

CS2005: Lab 2

Intro to HTML, CSS and JS

Objective

Understand the basics of web core technologies - HTML, CSS and JS
- and how these work together to render websites in browsers.

This is a quick grounding on a core technology area, and something
that CS students should know.

We will quickly go over some basics (and hopefully you have some exposure), and then dive deeper into JS and frontend design patterns in later labs.

Introduction

- **HTML** (HyperText Markup Language) is the language that provides the **structure** for websites.
- **CSS** (Cascading Stylesheets) is a language specification that helps define the **presentation** of a markup language like HTML.
- Finally, **JS** (Javascript) is the high-level language that enables **interactivity** for websites.

HTML

A very, very simple basic HTML template.

```
<html>
<head>
  <title>Hello World</title>
</head>
<body>
  <h1>Hello World!</h1>
</body>
</html>
```

Hopefully you have seen this before.

HTML 2

A quick selection of tags you should know:

- Paragraphs: `<p>`
- Headings: `<h1>` to `<h6>`
- Ordered and unordered lists: ``, `` and ``
- Links: ``
- Images: ``
- Divs and spans: `<div>` and ``

Other useful ones: `iframe`, `table`, `form`, etc.

Resource: [W3schools](#)

Resource: [HTML.com](#)

CSS

Sample CSS code snippet:

```
/* this are classes */
.term-grid {
  max-width: 1000px;
  margin: auto;
  color: #222;
  border-bottom: 1px solid #ddd;
  padding: 1em 0;
}

.term-grid label:first-child {
  text-align: right;
}

/* this is an id */
#helloWorld {
  color: #fff;
}
```

CSS 2

Some basic things that CSS can specify:

- Colors, backgrounds, borders
- Basic box model: margins, paddings, height, width, overflow
- Images, media, forms styling
- Fonts styling
- Transformations - translations, rotations, etc.
- Interface properties: visibility, hidden, hover, etc.
- Layout: flexbox, grid

Resource: [W3schools](#)

CSS 3

Things you should know:

- Specifying selectors: By **class**, **id**, **attribute**, etc. (here's a **list**)
- Cascade and inheritance. Basically a later rule overwrites earlier ones, and a tag can inherit various rules (more **details**)
- Data types, especially specifying units. Most common: **px**, **em** (more **details**)
- Debugging: code inspector (use Chrome)

CSS 4

Responsitivity: Because you want the code to work across multiple form factors.

Sample media query CSS code on div class:

```
/* If the screen size is 601px or more, set font-size to 80px */
@media (min-width: 601px) {
  div.example {
    font-size: 80px;
  }
}

/* If the screen size is 600px or less, set font-size to 30px */
@media (max-width: 600px) {
  div.example {
    font-size: 30px;
  }
}
```

Other useful responsive tags: **grid**, **flexbox**

Javascript

JS is everywhere. Originally used for website client-side page behavior, its popularity has led to it being used on backend tech stacks / frameworks like node.js.

JS is a multi-paradigm mix and match of functional, object-oriented and procedural language.

Comprehensive Resource: javascript.info

JS 2

Things you should know:

- Conditionals, loops, variables, functions
- Objects, data types, arrays
- Basic debugging. `Console.log` is your friend!

JS 3

Eventually you will touch on these in JS, so good to know:

Will be covering parts of this in lab 3.

- Scoping, function binding, callbacks
- JSON
- Anonymous functions, arrow notation
- Ajax / fetch and Error handling (try / catch)
- Promises, async / await
- Recursion, classes

DOM

- The Document Object Model (DOM) is the data representation of the objects that comprise the structure and content of a document on the web.
- The backbone of HTML is tags, and every tag is an object in the DOM.
- Nested tags are children of the enclosing one.
- All these objects are accessible using JS, and we can use these to modify the page.
- For example you can access the body tag with `document.body`

Resource: DOM nodes

Resource: Walking the DOM

Assignment

Lab 2: Favourite YouTube Video

Assignment Lab 2: Setup

To be completed by **28-09-2020**(Mon) 12pm

- On your GitHub Pages, create your own repo **CSC2005Lab2**
- In the repo create a basic index.html file, and make it visible on GitHub repo pages (it's under the settings tab, under GitHub pages).
- This repo link will be of the format
`https://<username>.github.io/CSC2005Lab2/`
- You can put your CSS style tags, HTML and JS script code into one file, or organize them into files/directories.
- Check that the site is published and visible!

Assignment Lab 2: Problem

- In the index.html page, embed your favourite YouTube video in the centre of the page and centralize it.
- Add a heading and paragraph on why it is your current favourite YouTube video.
- Use CSS to style the page, demonstrating at least one use of class and/or id specification.

Assignment Lab 2: Extra Challenge

- Use a font of your choice from an external resource like Google font.
- Add a background of your choice (e.g. tiled images, gradient, etc.) and other elements (icons) such that it looks good.
- Make sure that the site looks okay whether on desktop on mobile. Write at least one simple media query. Or use grid/flexbox.

Assignment Lab 2: Comments

- This is not a design challenge - you will not be graded on design.
- You can achieve the above in multiple ways in the frontend, e.g. for centralizing you can use transform(translate), margin, vw / vh specifications, grid/flexbox, or even JS calculations.
- As long as you can demonstrate you know basic HTML /CSS, I am fine with this. You do not need to use JS in this assignment, but you can choose to use this if needed.

Questions?



Chi-Loong | V/R