



< Web Development />

Presented by

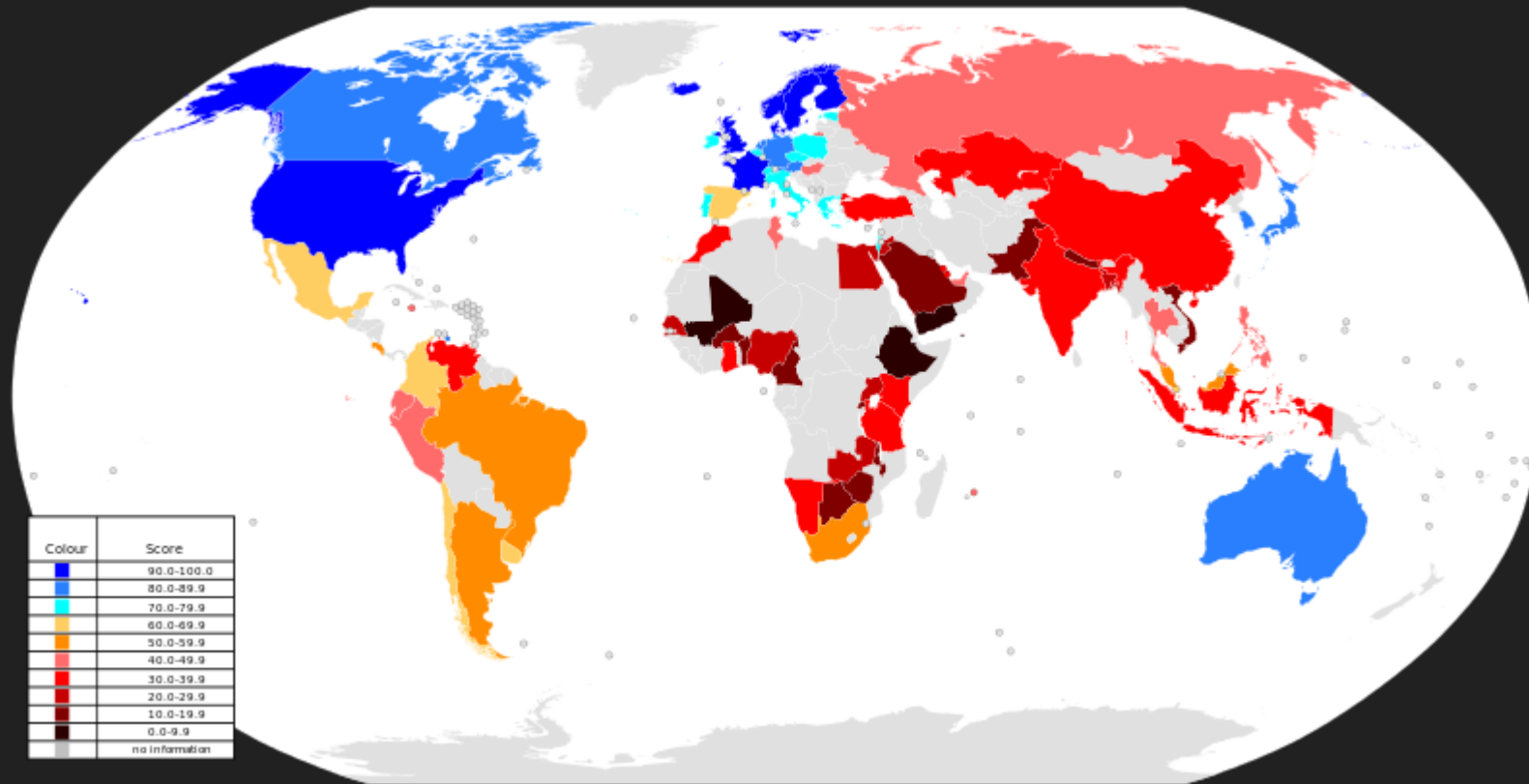
CSEC

Table of Contents

1. The World Wide Web
2. HTML
3. CSS
4. JavaScript
 - DOM
 - AJAX
5. Bootstrap, JQuery, and CDNs
6. Web Development Stacks
7. Fun Projects

Introduction to the WWW

- WWW stands for the World Wide Web
- Created by scientist Tim Berners-Lee at CERN in 1989
- Accessed by Uniform Resource Locators (URLs)
- The World Wide Web is HUGE now



Availability of the World Wide Web as of 2010

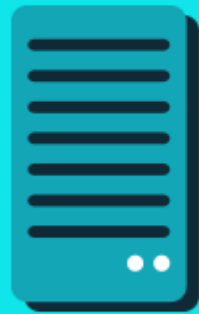
Blue = Good

Red = Bad

Web Development Roles

FULL STACK

BACK-END



FRONT-END



Some Key Terms

- HTML Hyper Text Markup Language
- CSS Cascading Style Sheets
- JS JavaScript
- SQL Structured Query Language
- DOM Domain Object Model
- AJAX Asynchronous JavaScript and XML
- LAMP Linux, Apache, MySQL, Python
- MEAN Mongo, Express, Angular, Node

HTML



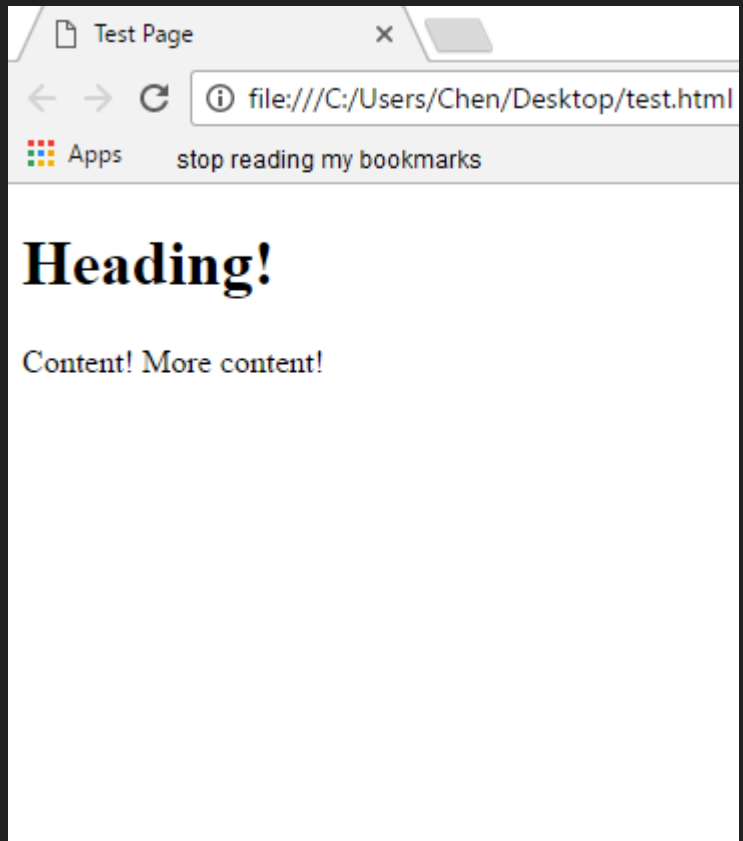
< H T M L >

HTML - Intro

HTML is the Hyper Text Markup Language

- Standard markup language for creating web pages and web applications
- “Skeleton” of a webpage – only contains content
- Has ‘tags’ that defines divisions in content (e.g. ` Hello `)
- Has ‘attributes’ that define content behavior (e.g. ` 9 `)

HTML – Starting Off



```
<!DOCTYPE html>
<html>
  <header>
    <title> Test Page </title>
  </header>
  <body>
    <h1> Heading! </h1>
    <p>
      Content! More content!
    </p>
  </body>
</html>
```

HTML - Tags

Let's look at some popular and useful tags:

- `<h[x]>` Heading of x Importance (e.g. h1, h2 ... h6)
- `<p>` Paragraph
- `` A span used to group in-line elements
- `<div>` Divisor
- `` List (can be `` or ``)
- `<form>` A form to be filled out
- `<hr>` A stylistic horizontal rule
- `
` A line break

HTML – Basic Wordplay

Introducing Headers <hx>, Spans , and Paragraph <p> tags

Basic Wordplay

Heading Tags

They can vary in size!

They also indicate importance

So this one isn't very important...

Span Tags

They can **CHANGE COLOR**. Wow!

Paragraph Tags

Your vanilla text fields

```
<body>
```

```
<h1> Basic Wordplay </h1>
```

```
<h2> Heading Tags </h2>
```

```
<p> They can vary in size! </p>
```

```
<h4> They also indicate importance </h4>
```

```
<p> So this one isn't very important... </p>
```

```
<h2> Span Tags </h2>
```

```
<p> They can <span style="color: #42aeefa"> CHANGE  
COLOR </span>. Wow!</p>
```

```
<h2> Paragraph Tags </h2>
```

```
<p> Your vanilla text fields </p>
```

```
</body>
```

HTML – Introducing the DIV

The <div> tag defines a division or a section in an HTML document

DIV example

This is a div!

This is a div within a div!

```
<body>
```

```
<h1> DIV example </h1>
```

```
<div style="color: white; background-color: #7f7f7f">
```

```
  This is a div!
```

```
<div style="background-color: #3f3f3f; margin-left: 20px">
```

```
  This is a div within a div!
```

```
</div>
```

```
</div>
```

```
</body>
```

HTML – Lists

The or tags define an (un)ordered list. are elements

Having fun with Lists

Ordered List Unordered List

- | | |
|-----------|-------------------------------|
| 1. CSCA08 | • Cream colored ponies |
| 2. CSCA08 | • Crisp apple strudels |
| 3. CSCA08 | • Door bells and sleigh bells |
| 4. CSCA08 | • Schnitzel with noodles |

```
<body>
  <h1> Having fun with Lists </h1>
  <div style="float: left">
    <h3> Ordered List </h3>
    <ol>
      <li> CSCA08 </li>
      ...
      <li> CSCA08 </li>
    </ol>
  </div>
  <div style="float: left; margin-left: 20px">
    <h3> Unordered List </h3>
    <ul>
      <li> Cream colored ponies </li>
      ...
      <li> Schnitzel with noodles </li>
    </ul>
  </div>
</body>
```

HTML – Forms

The <form> suggests a form field for filling out information

Oh Joy! Forms...

Register for the Overwatch Beta

First name:

Last name:

```
<body>
```

```
<h1> Oh Joy! Forms... </h1>
```

```
<form onsubmit="alert('Haha no way you get it!');">
```

```
<h3> Register for the Overwatch Beta </h3>
```

```
First name:<br>
```

```
<input type="text" name="firstname" placeholder="Hanzo">
```

```
<br>
```

```
Last name:<br>
```

```
<input type="text" name="lastname" placeholder="Shimada">
```

```
<br><br>
```

```
<input type="submit" value="Submit">
```

```
</form>
```

```
</body>
```

HTML – Typesetting

Introduction to `<hr>` and `
` tags, as well as HTML entities

Special HTML stuff

HR and BR?

the hr tag represents a Horizontal Rule

like that. The br represents a new line. Like that.

HTML Entities

If we type ' ' HTML will only interpret it as one space.
So we need to type ' ' to have multiple spaces.

Similarly, they act as escape characters for ", ', and unique things like ©

<body>

Special HTML stuff

HR and BR?

the hr tag represents a Horizontal Rule

like that. The `br` represents a `
` new line. Like that.

HTML Entities

If we type ' ' HTML will only interpret it as one space.

So we need to type ' ' to have multiple spaces.

```
<br><br>
```

Similarly, they act as escape characters for `"`, `'`, and unique things like `©`

</body>

<CSS>



CSS - Intro

CSS is the Cascading Style Sheet

- CSS adds the pizzazz and *style* to the webpage
- Can define multiple rules, and has many properties
- Has Classes and IDs
- Be careful of responsiveness!

CSS – The Approach

For each HTML element, we can define a specific set of rules for it to follow

Let's see some examples:

- color
- margin
- font
- font-size
- background-color
- border

Plain HTML Form

```
<body>
  <div>
    <form action="action_page.php">
      <label for="fname"> First Name </label>
      <input type="text" id="fname" name="firstname">
      <label for="country"> Country </label>
      <select id="country" name="country">
        <option value="australia"> Australia </option>
        <option value="canada"> Canada </option>
        <option value="usa"> USA </option>
      </select>
      <input type="submit" value="Submit">
    </form>
  </div>
</body>
```

Fill out the form!

First Name

Last Name

Country

HTML

```
<header>
  <link rel="stylesheet" href="style.css">
</header>
<body>
  <div>
    <form action="action_page.php">
      <label for="fname">First Name</label>
      <input type="text" id="fname" name="firstname">
      <label for="country">Country</label>
      <select id="country" name="country">
        <option
value="australia">Australia</option>
        <option value="canada">Canada</option>
        <option value="usa">USA</option>
      </select>
      <input type="submit" value="Submit">
    </form>
  </div>
</body>
```

CSS

```
input[type=text], select {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  border-radius: 4px;
  box-sizing: border-box;
}

input[type=submit]:hover {
  background-color: #45a049;
}

div {
  border-radius: 5px;
  background-color: #f2f2f2;
  padding: 20px;
}

input[type=submit] {
  width: 100%;
  background-color: #4CAF50;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}
```

Using CSS to style a HTML Form

First Name

Last Name

State

Submit

Fill out the form!

First Name

Last Name

Country

Australia ▼

Submit

CSS – Classes and IDs

ID's and Classes are "hooks". Style elements consistently:

Class

Consistently style multiple elements throughout the page

```
<div class="picture">
```

IDs

Style one element in particular

```
<div id="user-content">
```

Classes / IDs in CSS

HTML

```
<body>
  <div>
    
  </div>
  <div
    <p id="user-content">
      This is the Osteospermum, also called the
      daisybushes
    </p>
  </div>
</body>
```

CSS

```
#picture{
    border: 2px;
    border-radius: 2px;
}

.user-content{
    color: white;
    margin: 15px;
    font-size: 30px;
}
```

Classes / IDs in CSS



This is the Osteospermum, also called the daisybushes

J A A S C R I P T

HTML



CSS



JS



JavaScript - Intro

JS is JavaScript, is a high-level programming language for the web

- Doing calculations
- Adding things to the webpage after it loads
- Make the webpage 'dynamic'
- Asynchronous JavaScript and XML (AJAX)

```
<body>
  Last login:
  <script>
    var today = new Date();
    var dd = today.getDate();
    var mm = today.getMonth()+1; //January is 0!
    var yyyy = today.getFullYear();
    if(dd<10) {
      dd='0'+dd
    }
    if(mm<10) {
      mm='0'+mm
    }
    today = mm+'/'+dd+'/'+yyyy;
    document.write(today);
  </script>
</body>
```

Last login: 10/27/2016

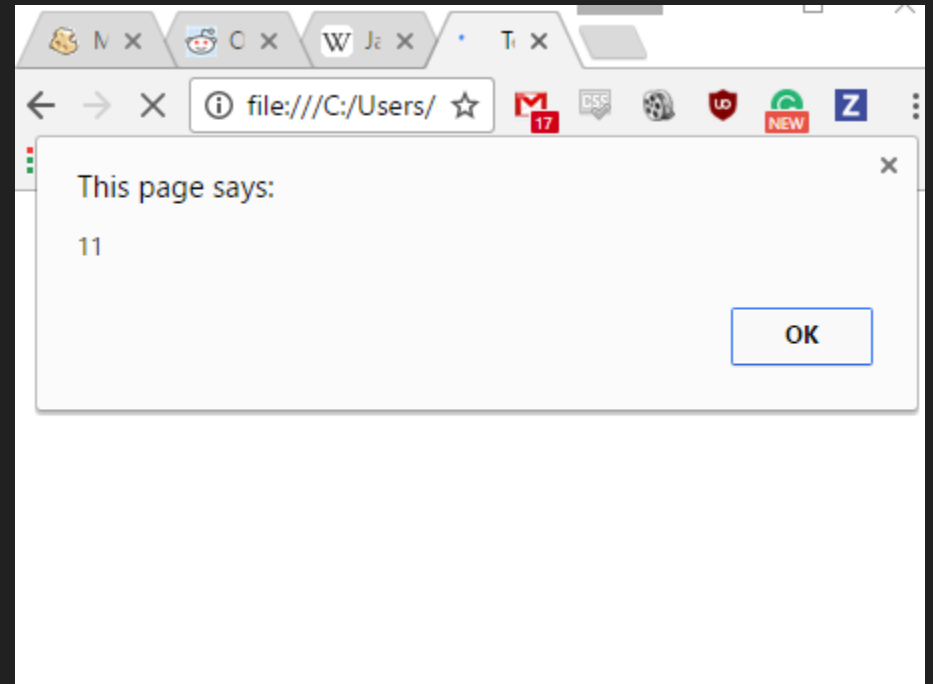
JavaScript - Functions

We can create JavaScript functions as well. Syntax:

```
function addTwoNumbers(){  
    var x = 5;  
    var y = 6;  
    alert(x+y);  
}
```

```
<body onload="addTwoNumbers();">  
    Something...  
</body>
```

```
<script>  
    function addTwoNumbers(){  
        var x = 5;  
        var y = 6;  
        alert(x+y);  
    }  
</script>
```



JavaScript - CDNs and JQuery

How do we use a library? We have one of two choices:

1. Download the JavaScript file and put it into our folder

```
<script src="js/smoothscroll.js"></script>
```

2. Use JQuery to 'insert' the JavaScript into our code

```
<link rel="stylesheet" href=https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css>
```

JavaScript – The DOM

DOM stands for Document Object Model.

- Anything found in an HTML or XML document can be accessed, changed, deleted, or added using the Document Object Model
- Can adjust style, can adjust content, etc.
- Very powerful tool to be able to create dynamic webpages

JavaScript - Functions

We can use the DOM style of programming by doing this:

```
<script>
  function changeColor(newColor) {
    var elem = document.getElementById("para1");
    elem.style.color = newColor;
  }
</script>
<body>
  <p id="para1">Some text here</p>
  <button onclick="changeColor('blue');">blue</button>
  <button onclick="changeColor('red');">red</button>
</body>
```



JavaScript – AJAX

AJAX stands for Asynchronous JavaScript and XML

- Web applications can send data to and retrieve from a server asynchronously
- Does not interfere with the display and behavior of the existing page
- Complex name for a simple mechanism

JavaScript - Functions

We can use AJAX like this...













Let's take a look at www.byxc.me/ta/a08

Notice how the content changes when I 'cat' something? If we look at the source code, we can see that it's not hardcoded.

Putting it together

How do we organize our three main players?

We recommend something of the following structure:

 assets	10/24/2016 10:00 ...	File folder
↳  logo	10/24/2016 12:17 AM	PNG File
↳  spc	10/24/2016 12:17 AM	JPG File
 css	10/24/2016 9:52 PM	File folder
↳  style	10/24/2016 9:52 PM	Cascading Style S...
 js	10/24/2016 12:17 ...	File folder
↳  app	10/24/2016 12:17 ...	JavaScript File
↳  index	10/24/2016 12:17 ...	JavaScript File
↳  papaparse.min	10/24/2016 12:17 ...	JavaScript File
 home	10/24/2016 10:00 ...	HTML File
 index	10/24/2016 10:00 ...	HTML File
 register	10/24/2016 10:00 ...	HTML File

Putting it together

Tell a HTML page to refer to CSS and JavaScript files, indicate in head

css: <link rel="stylesheet" href="LOCATION">

JavaScript: <script src="LOCATION"></script>

```
<!doctype html>
<html lang="en" class="no-js">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/png" sizes="32x32" href="img/favicon.png">
  <link rel="stylesheet" href="css/style.css"> <!-- Resource style -->
  <script src="js/modernizr.js"></script> <!-- Modernizr -->
  <title>Brian Chen</title>
</head>
```



Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web.

BootStrap - Intro

Bootstrap is very powerful. It allows us to:

- Very quickly create working websites
- Use existing JavaScript/CSS libraries to complete our site
- Make sure its responsive

BootStrap - Example

We can use Bootstrap like this...

Let's take a look at <http://presentation.creative-tim.com/>

- Navigation Bar
- Column Divisors
- Smooth Scrolling
- Responsiveness

Relational Databases



What are Relational Databases

Relational Databases are digital databases based on relational models of data

- Databases are basically used in everything, it is **very** useful knowledge to know how to create an efficient data structure
- A lot goes into creating efficient databases
- Data Science is the field that covers this. Offered at UTSC is CSCC43 and CSCD43

Structured Query Language - Intro

SQL is by far the most used relational database management language

- A few different iterations all with different uses: MySQL, NoSQL, SQLite, etc
- Stores in JSON (most common), Array, CSV, Dictionaries, etc.
- Queries read mostly like plain English

- Query syntax
 - SELECT** <desired attributes>
 - FROM** <one or more tables>
 - WHERE** <predicate holds for selected tuple>
 - GROUP BY** <key columns, aggregations>
 - HAVING** <predicate holds for selected group>
 - ORDER BY** <columns to sort>

“SELECT * FROM USERS WHERE AGE > 18”



MERN

The easiest way to build isomorphic **JavaScript** apps using **React** and **Redux**.

```
> npm install -g mern-cli  
> mern <YourAppName>|
```



MEAN/LAMP

MEAN – MongoDB, Express.js, Angular.js, Node.js

LAMP – Linux, Apache, MySQL, PHP

- MEAN is much newer however as a result has much less documentations so may be harder to implement
- MEAN is full-stack JavaScript, no need for SQL or anything like that
- There are some iterations of these stacks such as MERN, replacing Angular with React, MongoDB with PostgreSQL, and so forth

Projects and Inspirations

A bit of a talk by **Samiul Haque**

Up Next...

Complexity And Asymptotic Analysis

HINT: READ UP ON DATA STRUCTURES

Thanks for coming

Our presentation slides can be found at:
www.utsc.utoronto.ca/~csec/resources.html

Resources



Presentation Slides

[CSEC Orientation](#)

[Intro to C++](#)

[Analyzing Problems](#)

[Data Structures](#)

Seminar Slides

[World of Work](#)