CS50 Section 10
Somewhere in Between

Annaleah Ernst, TF

Agenda

- Javascript
  - (and HTML)
  - jQuery
- The DOM

Javascript

- Dynamic programming language used by web browsers on the client side
  - As opposed to Python which is on the server side
  - Allows users to communicate asynchronously with browser
- Has characteristics of…
  - Functional programming
  - Object-oriented programming
  - Imperative programming
  - Seemingly multiparidigm
- Like Python, it's an interpreted language
- Like Python, it's loosely typed (variables signified with var tag)
- Ability to use the Document Object Model (DOM) makes it popular

Javascript

- Client-side scripting language
  - Not compiled or executed on the server (as opposed to php)
  - Load a Javascript page and it executes on you machine
    - This is what we mean by client side
- Syntax is similar to C, Python
- Error check via console
  - console.log(“stuff”)

Javascript – A quick example…

- index.html

```
<!doctype html>
<html>
    <head>
        <script type = "text/javascript" src="hello.js"></script>
    </head>
    <body>
        <p>derp</p>
    </body>
</html>
```

- hello.js

```
alert("Hello, world!");
```

Javascript - Variable Declarations

- Loosely typed (don't need put type of variable)
- `var <variable_name> = <value>;`
  - If you leave off the var, you get a globally scoped variable
  - So if you want something to be in scope, use var, else its global
- Easy to change type
- Be careful with different types
- Take a look at examples.js for some examples of fun with types

Javascript - Loops

- For loop (the same as C)
  - `for (<initialize> ; <condition> ; <update>)`
    - `// do code`
- While loop (also the same as C)
  - `while (<condition>)`
    - `// do code`
- Do While (also the same as C)
  - `do`
    - `// code`
  - `while (<condition>);`
- Concrete examples in the examples.js file

Javascript - Functions

- Two ways to do functions:
- Option 1: c-like function
  - `function <function_name>(<arguments>)`
    - `return <value>;`
- Option 2: anonymous function
  - `var <variable name> = function(<arguments>)`
    - `return <value>;`
- More examples in the examples.js file!

Javascript - Arrays

- Similar to Python list
- Don’t care about different types
- Zero indexed
- Dynamically sized
- Get length with `.length`
- `var <array_name> = [<value>,<value>,…];`
- Examples in…you guessed it! examples.js

Javascript - Objects

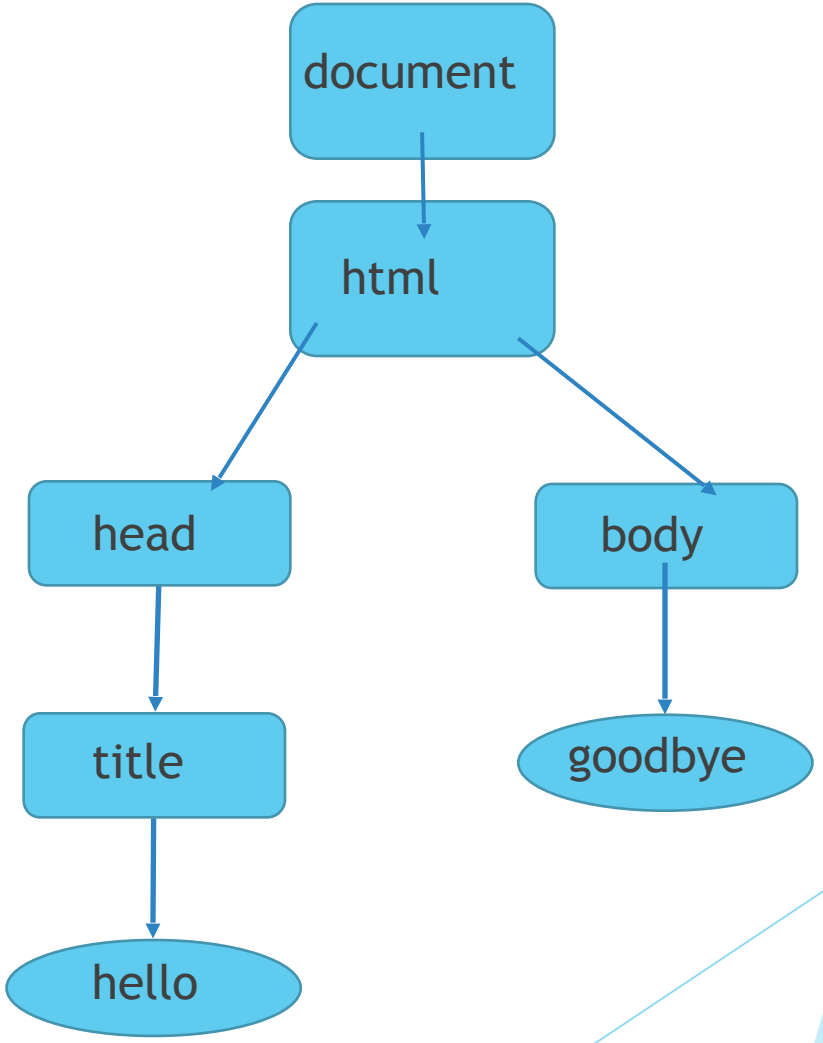
- Similar to structs in C (and dictionaries in Python)
- `var <object_name> = {};`
- Two ways to store things in the object:
  - `<object_name>["<property>"] = <value>;`
  - `<object_name>.<property>= <value>;`
- Two ways to retrieve things:
  - `<object_name>["<property>"];`
  - `<object_name>.<property>;`
- Actual examples in examples.js!

Javascript – Object, cont

- Can also just create and fill an object!
- `var <object_name> {`
  - `"<key>" : value,`
  - `"<key>" : value,`
  - `…`
- `};`
- Examples in the same place as allll the other examples about syntax

DOM – the document object model

```
<!doctype html>
<html>
    <head>
        <title>hello, world!</title>
    </head>
    <body>
        goodbye, world!
    </body>
</html>
```

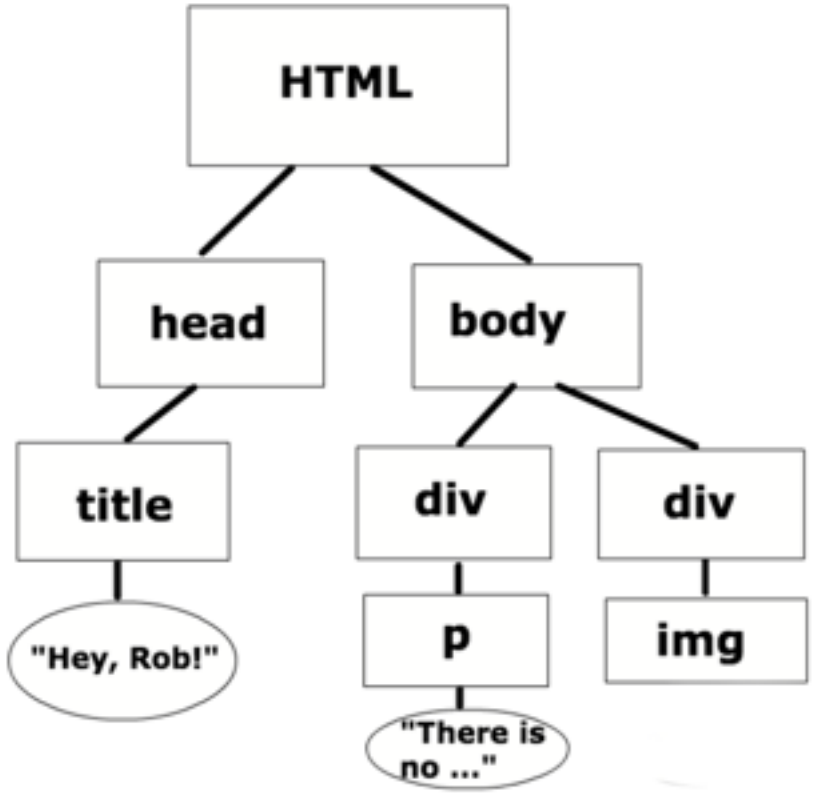


Your turn!

- Draw the DOM for the following chunk of code:

```
<!DOCTYPE html>
    <head>
        <title id="title">Hey Rob!</title>
    </head>
    <body>
        <div>
            <img id="pic" src="rob.jpg" alt="rob">
        </div>
        <div>
            <p id="quote">There is no happiness in the world, only rice...</p>
        </div>
    </body>
</html>
```

Solution:



Why is JS great with the DOM?

- Well, we can change the contents of the dom without reloading the webpage!
- Instead of using an outside form like apologize (last week), we can do dynamic alerts

DOM in JS

- `document` is a reserved word in JS
  - allows us to access the html calling the script
- Examples:
  - `document.title`
  - `document.body`
  - `document.body.innerHTML`
- Useful functions:
  - `document.getElementById("<ID>")`
  - `document.getElementByClassName("<class>")`
  - `document.getElementByTagName("<tag>")`

Your turn! clock.js

- I've made an html webpage, but I can't seem to figure out how to make it update! Maybe I should use JavaScript…
- Help me implement the functionality of my clock in "clock.js"

```
<!doctype html>
<html>
    <head>
        <link rel = "stylesheet" type="text/css" href="clock.css">
    </head>
    <body>
        <div id ="clock">This should be a clock!</div>
        <script type="text/javascript" src="clock.js"> </script>
    </body>
</html>
```

JavaScript Events

- Largely built in to JS
- User does something, JS responds
- Example: respond to load and click

```
window.onload = function () {
    var searchbutton = document.getElementById("search_button");

    searchbutton.onclick = function () {
        alert("You clicked the search button!");
    }
}
```

Your turn! party.js

- I want to throw a party on my site, but my webpage is boring!
- Help me spice things up by changing the background to a random color
- Useful functions: `Math.random`, `Math.floor`, `element.style`

```
<!doctype html>
<html>
    <head>
        <link rel = "stylesheet" type = "text/css" href = "party.css">
    </head>
    <body id="background">
        <a id = "party" href="#"> CLICK HERE TO PARTY</a>
        <script type = "text/javascript" src="party.js"></script>
    </body>
</html>
```

JQuery

- Library built on top of JS
- in your index.html file
  - `<script src = "url_to_js/jquery.js"></script>`
  - `<script src = "js/scripts.js"></script>`
- Really big, so best to link in source from Google
- in index.js file
  - `$(document).ready(function() {`
    - `// all your code (when using jquery)`
  - `});`
  - Note that the $ is essentially shorthand for Jquery
- Instinct: "Has someone already implemented this for me in Jquery?"