HTML

Basics & GitHub

While you're waiting, set up your GitHub account if you don't already have one.

HTML Basics

- We'll be learning HTML5 and CSS3
- These are new versions - not currently supported fully across all browsers
- As we learn, we will need to learn to address browser support issues.
- This is a major issue in web development

Template

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Title Goes Here</title>
</head>
<body>Page body goes here...</body>
</html>
```

The DOCTYPE

The Document Type Definition is a special tag.

It tells anyone or anything reading the file what the file type is.

For HTML 5 it must look exactly like this:

```
<!DOCTYPE html>
```

Older DOCTYPE tags

The HTML 4.1 “Strict” DOCTYPE:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

The HTML 4.1 “Transitional” DOCTYPE:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN" "http://www.w3.org/TR/html4/loose.
dtd">
```

This is just to illustrate that they are different…

The <html> Tag

The <html> tags enclose the entire document

```
<html lang="en">
   <head>... stuff removed ...</head>
   <body>... stuff removed ...</body>
</html>
```

It always contains the <head> and <body>

Attributes

The <html> tag contains an attribute that specifies the "spoken" language of the document.

```
<html lang="en">
```

Attributes are always in this format, with the value in quotes.

Parents and Children

- HTML is a structured, hierarchical language
- Tags often (not always) contain other tags or text
- This forms a parent/child relationship

```
<parent>
  <child>
    <grandchild>Some text</grandchild>
  </child>
  <sibling>More text</sibling>
</parent>
```

<html>'s Children

- head - contains stuff "about" the page
- body - contains stuff you see on the page

```
<html lang="en">
   <head>... stuff removed ...</head>
   <body>... stuff removed ...</body>
</html>
```

Important <head> tags

```
<head>
    <meta charset="utf-8">
    <title>Title Goes Here</title>
</head>
```

- meta - there are many possible meta tags using different attributes.
- There should always be a meta charset tag.
- title - the browser window or tab title

The <body>

- The children you add to body determine what the web page looks like.
- There are many possibilities.
- We will learn new tags all semester.
  - We will learn a few tonight
  - Many more will be in the reading for Ch 2
  - More will come throughout the class

Some starter tags...

Paragraph text:

```
<p>Some paragraph text goes here.</p>
```

Headings levels 1 - 6:

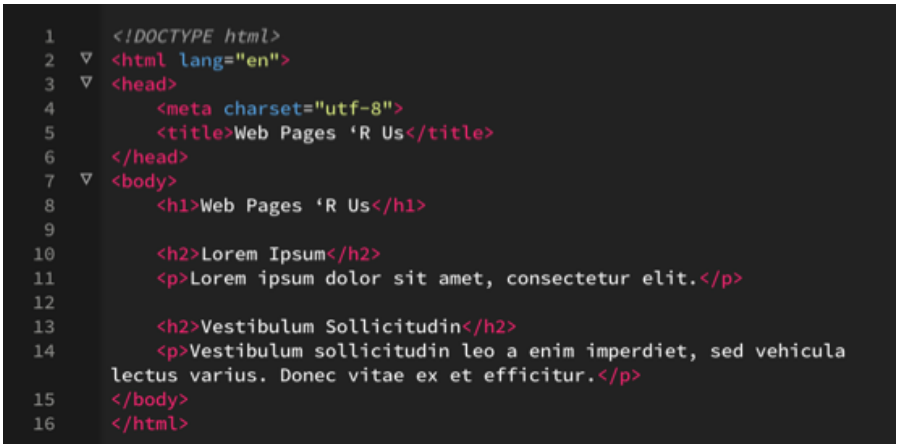
```
<h1>First Level Heading</h1>
<h2>Second Level Heading</h2>
```

Put it all together

```
<body>
    <h1>Web Pages 'R Us</h1>

    <h2>Lorem Ipsum</h2>
    <p>Lorem ipsum dolor sit amet, consectetur elit.</p>

    <h2>Vestibulum Sollicitudin</h2>
    <p>Vestibulum sollicitudin leo a enim imperdiet, sed
vehicula lectus varius. Donec vitae ex et efficitur.</p>
</body>
```



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Web Pages 'R Us</title>
</head>
<body>
    <h1>Web Pages 'R Us</h1>

    <h2>Lorem Ipsum</h2>
    <p>Lorem ipsum dolor sit amet, consectetur elit.</p>

    <h2>Vestibulum Sollicitudin</h2>
    <p>Vestibulum sollicitudin leo a enim imperdiet, sed vehicula
lectus varius. Donec vitae ex et efficitur.</p>
</body>
</html>
```

Really putting it all together...

GitHub

As I mentioned, we'll be using GitHub for the assignments.

We're going to work through the basic workflow to get you into our "Student Directory".

Get an account

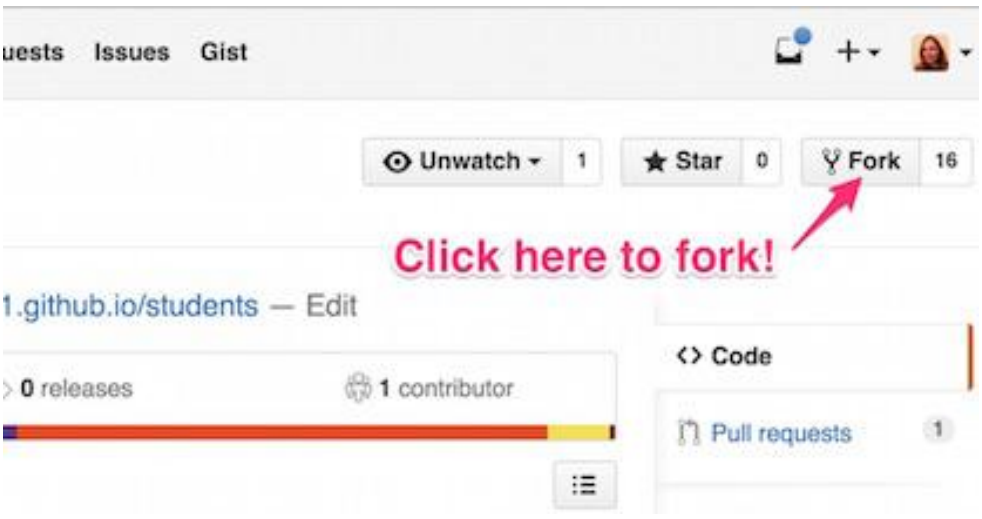
- If you don't already have an account, register now!
- Use a nice professional user name & put your actual name in the account profile
- Get a free micro account (5 private repos), by registering for a GitHub Student Pack

Go to our Org Page

https://github.com/htc-ccis1301

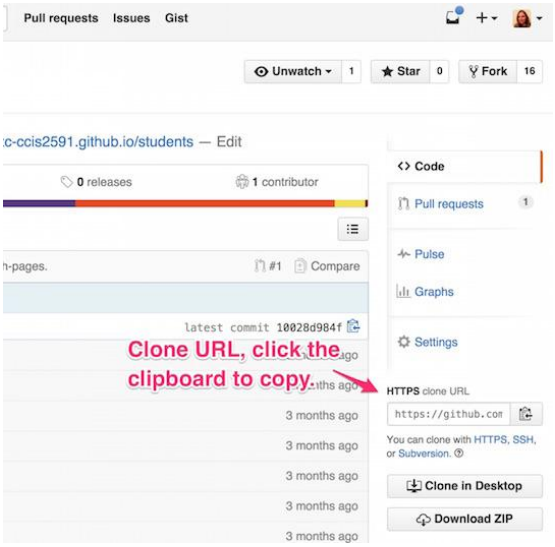
- Look for the students repository
- Click on the link to view the files
- Click fork button to make a personal copy

Fork the students repo



Clone to get local

- Copy the URL and open the Git Shell on your computer.
- Type in “git clone”, a space, and paste
- Make sure the url has your user id, not the htc-ccis org name



Configure User Info

git config --global user.name “UserName”
git config --global user.email your@email.com

Make changes

To add yourself to the course directory, add a new USERNAME.yml file under the appropriate _data/TERM_YEAR/ directory.

Make sure to name it your GitHub user ID.

Copy the example file mbmosman.yml, and replace the values with your own name, emoji, and introduction.

Check Status

The “git status” command will show you changes between your local repository and GitHub.

You should see that you have an untracked file. Use “git add” to add it to the repository.

git add . - will add all files

git add <file-name-here> - will add one file

Commit changes

- Check “git status” again and you’ll see you have one new file.
- The “git commit” command will version your work. This gives you something to come back to later.

git commit -m “Your message here”

Push changes

The “git push” command will send your changes to GitHub.

This expects your local & remote branches to be the same. If not you need to add it.

git push

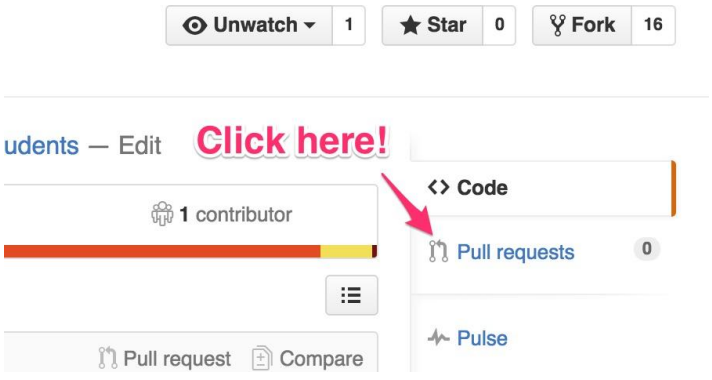
git push origin <remote-branch>

Verify on GitHub

You should now see your files on your copy of the students repository.

Pull Request

The pull request is how changes to your personal repository can be sent back to the class repository.



That's It!

- Once the pull request is in, I can merge all of our separate changes together.
- This is how real-world teamwork happens.
- For most class assignments I will not merge in the changes, just comment on and close the requests.