Web Programming
with Python and JavaScript
Git
HTML, CSS
SQL
JavaScript
Front Ends
Django
Testing, CI/CD
Scalability
Security
• Project 0
• Project 1
• Project 2
• Project 3
• Final Project
Version Control
git
Keep track of changes to code.

Create file

\begin{tabular}{l}
  a & 1 \\
  b & 2 \\
  c & 3 \\
\end{tabular}
Keep track of changes to code.

Create file

```
a = 1
b = 2
c = 3
```

Add a line

```
a = 1
b = 2
c = 3
d = 4
```
Keep track of changes to code.

Create file

Add a line

Remove a line
Synchronizes code between different people.
Synchronizes code between different people.
Synchronizes code between different people.
Synchronizes code between different people.
Synchronizes code between different people.
Test changes to code without losing the original.
Test changes to code without losing the original.

```
a = 1
b = 2
c = 3
```

```
a = 1
b = 2
c = 3
d = 4
```
Test changes to code without losing the original.
Revert back to old versions of code.

- Create file: $a = 1$
  $b = 2$
  $c = 3$

- Add a line: $a = 1$
  $b = 2$
  $c = 3$
  $d = 4$

- Remove a line: $a = 1$
  $c = 3$
  $d = 4$
Revert back to old versions of code.

Create file

Add a line
git clone
git clone <url>

```plaintext
da = 1  
b = 2  
c = 3  
d = 4
```
git clone <url>

a = 1
b = 2
c = 3
d = 4
git clone <url>

a = 1
b = 2
c = 3
d = 4

git clone <url>
git add
git add <filename>

```c
int b = 2;
int c = 3;
int d = 4;
int a = 1;
```
`git add <filename>`

```
int b = 2;
int c = 3;
int d = 4;
int a = 1;
```

```
{a = 1
 b = 2
 c = 3
 d = 4
 e = 5
}  
```
git add <filename>

```python
a = 1
b = 2
c = 3
d = 4
e = 5
```

```c
int a = 1;
int b = 2;
int c = 3;
int d = 4;
```
git add <filename>

Changes to be committed:
modified: foo.py

```python
a = 1
b = 2
c = 3
d = 4
```

```c
int a = 1;
int b = 2;
int c = 3;
int d = 4;
```
git commit
git commit -m "message"
git commit -m "message"

int a = 1;
int b = 2;
int c = 3;
int d = 4;

int a = 1;
int b = 2;
int c = 3;
int d = 4;

int a = 1;
int b = 2;
int c = 3;
int d = 4;

int a = 1;
int b = 2;
int c = 3;
int d = 4;
int e = 5;

git commit -m "Add line"
git commit -m "message"

Add line

```
a = 1
b = 2
c = 3
d = 4
```

Add line

```
a = 1
b = 2
c = 3
d = 4
e = 5
```
git status
git status

a = 1
b = 2
c = 3
d = 4

Add line

a = 1
b = 2
c = 3
d = 4
e = 5
git status

Add line

a = 1
b = 2
c = 3
d = 4
e = 5

a = 1
b = 2
c = 3
d = 4
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
(use "git push" to publish your local commits)
git push
git push

Add line

a = 1
b = 2
c = 3
d = 4
e = 5
git push

Add line

```plaintext
a = 1
b = 2
c = 3
d = 4
e = 5
```
git push

Add line

Add line
git pull
git pull

```
int a = 1;
int b = 2;
int c = 3;
int d = 4;
int e = 5;
```
git pull

Add line

Remove line

a = 1
b = 2
c = 3
d = 4
e = 5

a = 1
b = 2
c = 3
d = 4
e = 5

a = 1
b = 2
c = 3
d = 4

a = 1
b = 2
c = 3
d = 4
e = 5

git pull

Add line
git pull

a = 1
b = 2
c = 3
d = 4
e = 5

Add line

Remove line
Merge Conflicts
Merge Conflicts
Merge Conflicts

git pull
Merge Conflicts

git pull

CONFLICT (content): Merge conflict in foo.py
Automatic merge failed; fix conflicts and then commit the result.
Merge Conflicts

```plaintext
a = 1
<<<<<< HEAD
b = 2
=====
b = 0
>>>>>> 57656c636f6d6520746f20576562
c = 3
d = 4
e = 5
```
Merge Conflicts

```
git pull

your changes
```

```
remote changes
```

```
<<<<<<< HEAD
{ b = 2
 =====
{ b = 0
 >>>>> 57656c636f6d6520746f20576562
 c = 3
d = 4
e = 5
```
Merge Conflicts

git pull

d = 4

e = 5

a = 1
<<<<<< HEAD
b = 2
=====
b = 0
>>>>> 57656c636f6d6520746f20576562
c = 3
Merge Conflicts

```
git pull
```

```
a = 1
b = 2
c = 3
d = 4
e = 5
```
Merge Conflicts

git pull

a = 1
b = 2
c = 3
d = 4
e = 5
git log
git log
git log
commit 436f6d6d6974204d73672048657265
Author: Brian Yu <brian@cs.harvard.edu>
Date: Mon Jan 22 14:06:28 2018 -0400

Remove a line

commit 57656c636f6d6520746f20576562
Author: Brian Yu <brian@cs.harvard.edu>
Date: Mon Jan 22 14:05:28 2018 -0400

Add a line
git reset
**git reset**

- `git reset --hard <commit>`
- `git reset --hard origin/master`
git reset

- git reset --hard <commit>
- git reset --hard origin/master
git reset

- `git reset --hard <commit>`
- `git reset --hard origin/master`
HTML
A First Webpage

<!DOCTYPE html>
<html>
  <head>
    <title>Hello!</title>
  </head>
  <body>
    Hello, world!
  </body>
</html>
Common HTML Tags

- `<h1>`, `<h2>`, ..., `<h6>`
- `<ol>`, `<ul>`
- `<img>`
- `<a>`
- `<table>`
- `<form>`
<!DOCTYPE html>
<html>
  <head>
    <title>Hello!</title>
  </head>
  <body>
    Hello, world!
  </body>
</html>
CSS
Common CSS Properties

- color
- text-align
- width, height
- margin, padding
- font-family, font-size, font-weight
- border
Non-Semantic Elements and Attributes

- div
- span
- id
- class
GitHub Pages