Class 00

Hola. These are the class notes I've typed up for everyone that did not make it to the first class. If you have any questions, don't hesitate to ask on Slack or send me an email.

Syllabus

Read over the syllabus (which is posted on Blackboard)

With regards to attendance, if you missed the first class – don't worry about it. The registrar issue along with students on the waitlist caused a lot of confusion.

I hope everyone has a mac (laptop) to bring to class. If you have a PC, let me know ASAP so I can plan for both systems. These notes will be slightly harder to follow if you have a PC due to differences between the 'Terminal' and the 'Command Prompt'

Slack

Click on the Slack link to join the Slack group:
https://join.slack.com/t/idesn3535-01/shared_invite/MjM3NTYzMjIyODM0LTE1MDQ3ODY4NTQtMGU2MTJmM2I5OA

Once there, please go to the #class0 channel – there is a link in #announcements. We'll have a separate channel for each class with a list of resources with links.
(It's class0 and not class1 because this first class was a surprise and I didn't feel like changing anything!)

Spirit Animal / Favorite Animal / Patronus

Choose your spirit animal/favorite animal/patronus.
Head over to The Noun Project, register for an account, and find an icon of your animal to download the png for and add it as your Slack profile picture.

Your Code Editor

Choose a code editor and install it. I suggest Sublime, but choose whatever works for you.
Sublime - https://www.sublimetext.com/3
Atom - https://atom.io/

Class Folder

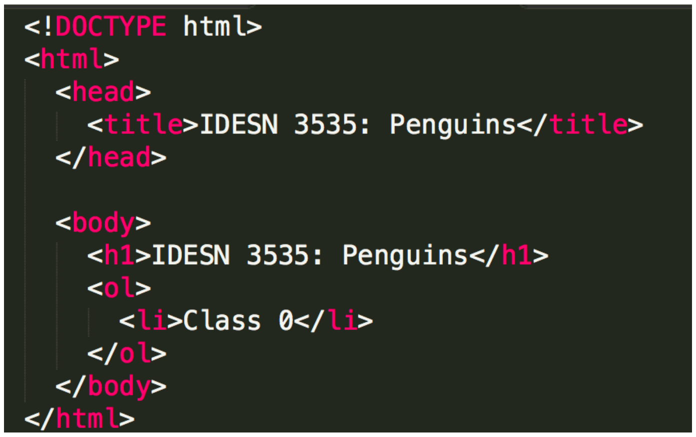
Create a folder somewhere on your computer that you will use throughout the class. I suggest you name it something similar to idesn3535-penguin (replace 'penguin' with your animal). I also suggest you do not use spaces or capital letters.

Open the folder in your text editor.

index.html

Create a file called index.html. This will be our index for every project/homework/etc in the class. Go ahead and find index.html in your find and open it in your browser (I suggest Chrome for this class, but again, choose what you please – except for ie… that'd be painful to debug). All you should see right now is a blank page.

Let's change that by filling up index with the following code. If you do not understand what any of this does, please review your notes from the prerequisite class, search on google or stack overflow (the most helpful resources in this class), or reach out to me via Slack or email. It is important that you understand what the code is doing.



```
<!DOCTYPE html>
<html>
  <head>
    <title>IDESN 3535: Penguins</title>
  </head>

  <body>
    <h1>IDESN 3535: Penguins</h1>
    <ol>
      <li>Class 0</li>
    </ol>
  </body>
</html>
```

You can go back to your browser, refresh the page, and see your changes.. (Please replace 'penguin' with your animal)

GitHub (and git)

Head on over to https://github.com/ and create an account if you don't already have one. Then create a new repository and name it "idesn3535-penguin" but replace "penguin" with your animal. Do not add a README (sorry for everyone in class who I had check this box). We'll come back to this later.

The Terminal

Open the 'Terminal' app on your computer. An easy way to do this is to click the little spotlight icon in the top right and type 'Terminal' and open it up. (I think the shortcut for this is command+space)

We need to change directory in the folder we created earlier. To do that, we use the command 'cd'. The Terminal opens in your user's directory, so it should be pretty easy to determine where you need to go. Another helpful command is 'ls' to list the files in the current directory.

If your folder is located on your Desktop, the example command would be something like:

```
cd Desktop/idesn3535-penguin
```

Capital letters and spaces do matter, if you have a space in a folder name, you need to put a backslash in front of the space, like so:

```
cd Desktop/Fall\ 2017/idesn3535-penguin
```

When you get to your folder, type 'ls' and you should see your index.html file listed.

Git

The next step is to ensure that you have git installed. Git is a type of version control which will allow us to backup our code, track changes, and see every version of our site. It is also helpful when multiple people are working on the same code to help resolve conflicts and not overwrite each others changes. To check if you have git installed, please type:

```
git --version
```

You should see something that looks like:

```
git version 2.11.0 (Apple Git-81)
```

(potentially with a different version)

If git is not installed, you will be prompted to install it. If you ever are asked for a password in the Terminal, this is the password you use to log into your computer. Note: as you type a password in the terminal, it will not visually show up. Type it out, press enter, and you should be good to go.

When finished installing git, type `git --version` once again to ensure that it is installed.

git init

The next step is to initialize our git repository in our folder. In Terminal, type `git init`. This will setup git in your project.

Whenever you are unsure of the state of your files, type `git status` and you will a list of new/modified/staged files. Right now we just have one new file of index.html. If we type `git add index.html` and then run `git status` again, we should see that our index.html file is now 'staged'. This means that it is ready to be committed (a fancy way of saving). (Note: you can also type `git add .` to add all new/changed files to staging)

To commit you changes, type:

```
git commit -m "Initial Commit"
```

The message in "quotes" is your commit message. This should describe the changes that you are making to help understand why you made some changes when you look back at your code.

If you leave out the commit message, you will be entered into a program call vim which we will not cover in this class. If that happens to you, you will need to exit vim and commit again with the commit message. You'll notice if you get to this point. To exit vim:

- press 'Escape'
- Hold 'Shift' and press 'Q'
- Type 'x' and press 'Enter'
- The commit will be aborted.

Now we need to push our changes up to github.com. To do this, go back to github.com and copy the url for your repository. You may need to click a link that says ‘use https’ as the ‘ssh’ version will not work with our current setup. Your link should look something like this:

https://github.com/bvandorn/idesn3535-penguin.git

but with your username in place of ‘bvandorn’ and your animal in place of ‘penguin’

Now pop back over to the Terminal and type:

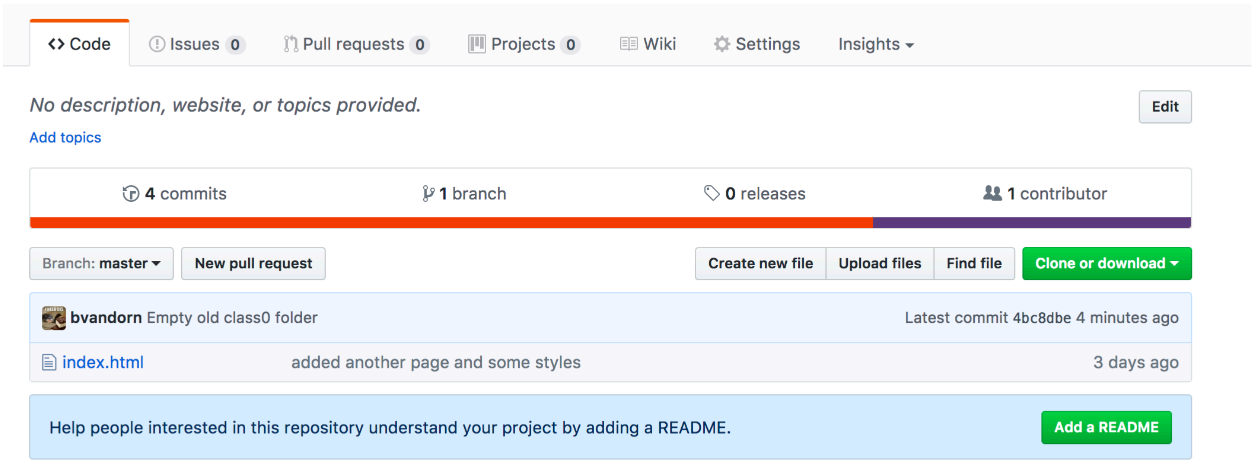
```
git remote add origin https://github.com/bvandorn/idesn3535-penguin.git
```

But replace that url with your url. This tells our computer where to store the changes we have committed.

Once that is complete, type

```
git push -u origin master
```

To push your changes. You should be able to go back to github.com, refresh, and you will see your index.html file listed similar to this:



No description, website, or topics provided.

4 commits | 1 branch | 0 releases | 1 contributor

bvandorn Empty old class0 folder — Latest commit 4bc8dbe 4 minutes ago

index.html — added another page and some styles — 3 days ago

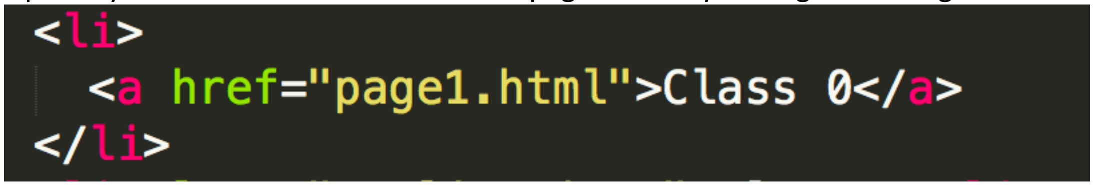
Help people interested in this repository understand your project by adding a README.

If you have trouble here, or would like to know more, please read through this blog post for a helpful introduction to git: http://product.hubspot.com/blog/git-and-github-tutorial-for-beginners

Site Changes

The next step is to make some changes to our site. We’re going to create a new file called ‘page1.html’ and add a link to that file.. You can put whatever you want in page1.html, but it should be of a similar structure of index.html.

Update your <li> in index.html to link to page1.html by adding an <a> tag in index.html.



```
<li>
  <a href="page1.html">Class 0</a>
</li>
```

You should be able to refresh your browser and see a new link for ‘Class 0’ – clicking this should bring you to page1.

The next step is to commit our changes.

Type `git status` to see the state of your files.

Add your index.html changes and your new page1.html by typing `git add .`

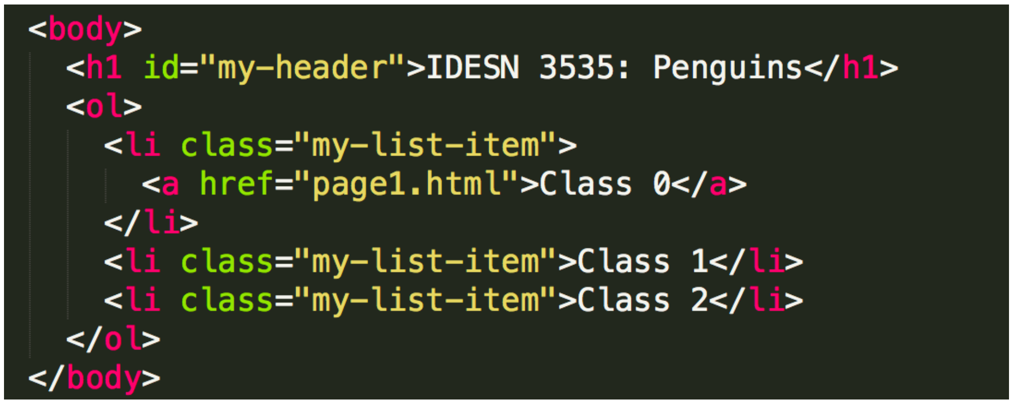
Type `git status` again too see that your files have been added to the index

Type `git commit -m "Add page1 and link to it"` to commit your files with a helpful message.

Type `git push` to push your files to github.com. Navigating to github.com will now show your page1.html along with your index.html.

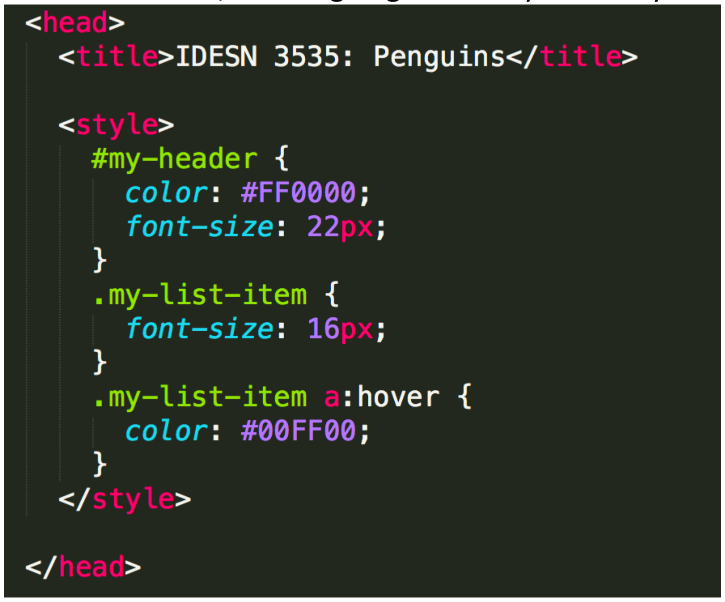
Adding Styles

The next step is to add some styles to our page. We will add an id to our <h1> and classes to each of our <li> entities. This should be a review.



```html
<body>
  <h1 id="my-header">IDESN 3535: Penguins</h1>
  <ol>
    <li class="my-list-item">
      <a href="page1.html">Class 0</a>
    </li>
    <li class="my-list-item">Class 1</li>
    <li class="my-list-item">Class 2</li>
  </ol>
</body>
```

Now in the head, we are going to add styles in <style> tags.



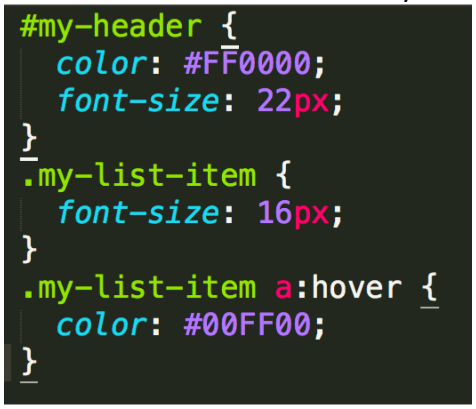
```html
<head>
  <title>IDESN 3535: Penguins</title>

  <style>
    #my-header {
      color: #FF0000;
      font-size: 22px;
    }
    .my-list-item {
      font-size: 16px;
    }
    .my-list-item a:hover {
      color: #00FF00;
    }
  </style>

</head>
```

This will give us bigger text, a red header, and turn our link green when we hover over it when refreshing our page. If any of this is confusing, please review your previous classwork, search google, or ask questions.

Now commit these changes and push them to github.

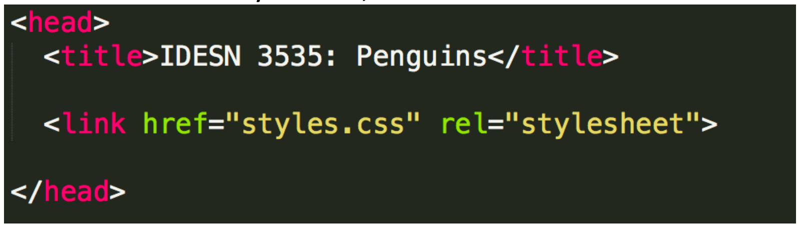
Sharing Styles and Adding Fonts

We want these styles on both index.html and page1.html without having to define them all twice. Create a new file called 'styles.css' and copy everything in your <style> tags into that file. This is called a stylesheet. Note that we do NOT need <style> tags in a stylesheet.



```
#my-header {
  color: #FF0000;
  font-size: 22px;
}
.my-list-item {
  font-size: 16px;
}
.my-list-item a:hover {
  color: #00FF00;
}
```

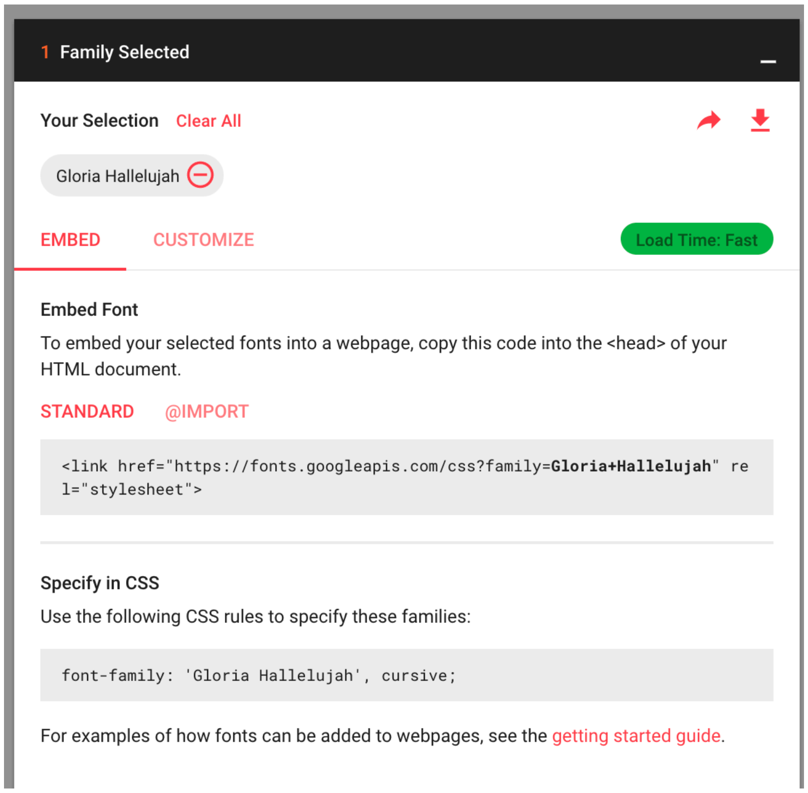
To reference our stylesheet, we need to create a link to it in the <head> of our index.html.. Like so:



```
<head>
  <title>IDESN 3535: Penguins</title>

  <link href="styles.css" rel="stylesheet">

</head>
```

Refresh and everything should still be working. If you add this link in the <head> tag of page1.html, that page will be able to share the same stylesheet.

Now we want to add a different font to our page. Go to Google Fonts and choose whatever font you want (as long as it's English and readable – also don't pick Gloria Hallelujah, that's my example ☺) – I want you to make this your own. Once you've selected your font, click the (+) button and open the menu that appears at the bottom of the screen.



Embed Font

To embed your selected fonts into a webpage, copy this code into the <head> of your HTML document.

```
<link href="https://fonts.googleapis.com/css?family=Gloria+Hallelujah" re
l="stylesheet">
```

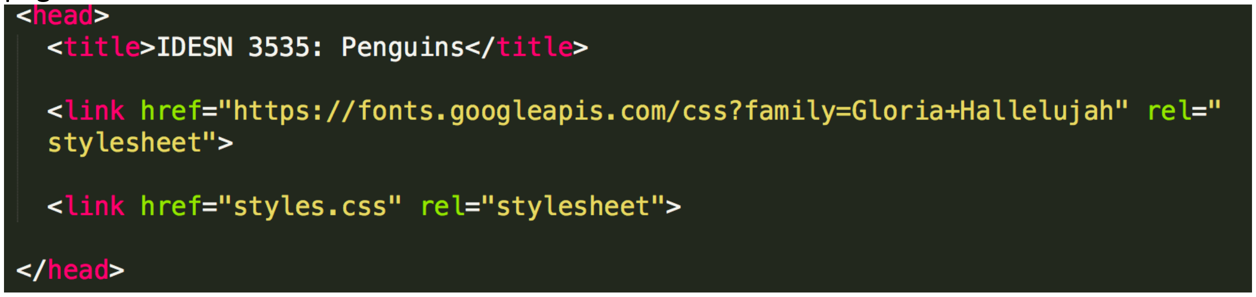
Specify in CSS

Use the following CSS rules to specify these families:

```
font-family: 'Gloria Hallelujah', cursive;
```

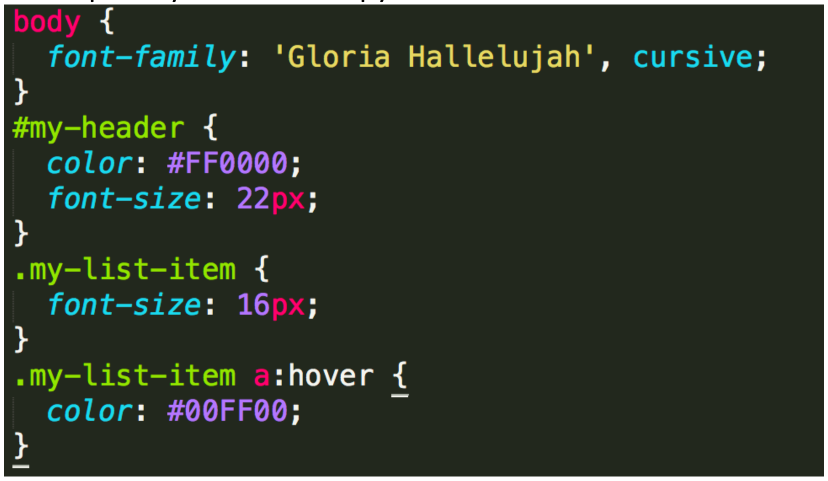
For examples of how fonts can be added to webpages, see the getting started guide.

Copy the <link …. > code that you are presented with and stick that into the head of both index.html and page1.html.



```
<head>
  <title>IDESN 3535: Penguins</title>

  <link href="https://fonts.googleapis.com/css?family=Gloria+Hallelujah" rel="
  stylesheet">

  <link href="styles.css" rel="stylesheet">

</head>
```

Now open styles.css and copy the CSS that is below that link in the menu and place it into the body selector.



```
body {
  font-family: 'Gloria Hallelujah', cursive;
}
#my-header {
  color: #FF0000;
  font-size: 22px;
}
.my-list-item {
  font-size: 16px;
}
.my-list-item a:hover {
  color: #00FF00;
}
```

You should be able to save all your files, refresh, and see your font!

Commit your changes and push them to github.

Node and NPM

First, open Terminal to check if you have node and npm installed:

`npm -v` you should see something similar to 5.3.0 (don't worry about the actual version)

`node -v` again, don't worry too much about the version, just worry if it says you don't have it installed.

If you don't have either of them installed, head over to https://nodejs.org/en/ and download the v6.11.3 LTS and install it. Once installed, you should be able to run those commands again without any issues.

Surge

Now we are going to host your site on a public URL. Head on over to http://surge.sh/ and follow the instructions to install Surge.

NOTE: Instead of running the 'surge' command by itself, you can run it with:

`surge -d idesn3535-penguin.surge.sh` (replace penguin with your animal)

The first time you run it, you will be prompted to make an account. Make one and continue. When it asks for a project path, just press enter to continue. If you did not use the -d flag in the note above, it will fill in a domain for use. Please use the format above (idesn3535-penguin.surge.sh) replacing ‘penguin’ with your animal. Press enter and your site will be hosted. You can navigate to the link it presents you with and you should see your site.

Every time you make changes you can run the surge command again to update your site.

Submission

To finish up, send me (bvandorn) a direct message on Slack with two urls:

- The URL for your hosted website (of the format idesn3535-penguin.surge.sh)
- Your github url (copy and paste what is in the address bar on github.com. Similar to this https://github.com/bvandorn/idesn3535-penguin)

Homework

For homework, please brush up on your html/css skills and update your index.html page to customize it and make it unique to you. It doesn’t need to look like a list, it just needs a title and links to other pages that we’ll use later. Feel free to do whatever you want. Spend 30-60 minutes on this and ask any questions you may have on Slack or via email. Upload your changes and we’ll have a mini-critique next class.

Here are some links to help:
http://www.cssbasics.com/
https://www.w3schools.com/tags/default.asp
https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML