Week 10

Last section :'(
The DOM and Javascript

http://bit.ly/2iCvxY1

Attendance

Javascript

- Client side scripting
- Python was server side scripting
- What are the benefits of client side scripting?

Javascript Variables

- Syntax similar to C and Python
- Declare variables with `let` keyword

```
let x = 8;
let name = "Thomas";
let b = true;
let arr = [1, 2, 3];
```

Javascript Loops

- All of the C loops are back, even do-while
- break and continue work, too
- break and continue can take *labels*

Break with label

```
outer: for (let i = 0; i < 10; i++) {
    for (let j = 0; j < 10; j++) {
        console.log(i, j);
        if (i == 5 && j == 5) {
            break outer;
        }
    }
}
```

Javascript Scope

- Scope works exactly like in C
- None of the weird Python **`global`** stuff
- But if you assign to an undeclared variable it automatically becomes global

```
function oops() {
    x = 5;
    return x;
}
```

Javascript Functions

- Use the `function` keyword

```
function addNumbers(a, b) {
    return a + b;
}
```

- Functions invoked just as in C or Python

```
let five = addNumbers(2, 3);
```

Javascript Objects

- Similar to C structs or Python dicts, but can contain functions

- Functions belonging to objects are *methods*

```
let student = {
    firstName: "Thomas",
    lastName: "Lively",
    fullName: function() {
        return this.firstName + " " + this.lastName;
    }
};
console.log(student.fullName());
```

JSON

- Javascript Object Notation
- Common data format for the Internet
- Object properties must be double quoted strings

```
{
    "firstName": "Thomas",
    "lastName": "Lively",
    "age": 21
}
```

For in loops

- Used to iterate over the properties names of an object
- Can be used to iterate over the indices of an array

For in loops

```
let student = {
    firstName: "Thomas",
    lastName: "Lively",
    fullName: function() {
        return this.firstName + " " + this.lastName;
    }
};

for (let item in student) {
    console.log(item);
}
```

For of loops

- Used to iterate over the properties values of an object
- Can be used to iterate over the values in an array
- Similar to Python’s for loops

For of loops

```
let students = [“Thomas", “Maria", “Brian", “Rob”];

for (let item of student) {
    console.log(item);
}
```

Template literals

- Similar to Python's format string literals

```
let num = 50;

console.log(`This is CS ${num}!`);
```

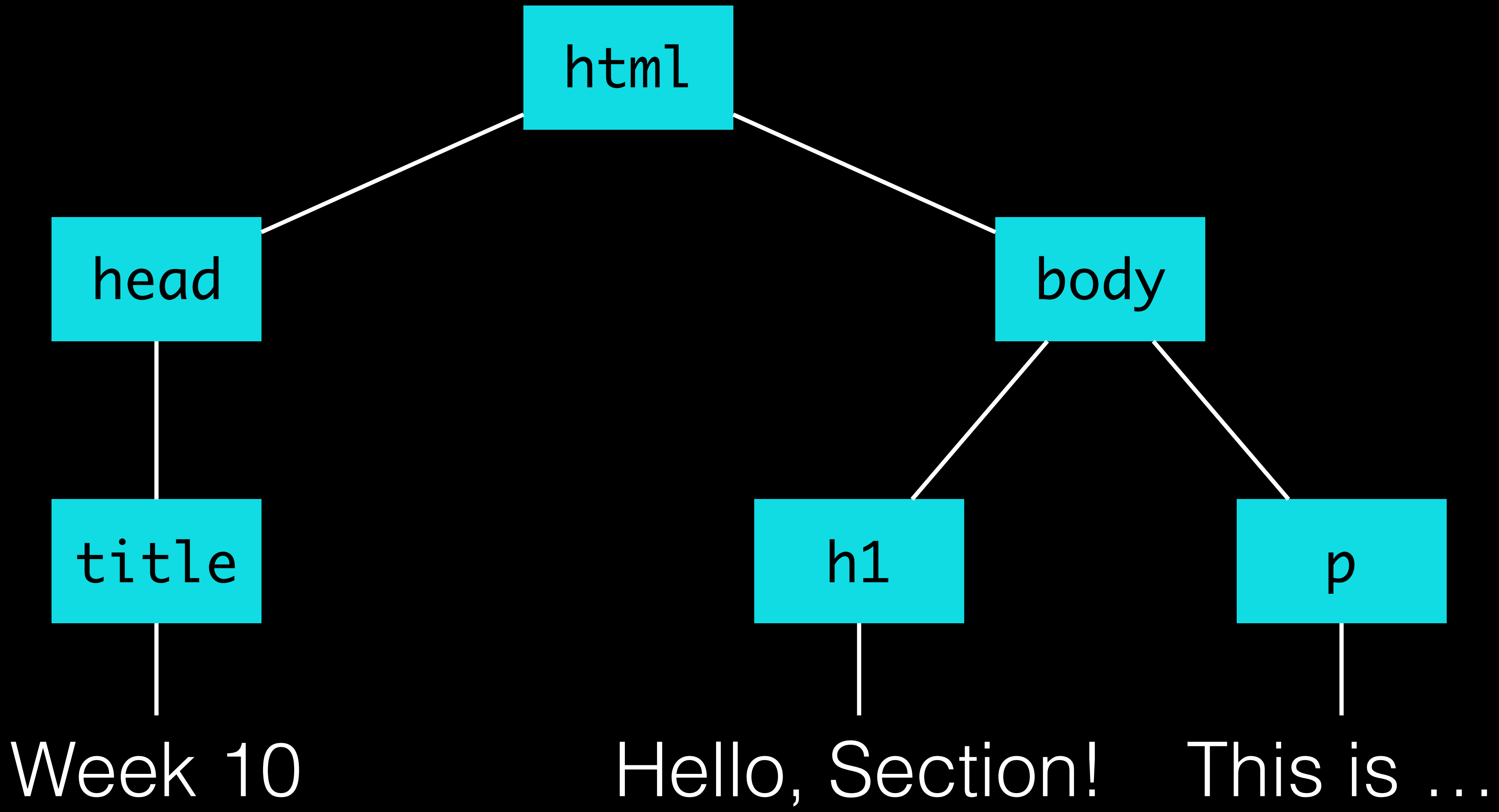
MDN Web Docs

- Formerly Mozilla Developer Network
- Now the official documentation for the web
- USE IT
- https://developer.mozilla.org/en-US/docs/Web/JavaScript
- Let’s look at the array documentation

The Document Object Model

- A hierarchal representation of the webpage
- Manipulating the DOM can change the page without refreshing

A Simple DOM



Manipulating the DOM

- Javascript can add, modify, or delete HTML elements from the DOM dynamically
- Select an element with `document.getElementById(id)`
- Set an element's inner HTML by setting `document.getElementById(id).innerHTML`

Using Javascript with HTML

- Javascript is placed between `<script>` tags
- Javascript is triggered by *events*, i.e. onclick, onload, and onchange
- Events can be defined in HTML attributes

Event Listeners

- To increase separation of concerns, event handlers can be inserted by the Javascript itself instead of in HTML attributes
- Use `element.addEventListener(event, function)`
- Multiple event handlers can be added to elements

Class Coding Exercise

- Create a webpage with a text box that can be used to dynamically add elements to a list on the page.