CSS and CS Principles

A Lesson in Which We Learn Basic CSS by Building on Our Prior Knowledge of Computer Science Principles!

Methods 2
Hunter College Advanced Certificate Program
Alex Moore
5-9-22

Materials:

- Slide Deck: https://docs.google.com/presentation/d/1RfChThEZNHYiWdSLTkWYutgsRRlG8xTApU9y0YtG5v0/edit?usp=sharing
- Shared repl.it with four HTML files, a CSS file, and a JavaScript file: *https://replit.com/@alexmoore77/DemoHTML#index.html*

Do Now

In three minutes in our small class chat, describe a time when you had to do something inefficiently.
Next, share out!

Activity #1

Assuming prior knowledge of basic HTML, which we have learned prior to this lesson, do the following in ten minutes:

1) *Go to this URL: https://replit.com/@alexmoore77/DemoHTML#index.html*
2) *Fork the project so you can edit your own copy.*
3) *On each of the four HTML pages, change the font color of <h1> to green and the background color to yellow.*

Introduction to CSS

```
1)  body {
2)    background-color: lightblue;
3)  }
4)
5)  h1 {
6)    color: white;
7)    text-align: center;
8)  }
9)
10) p {
11)   font-family: verdana;
12)   font-size: 20px;
13) }
```

Source: w3Schools.com

Algorithmic Efficiency and Big O Notation

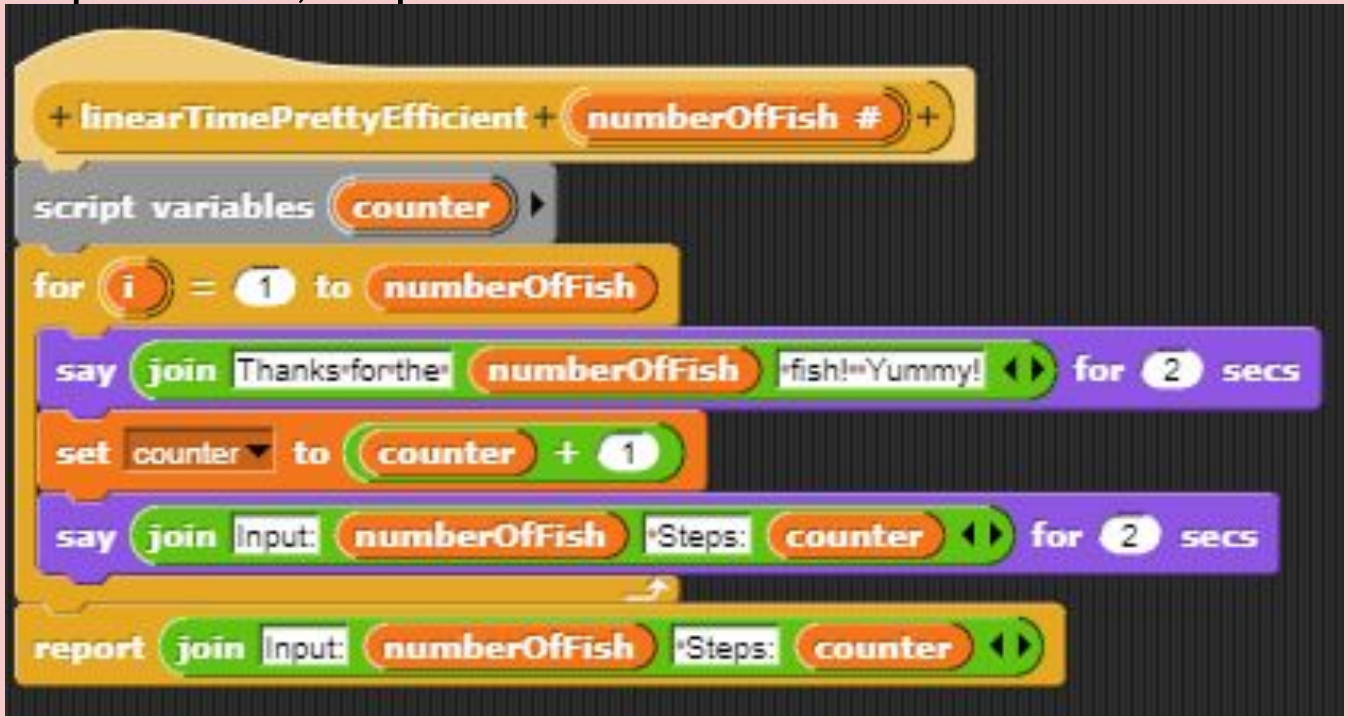
How can CSS be a more efficient way of handling styles than hard coding, and what should we know about algorithmic efficiency and Big O Notation?

Question 1: How can we measure the efficiency of algorithms in computer science?
Question 2: What is Big O Notation?
Question 3: If it takes us one minute to update each HTML file, and it takes one minute to update a CSS file, how can we describe the efficiency of using CSS to update styles instead of updating them in HTML?

Constant Time: O(1)
Input: 3, Steps: 1
Input: 10, Steps: 1
Input: 3000, Steps: 1

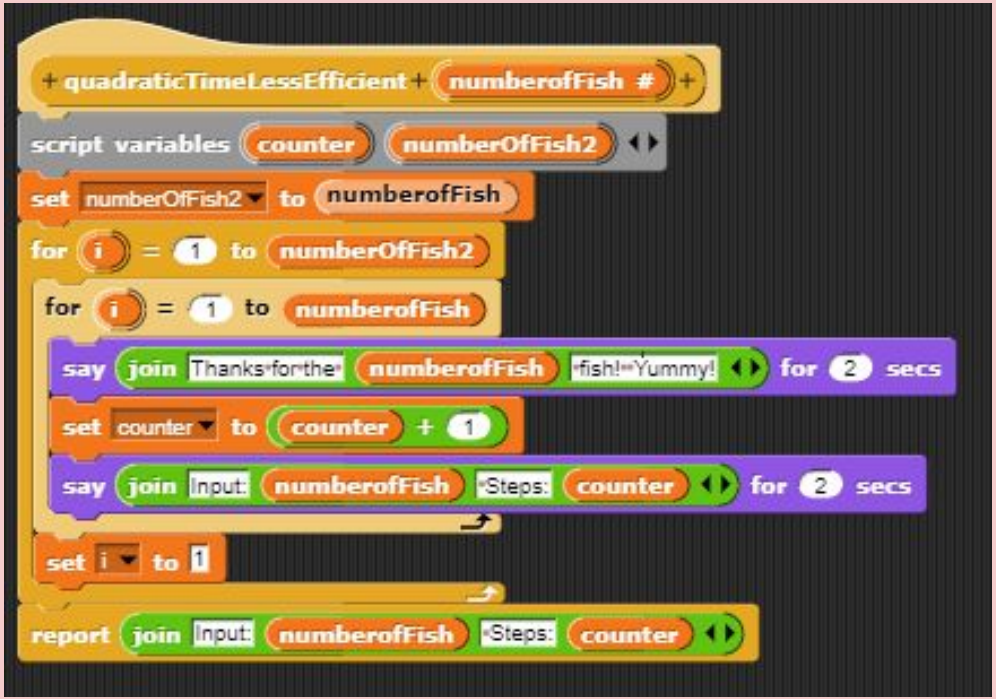
//Created by Mr. Moore (amoore@schools.nyc.gov)

Linear Time: O(n)
Input: 3, Steps: 3
Input: 10, Steps: 10
Input: 3000, Steps: 3000



//Created by Mr. Moore (amoore@schools.nyc.gov)

Quadratic Time: O(n^2)
Input: 3, Steps: 9
Input: 10, Steps: 100
Input: 3000, Steps: 9000000



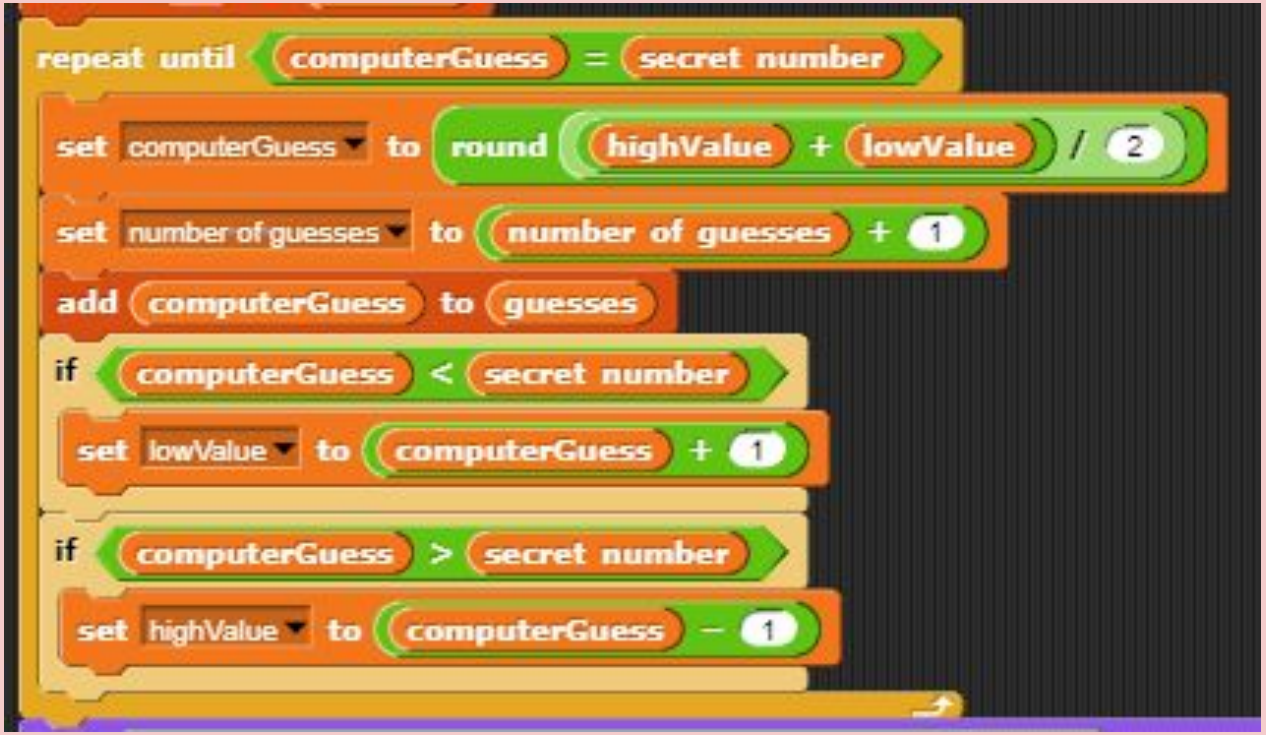
//Created by Mr. Moore (amoore@schools.nyc.gov)

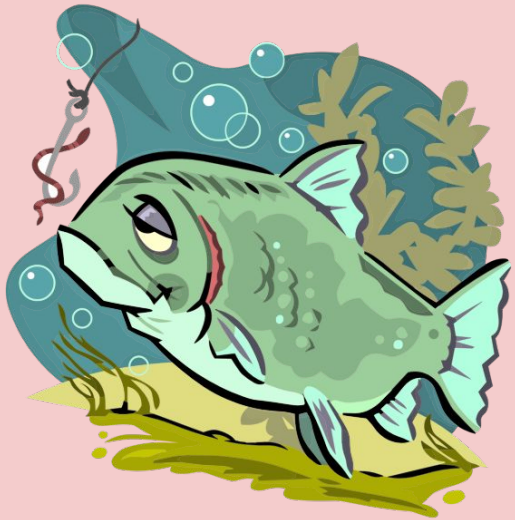
Linear Time: O(n)
Input: 3, Steps: 3
Input: 10, Steps: 10
Input: 3000, Steps: 3000

```
repeat until [computerGuess] = [secret number]
    set [computerGuess] to [computerGuess] + 1
    set [number of guesses] to [number of guesses] + 1
    add [computerGuess] to [guesses]
```

//Created by Mr. Moore (amoore@schools.nyc.gov)

Logarithmic Time: O(log n)
Input: 3, Steps: 2
Input: 10, Steps: 4
Input: 3000, Steps: ???





//Created by Mr. Moore (amoore@schools.nyc.gov)

Variable Scope

How can CSS set values for tags across all HTML files, and what should we know about variable scope?

Question 1: What is a local variable?
Question 2: What is a global variable?
Question 3: Is setting the styles for all HTML tags in a directory with CSS similar to setting global variables? Why or why not?

Source: Wired.com

Weekly AP Challenge

What does the code below display? Source: Albert.io

```
x ← 3
y ← 7
z ← 21
x ← y
y ← z
DISPLAY (x)
DISPLAY (y)
```

A) 3 7
B) 3 3
C) 21 21
D) 7 21

What is variable scope?

//Created by Mr. Moore (amoore@schools.nyc.gov)

Source: Wired.com

Weekly AP Challenge

What does the code below display? Source: Albert.io

```
x ← 3
y ← 7
z ← 21
x ← y
y ← z
DISPLAY (x)
DISPLAY (y)
```

A) 3 7
B) 3 3
C) 21 21
D) 7 21

What is variable scope?

Variable scope indicates whether a variable's values can be accessed and changed: in only a subset of the program (local) or perhaps the entire program (global).

Global Justin Bieber

Local Justin Bieber

//Created by Mr. Moore (amoore@schools.nyc.gov)

Source: Wired.com

Weekly AP Challenge

What does the code below display? Source: Albert.io

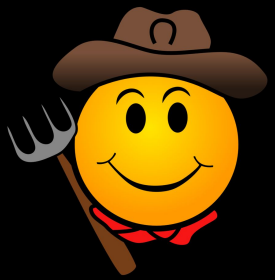
```
x ← 3
y ← 7
z ← 21
x ← y
y ← z
DISPLAY (x)
DISPLAY (y)
```

A) 3 7
B) 3 3
C) 21 21
D) 7 21

What is variable scope?

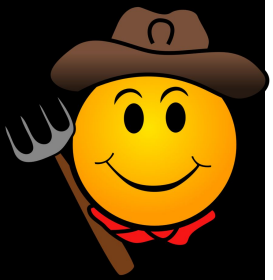
Variable scope indicates whether a variable's values can be accessed and changed: in only a subset of the program (local) or perhaps the entire program (global).

Global Justin Bieber



Value changes to farmer and everyone knows.

Local Justin Bieber



Value changes to farmer and only locals know.

//Created by Mr. Moore (amoore@schools.nyc.gov)

Source: Wired.com

Weekly AP Challenge

What does the code below display? Source: Albert.io

```
x ← 3
y ← 7
z ← 21
x ← y
y ← z
DISPLAY (x)
DISPLAY (y)
```

A) 3 7
B) 3 3
C) 21 21
D) 7 21

What is variable scope?

Variable scope indicates whether a variable's values can be accessed and changed: in only a subset of the program (local) or perhaps the entire program (global).

When you make a variable in Snap!, it is global by default. You can indicate whether it is accessible for all sprites or this sprite only.

You want to make a variable global if it should be accessible outside of the context in which it is initially used, such as **life** or **points**.

//Created by Mr. Moore (amoore@schools.nyc.gov)

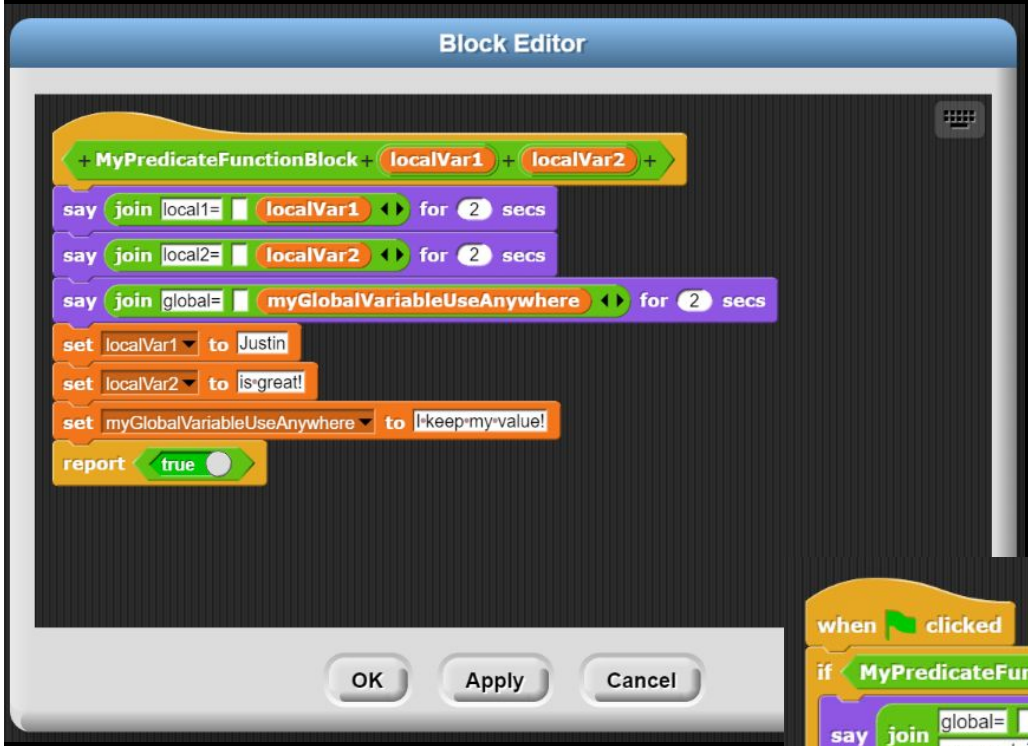
Source: Wired.com

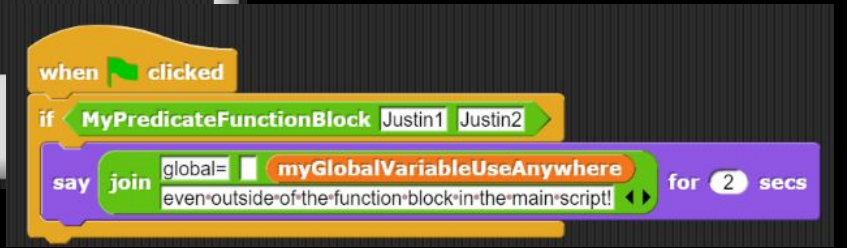
Weekly AP Challenge

What does the code below display? Source: Albert.io

```
x ← 3
y ← 7
z ← 21
x ← y
y ← z
DISPLAY (x)
DISPLAY (y)
```

A) 3 7
B) 3 3
C) 21 21
D) 7 21





Can you tell what variable keeps its value after exiting the predicate function block?

Why does it keep its value?

Why do the other variables lose their value?

//Created by Mr. Moore (amoore@schools.nyc.gov)

Networking

How does an HTML file that has been styled with CSS reach your computer, and what should we know about networking?

Question 1: What happens with packets on The Internet?
Question 2: Is a website on the World Wide Web, the Internet, or both?
Question 3: What Internet protocols are used when a page styled with CSS is accessed on a client computer?

Physical Layer(1)

and Data Link Layer(2)

The Physical (1) and Data Link (2) Layers describe the physical and wireless connection between end-point devices. The data travels from node to node. A node can be a client computer, a remote server, a gateway, a bridge, a modem, a router, a hub, or a switch. Nodes are connected on the network through ethernet, bluetooth, and wi-fi.

Point A

Client Comp
(communicatio
end-point)

Router (redistribution: nodes to networks)

Hub (redistribution: nodes to networks)

Switch (redistribution: nodes to networks)

Gateway (connection: node to node)

Bridge (connection: node to node)

Modem (connection: node to node)

Point B

Server (communication end-point)

//Created by Mr. Moore (amoore@schools.nyc.gov)

Network Layer(3)

The Network Layer (3) describes how packets are sent to IP addresses on the network (IPv4 or IPv6) and how DNS servers translate URLs to IP addresses.

DNS Server

Point A

Client Comp
(communicatio
end-point)

146.21.245.121 (IP address)

Router (redistribution: nodes to networks)

Hub (redistribution: nodes to networks)

Switch (redistribution: nodes to networks)

Gateway (connection: node to node)

Bridge (connection: node to node)

Modem (connection: node to node)

Point B

Server (communication end-point)

fluffydogs.com (URL)

146.21.245.122 (IP address)

//Created by Mr. Moore (amoore@schools.nyc.gov)

Transport Layer(4)

The Transport Layer (4) describes how data is transmitted through the network. TCP (Transmission Control Protocol) attempts to receive all packets. UDP (User Datagram Protocol) is focused more on speed of transmission and will accept the packets it gets even if it doesn't get all of them.

DNS Server

TCP (I want it right!)

UDP (I want it fast!)

Point A

Client Comp
(communicatio
end-point)

146.21.245.121 (IP address)

Router (redistribution: nodes to networks)

Hub (redistribution: nodes to networks)

Switch (redistribution: nodes to networks)

Gateway (connection: node to node)

Bridge (connection: node to node)

Modem (connection: node to node)

Point B

Server (communication end-point)

fluffydogs.com (URL)

146.21.245.122 (IP address)

//Created by Mr. Moore (amoore@schools.nyc.gov)

Application Layer(5)

The Application Layer (5) details how the internet follows different protocols for email (SMTP), Web browsing (HTTP/HTTPS), and file transfer (FTP).

SMTP
Simple Mail
Transfer
Protocol
(Email)

HTTP/HTTPS
Hypertext
Transfer
Protocol
(Websites)

DNS Server

TCP (I want it right!)

UDP (I want it fast!)

Point A

Client Comp
(communicatio
end-point)

146.21.245.121 (IP address)

Router
(redistribution:
nodes to networks)

Hub (redistribution:
nodes to networks)

Switch
(redistribution:
nodes to networks)

Gateway
(connection: node to
node)

Bridge (connection:
node to node)

Modem (connection:
node to node)

Point B

Server
(communication
end-point)

fluffydogs.com (URL)

146.21.245.123 (IP address)

//Created by Mr. Moore (amoore@schools.nyc.gov)

Abstraction

How does the use of CSS style sheets add a layer of abstraction to our project, and what should I know about abstraction?

Question 1: What is abstraction in Computer Science?
Question 2: Is a social media like button an abstraction?
Question 3: Could we say that using CSS is adding a layer of abstraction to our Web development projects?

See this review of programming languages from Professor Hu of Tsinghua University. What are differences between **high-level languages** and **low-level languages**?

Levels of Programming Languages

High-level program

```
class Triangle {
  ...
  float surface()
    return b*h/2;
  }
```

Low-level program

```
LOAD r1,b
LOAD r2,h
MUL r1,r2
DIV r1,#2
RET
```

Executable Machine code

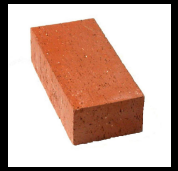
```
0001001001000101
0010010011101100
10101101001...
```

High-level Languages
More abstraction
Reduced complexity
Easier for a human to understand

Low-level Languages
Less abstraction
Increased complexity
Harder for a human to understand

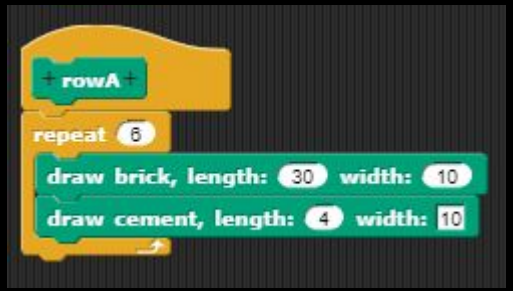
//Created by Mr. Moore (amoore@schools.nyc.gov)

This is an **abstraction** because it **hides complexity** so that the user has a **simplified** experience that is closer to what a **human** would understand.



VS

ABSTRACTION
Reduces complexity
Closer to human understanding
Anyone can use this!



ORIGINAL CODE
More complex
Closer to computer functioning
A computer or skilled programmer can use this!

(Note that even this code contains abstractions and can be broken down further and further until we get to the machine code level.)

//Created by Mr. Moore (amoore@schools.nyc.gov)

Review this explanation of procedural abstraction from BJC Unit 3 Lab 1 Page 4. What is **procedural abstraction**?

Vocabulary: Abstraction

As you learned in Unit 1 Lab 2 Page 2: Making Programs Talk, **procedural abstraction** is the process of developing a program by breaking up a large problem into smaller sub-problems.

Creating a **draw brick** block lets you think in terms of a procedure with a name related to the problem you are solving. This makes your code easier to read, and once you've coded and debugged the block, you don't need to think about how it works each time you use it. That's the beauty of procedural abstraction.

//Created by Mr. Moore (amoore@schools.nyc.gov)

Activity #2

Assuming prior knowledge of basic HTML, which we have learned prior to this lesson, do the following for homework:

1) *Go to this URL: https://replit.com/@alexmoore77/DemoHTML#index.html*
2) *Fork the project so you can edit your own copy.*
3) *Select one task below, and submit your work.*
    a) *Mild: Update the CSS file to change the font color of <h1> to green and the body background color to yellow on all four HTML pages.*
    b) *Medium: Update the CSS file to change the display of the links so that they are blocky and change color when moused over.*
    c) *Spicy: Update the CSS file to improve the layout of each page.*