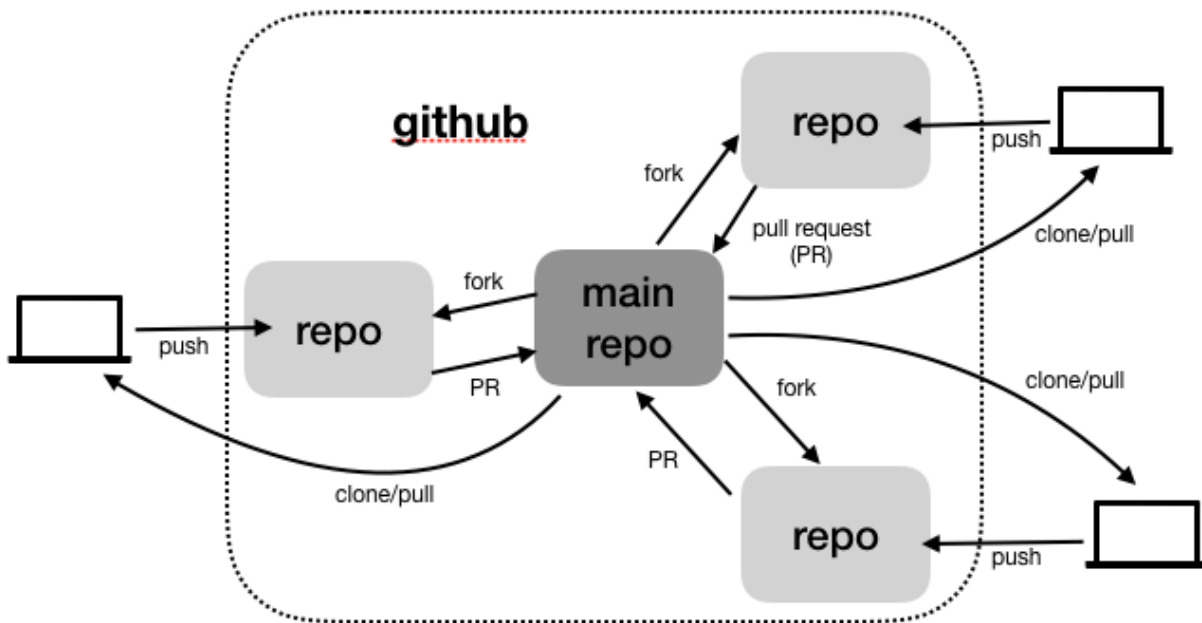


Code Review

Pull Requests

- The pull request is git's primary mechanism for getting feedback before integrating changes into the main repo
- Nobody pushes directly to the main repo
- Everybody has a "fork" of the main repo
- Workflow: (1) fork main repo to own repo on github (2) clone main repo to laptop, (3) make changes, (4) push changes to fork, (5) make pull request, suggesting that changes in the fork are merged back to the original



Remotes

- Note that we're pushing and pulling to/from different places!
- A "remote" is a name for one of these places.
- When you first clone, you automatically get a remote called "origin" referring to the main repo
- Add your own for your forked repo:
git remote add tyler git@github.com:tylerharter/demo.git
- View the remotes: **git remote -v**
- (1) git pull origin master, (2) changes, (3) git push tyler master, (4) PR

Animation

We'll sometimes want animations to visualize data (e.g., imagine an animated map of Madison showing traffic incidents over time).

Many Python libraries are based on ffmpeg; install it here: <https://ffmpeg.org/>. Make sure the binary program is in the PATH (i.e., you can execute `ffmpeg` in the terminal). `ffmpeg` can take many images, use them as frames, and stitch them together into a video clip.

```
In [1]: 1 %matplotlib inline
```

```
In [2]: 1 import pandas as pd
2 import geopandas # we'll use this to read shapefiles
3 from shapely.geometry import Polygon, Point
4 from matplotlib import pyplot as plt
5 from matplotlib import animation
6 from IPython.display import HTML
```

```
In [3]: 1 %%capture
2 fig, ax = plt.subplots()
```

```
In [4]: 1 city = geopandas.read_file("city")
2 lakes = geopandas.read_file("lakes")
3
4 def update_func(frame_num):
5     # each time this gets called, we compute new coordinates based on frame_num
6     points = pd.DataFrame([
7         {"loc": Point(-89.406749 + frame_num/100, 43.071478)},
8         {"loc": Point(-89.384054, 43.074617 + frame_num/100)},
9     ])
10    points = geopandas.GeoDataFrame(points, geometry="loc")
11
12    ax.cla() # clears plot area (try removing it!)
13    city.plot(color="lightgray", ax=ax)
14    lakes.plot(color="darkgray", ax=ax)
15    points.plot(color="black", marker="x", markersize=200, ax=ax)
16    ax.set_axis_off()
17
18    if frame_num > 10:
19        ax.text(-89.384054, 43.074617, "bye!")
20
21    HTML(animation.FuncAnimation(fig, update_func, frames=15, interval=200).to_html5_video())
```

- `%%capture` suppresses output for that cell
- **FuncAnimation** in the **animation** submodule of **matplotlib** calls the give update function (**update_func**) 15 times (**frames** parameter), once ever 200 milliseconds (**interval** parameter).
- **.to_html5_video()** converts the video to HTML code that you could copy/paste into a site
- **HTML(...)** embeds that returned HTML directly into the notebook