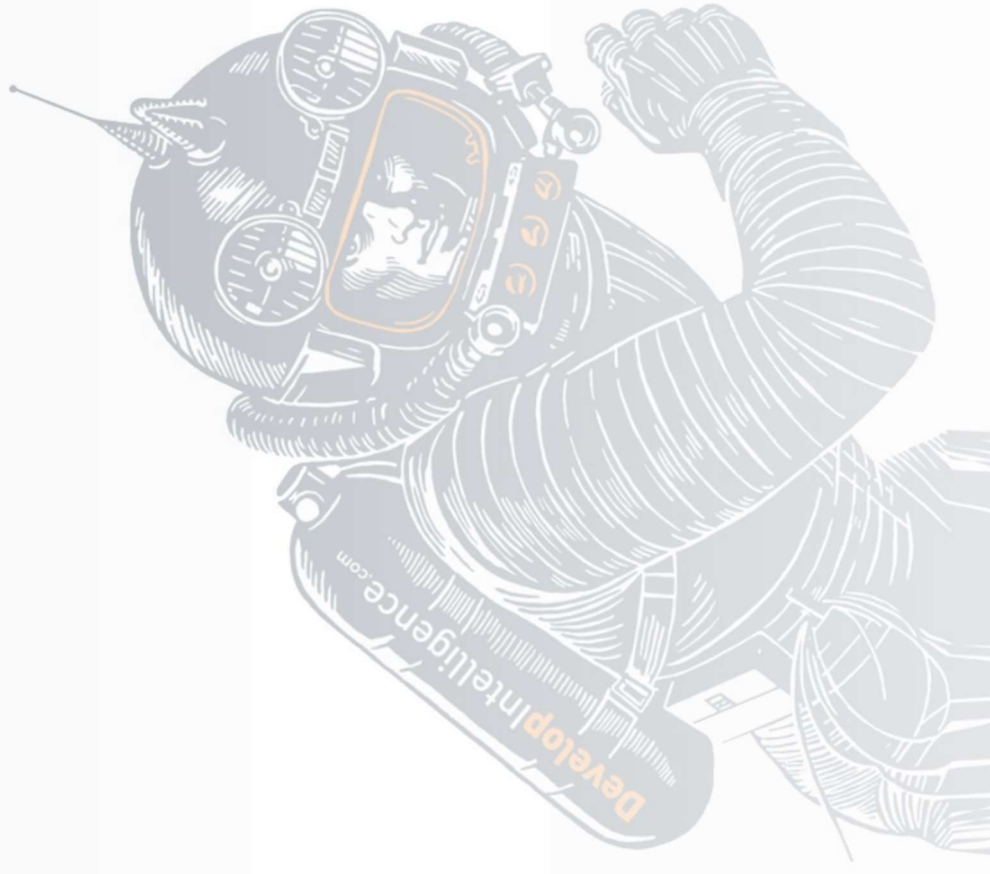
```
1 | git log --pretty=oneline
```



Things to Play With

- If you're inclined:
 - Aliases
 - Fancy History
- Next up: Branching and merging







Review

1. Define the term repository
2. Explain how to grab a public repo off github
3. Explain how to create a new repository locally

