Agenda

1. Git
2. Dependencies in Python projects
3. Flask
4. Demo
5. Flask task for the day

1. Git

1.1. Commonly used commands

In the next slide, I've included some mostly commonly used commands for Git. Do some research (alone or with friends) to find out what they mean and what they do.

Command	Purpose
`git init`	
`git clone <url>`	
`git add <file>`	
`git add .`	
`git reset`	
`git commit -m ""`	
`git status`	
`git push`	
`git pull`	

1.2. Wonderful article (36 min read)

Long read, but so worth it. Covers some beginner and intermediate concepts. You'll be fairly proficient at using Git if you understand most of the concepts in the article.

2. Dependencies in Python projects

2.1. If we use a package for our project (recall how to use a third-party package?), the package becomes a dependency to our project.

2.2. We keep track of a list of dependencies in `requirements.txt` .

2.3. Update `requirements.txt` by doing `pip freeze > requirements.txt`

3. Flask allows us to create dynamic web apps

But what does dynamic mean?

We did not prewrite any HTML code that says "Hello, David" or "Hello, Matt".
We also did not prewrite any routes that is `https://instagram.com/users/natgeo` .

4. Demo

5. Flask task

5.1. Set up a new Conda environment

> Install Flask and save it in your `requirements.txt`

5.2. Create a new local Git repo and link to your GitHub

> Make sure you remember to commit every step of the way!

5.3. Start a new Flask app

It should contain the following:

1. At least **3 routes** (and **3 corresponding HTML pages**)
2. Make use of `_layout.html`
3. At least **2 images** and **1 CSS file**
4. Use `if-else` at least once in Jinja, i.e. in one of your HTMLfiles
5. Extra challenge: use a `for-loop` in Jinja