Web Design – Lecture-4&5

INSTRUCTOR: VIDHI KESARIA

Northeastern University
Seattle

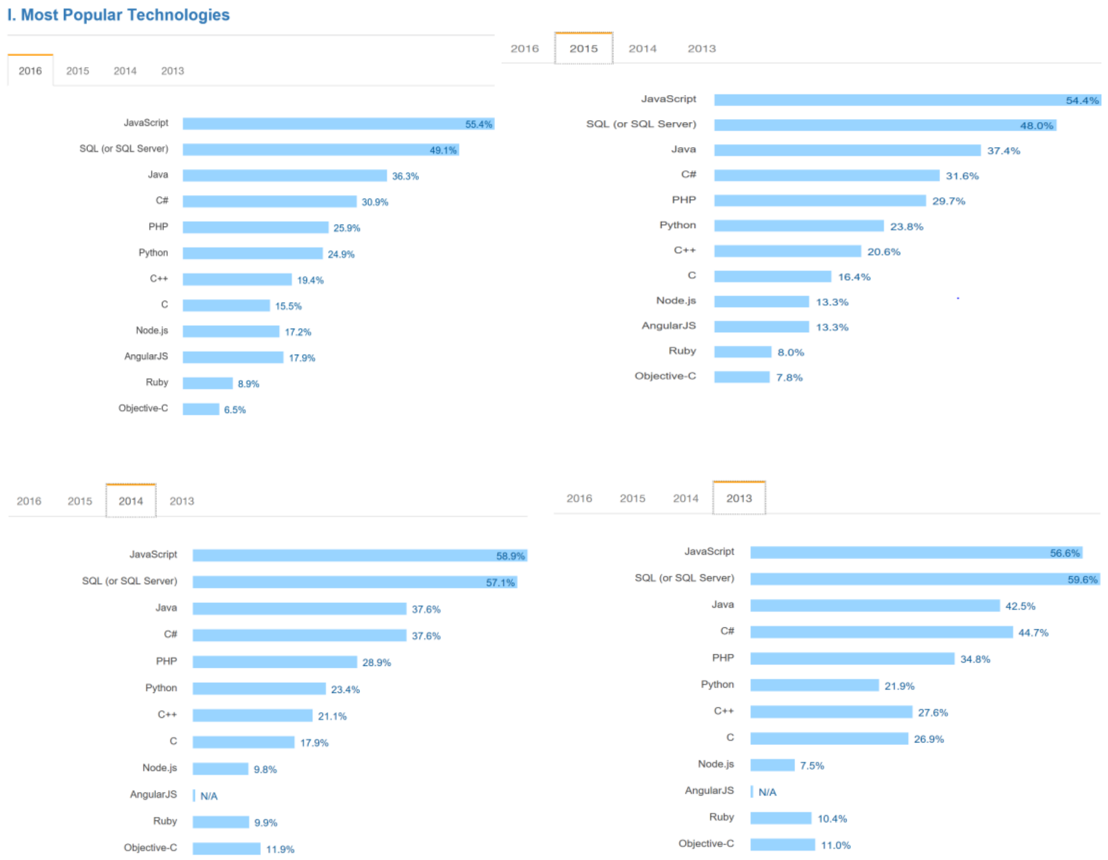
Class Schedule

- 1) Website Architecture, Design, Strategy and Planning and Creating web pages – Semantic HTML4/5
- 2) Design using CSS (covers CSS3)
- 3) Responsive web design,
- 4) Making the web page Interactive - JavaScript (Datatypes, Inline, Embedded and external JavaScript, variables, operators, loops, functions, arrays, objects
- 5) Working with DOM Model and JavaScript Events
- 6) Using Jquery, Bootstrap and other popular JavaScript libraries & Hosting on AWS
- 7) Assignment presentation with demo and Project work begins (learning project planning phase)
- 8) Reviewing wireframes and learning about design & style guide, development and content gathering
- 9) Review styled website with actual content, learn about interactive javascipt
- 10) Review functional websites, learn about website Personalization techniques - Storing information, Advanced HTML5 Apis
- 11) Review personalization features and learn about performance
- 12)Review performance & learn about AngularJS (Understand Angular JS, Separation of Responsibilities, Integrating Angular JS with existing JavaScript and JQuery Applications, Global APIs)
- 13) SEO and Accessibility Techniques & Tools + Project work
- 14) Written exams, Final project submission & Demo

Why learn Javascript?

- Essential web technology along with HTML and CSS (all browsers implement it) – only front end scripting language, on back end – demand for node is only rising

- **JavaScript** is everywhere you can even think of, **Web**, **Mobile**, **Desktop**, from client side, server side, to video games and robotics



What is JavaScript?

- The internationally standardized name of the language is ECMAScript, though it's still widely known as JavaScript as well
- It's the programming language of the web

- Usage:
  - Front-end: For interactivity (click, drag, tab, swipe)
  - Server-side as well as back-end development (through node.js)
  - JavaScript is also used to run databases like Mongo and can even be used to automate Photoshop

Resources

- Mozilla Developer Network
- "JavaScript: The Good Parts". This is written by Douglass Crockford

- Javascript
  - http://www.w3schools.com/js/default.asp
  - https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/JavaScript_basics

- Jquery
  - http://www.w3schools.com/jquery/
  - http://learn.jquery.com/
  - http://api.jquery.com/

Topics

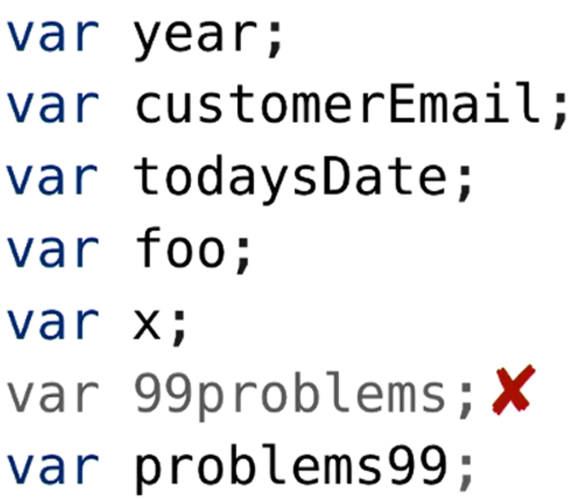
- Understanding the structure of JavaScript code
- Creating variables, functions, and loops
- Writing conditional code
- Sending messages to the console
- Working with different variable types and objects
- Creating and changing DOM objects
- Event handling
- Working with timers
- Debugging JavaScript
- Building smarter forms
- Working with CSS, HTML5, and JavaScript
- Using regular expressions

Structure of the code

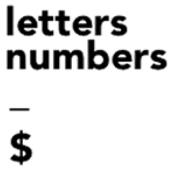
- Hello world example
- Javascript is case-sensitive
- Comment: //, <!- - multiline - - >
- Can add through below methods
  - Head through <script></script>
  - In body <script></script>
  - External script <script src= " "> -- Recommended

Variables

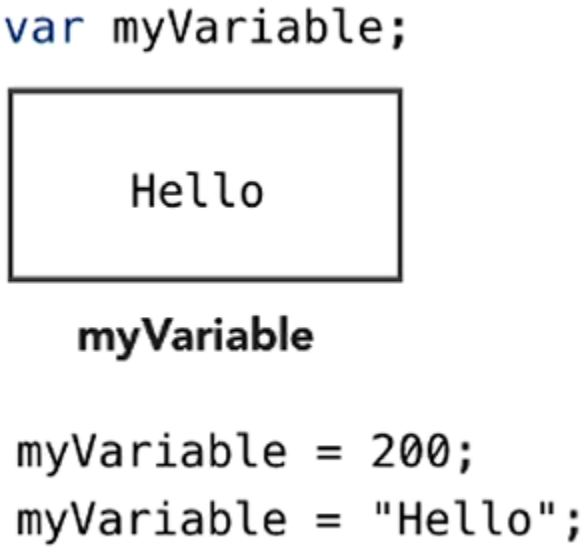
- Is a container - It's us grabbing a little piece of computer memory and giving it a name, so we can use it while our JavaScript is running.



```
var year;
var customerEmail;
var todaysDate;
var foo;
var x;
var 99problems;✗
var problems99;
```



letters
numbers
_
$



```
var myVariable;
```

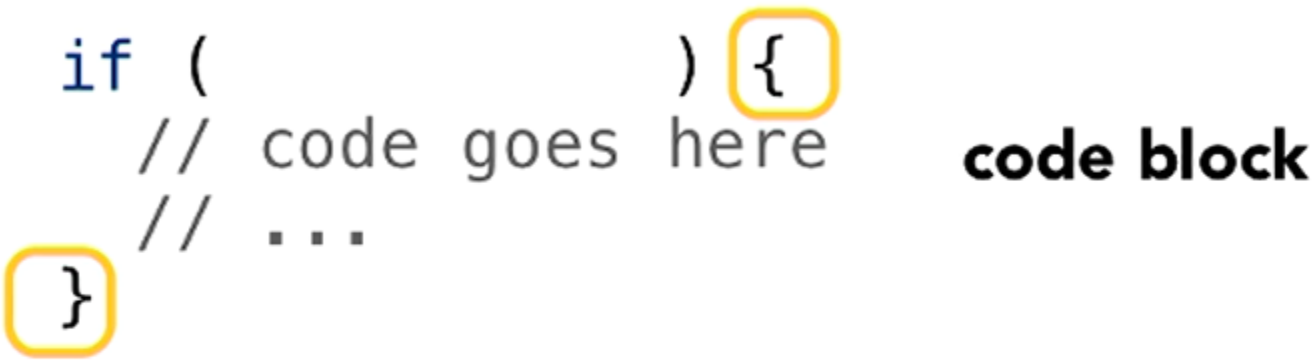
```
myVariable = 200;
myVariable = "Hello";
```

Declaration

Assignment

Conditional code

- Have code that only runs under certain conditions rather than all the time



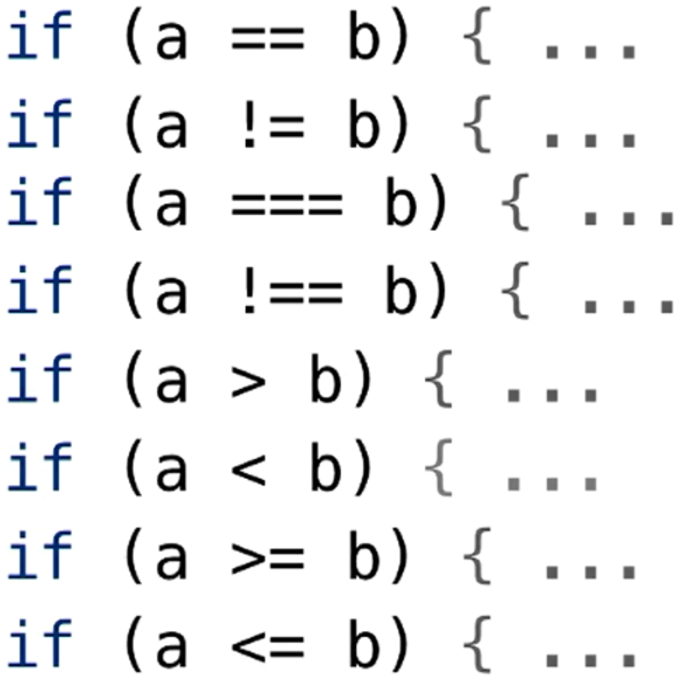
```
if (              ) {
  // code goes here
  // ...
}
```

- Let’s try that in the browser
  - var amount = 100; if(amount > 300){alert("Greater than 300");} else {alert("less than 300")};

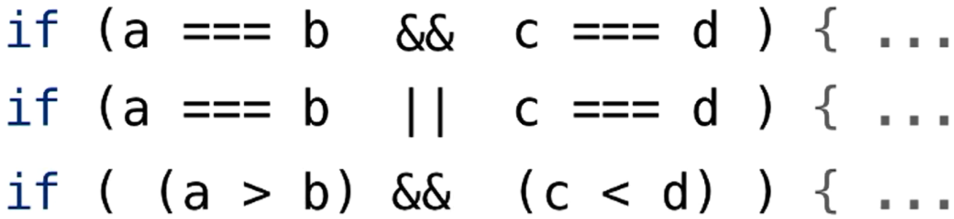
Arithmetic Operators & Assignment

- Arithmetic operator (*, /, +, -)
  - Eg. Count = count + 1; or count ++;
  - Precedence: (), X, /, +, - eg. 5+5*6 = ?
- Assignment: “=“
- Comparison operator
  - Difference between “==“ & “===“
  - Eg: var amount = "100", another = 100; if(amount === another){alert("It is strictly equal");} else if (amount == another){alert ("It is equal");}

Comparison & Logical Operators



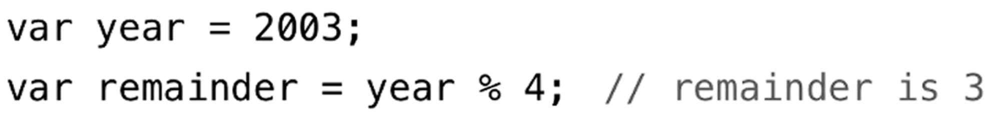
```
if (a == b) { ...
if (a != b) { ...
if (a === b) { ...
if (a !== b) { ...
if (a > b) { ...
if (a < b) { ...
if (a >= b) { ...
if (a <= b) { ...
```



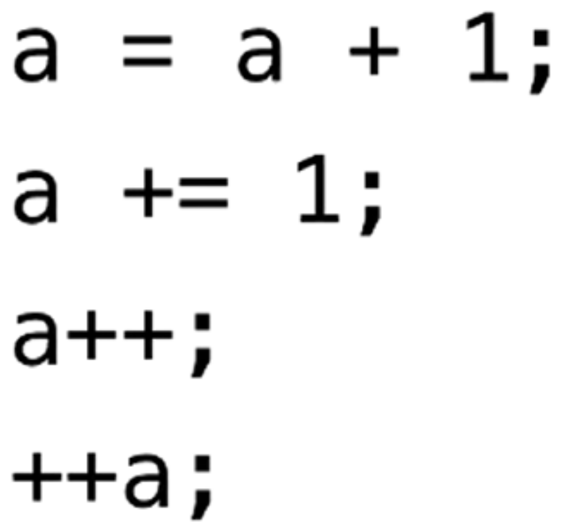
```
if (a === b  &&  c === d ) { ...
if (a === b  ||  c === d ) { ...
if ( (a > b) &&  (c < d) ) { ...
```

True/false

Modulas & Increment/decrement

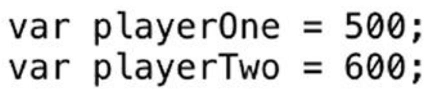


```
var year = 2003;
var remainder = year % 4;  // remainder is 3
```

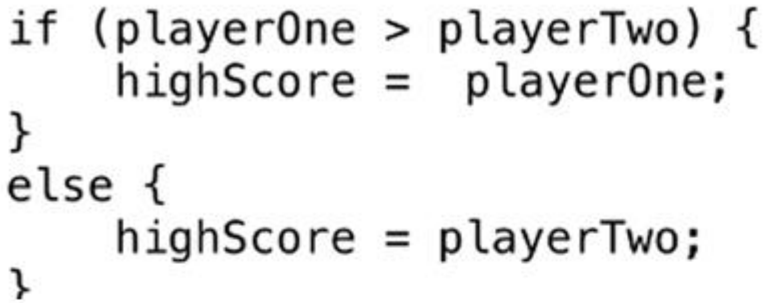


```
a = a + 1;
a += 1;
a++;
++a;
```

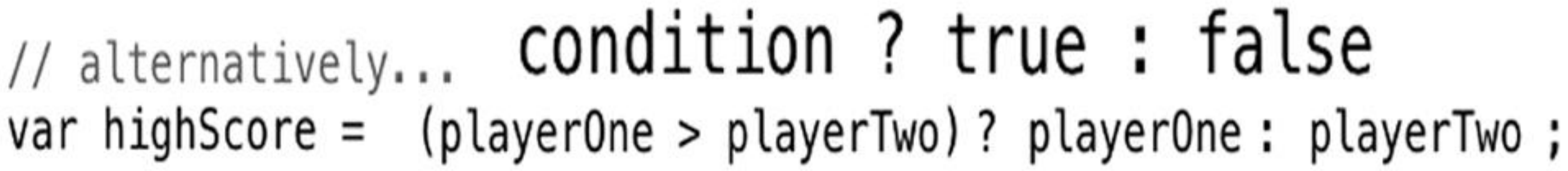
Ternary operator



```
var playerOne = 500;
var playerTwo = 600;
```



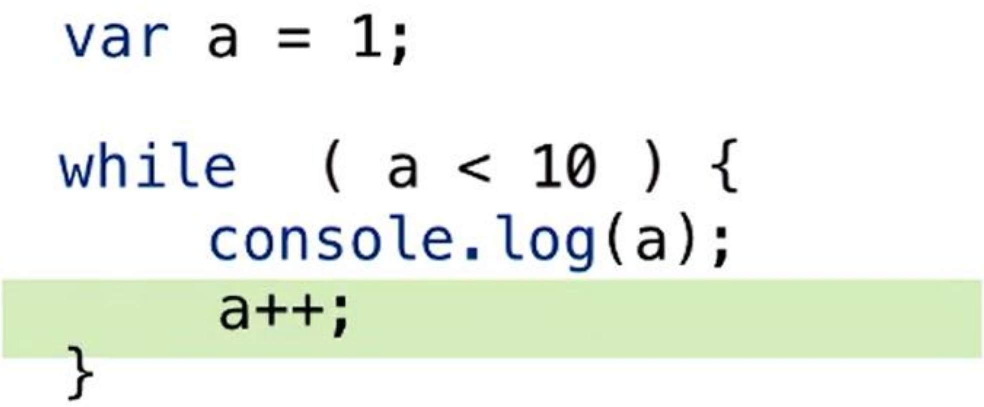
```
if (playerOne > playerTwo) {
     highScore =  playerOne;
}
else {
     highScore = playerTwo;
}
```



```
// alternatively...  condition ? true : false
var highScore =  (playerOne > playerTwo) ? playerOne : playerTwo ;
```

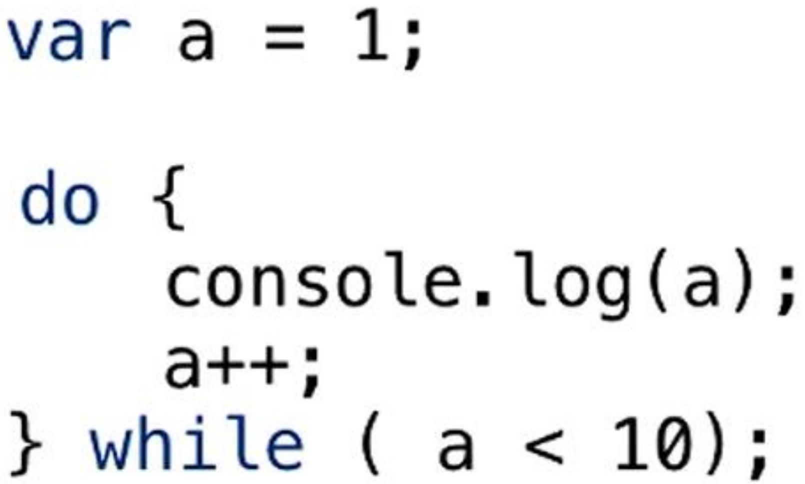
Loops

While



```
var a = 1;

while  ( a < 10 ) {
      console.log(a);
      a++;
}
```

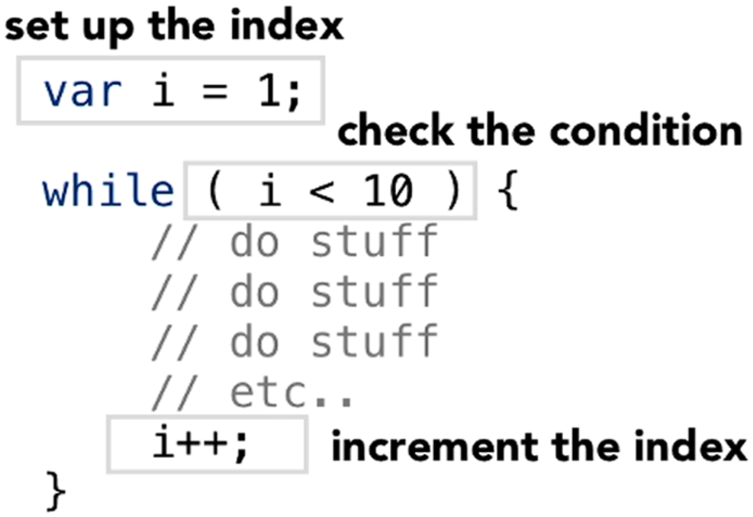
Do… while



```
var a = 1;

do {
      console.log(a);
      a++;
} while ( a < 10);
```

For Loop & Break keyword

for



set up the index

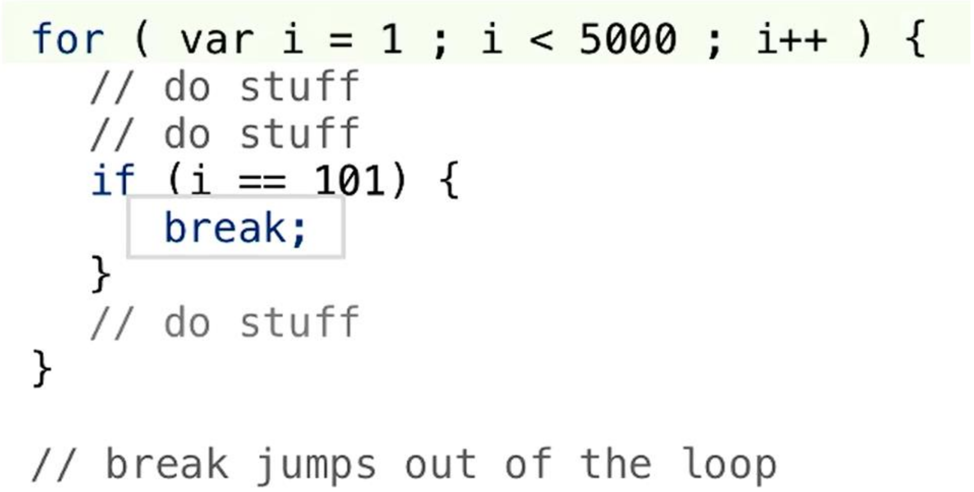
```
var i = 1;
```

check the condition

```
while ( i < 10 ) {
        // do stuff
        // do stuff
        // do stuff
        // etc..
      i++;
}
```

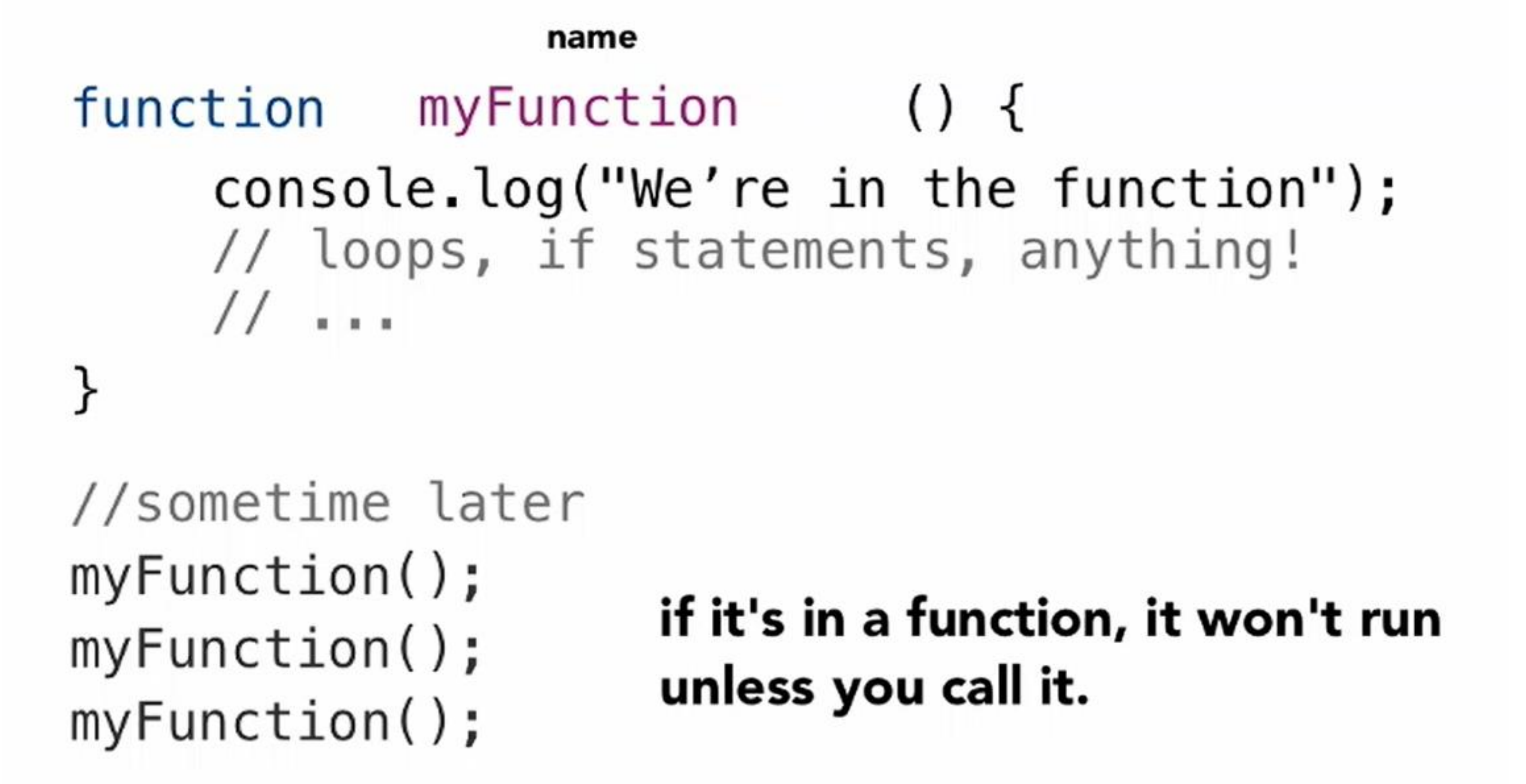
increment the index

Break/continue



```
for ( var i = 1 ; i < 5000 ; i++ ) {
   // do stuff
   // do stuff
   if (i == 101) {
      break;
   }
   // do stuff
}

// break jumps out of the loop
```

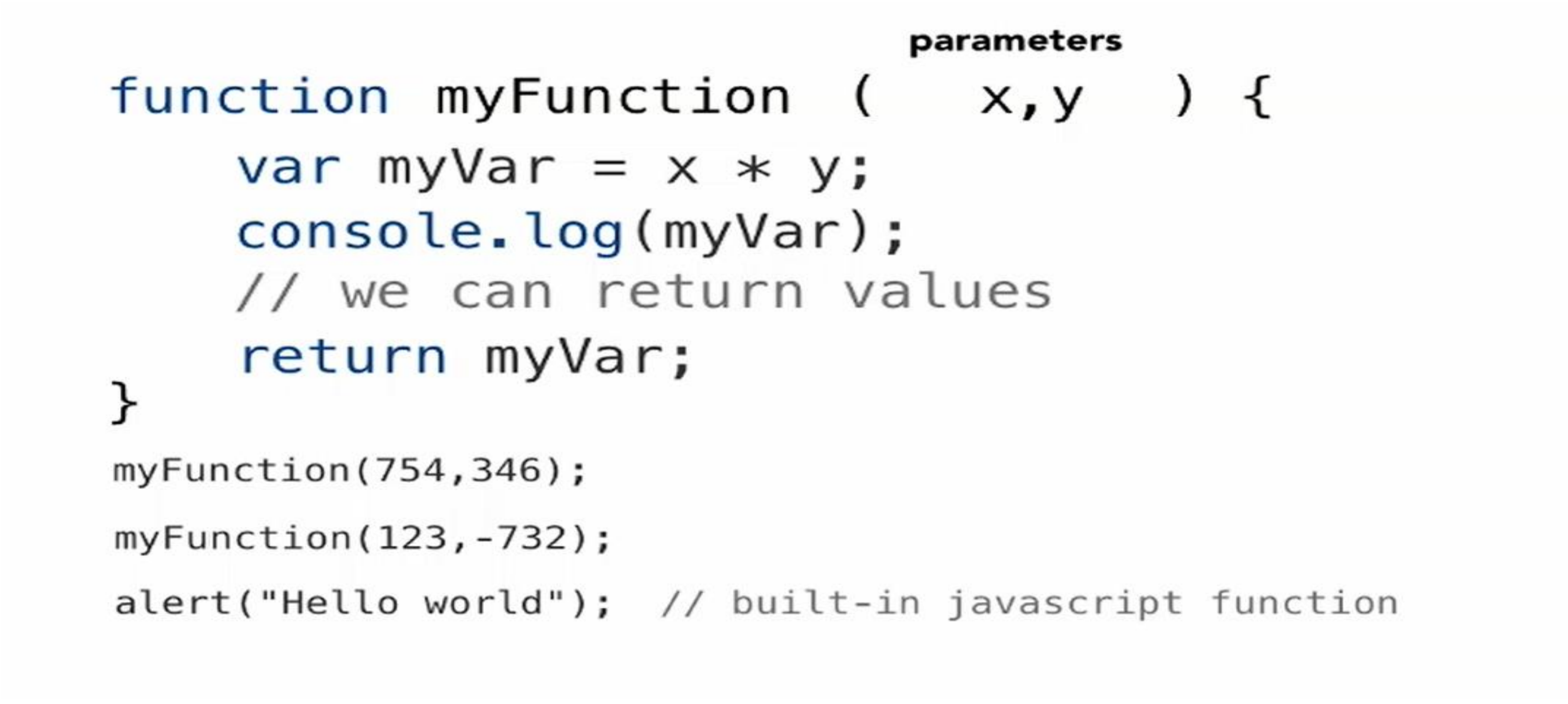
Functions (reusability)



```
function   myFunction     () {
     console.log("We're in the function");
     // loops, if statements, anything!
     // ...
}

//sometime later
myFunction();
myFunction();
myFunction();
```

if it's in a function, it won't run unless you call it.

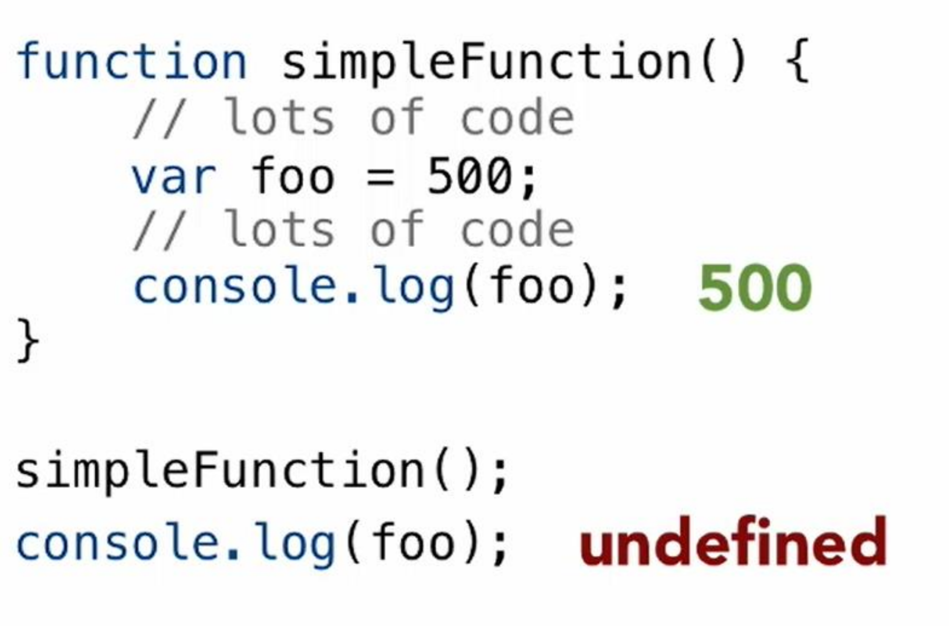
Parameterized function



```
                                  parameters
function myFunction (    x,y    ) {
    var myVar = x * y;
    console.log(myVar);
    // we can return values
    return myVar;
}
myFunction(754,346);

myFunction(123,-732);

alert("Hello world");  // built-in javascript function
```

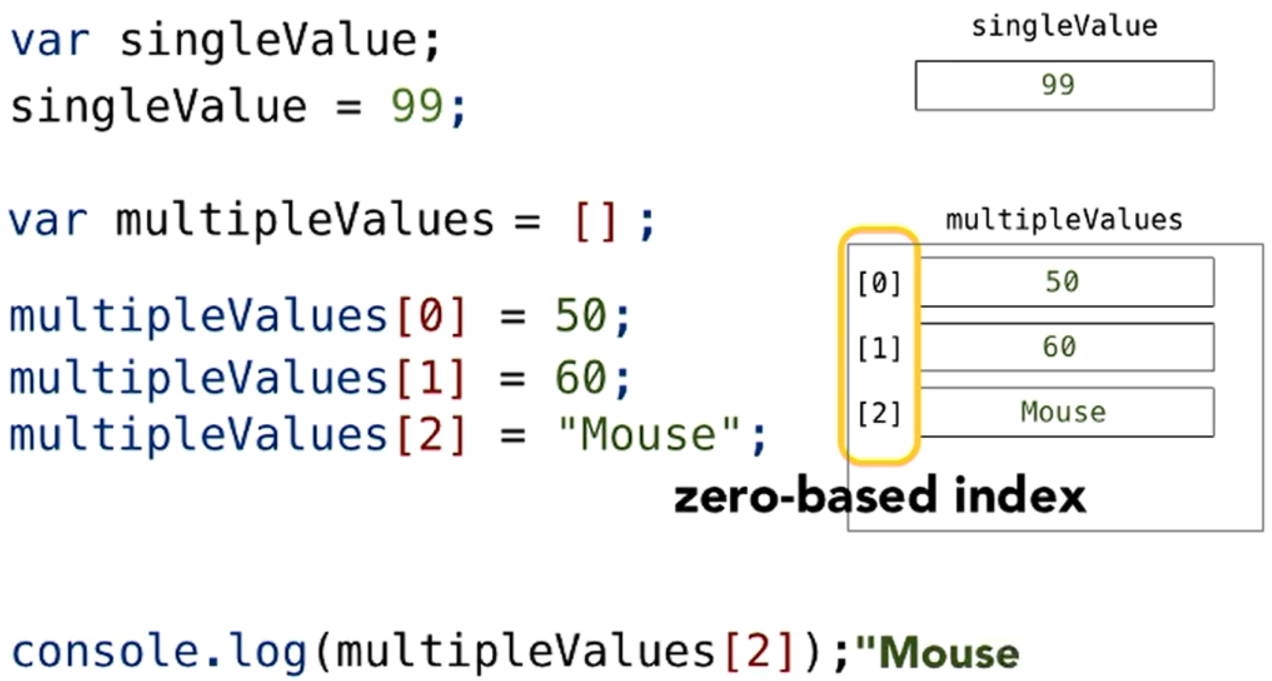
Scope

- Where is the code visible?



```
function simpleFunction() {
    // lots of code
    var foo = 500;
    // lots of code
    console.log(foo);   500
}

simpleFunction();
console.log(foo);   undefined
```

Creating arrays

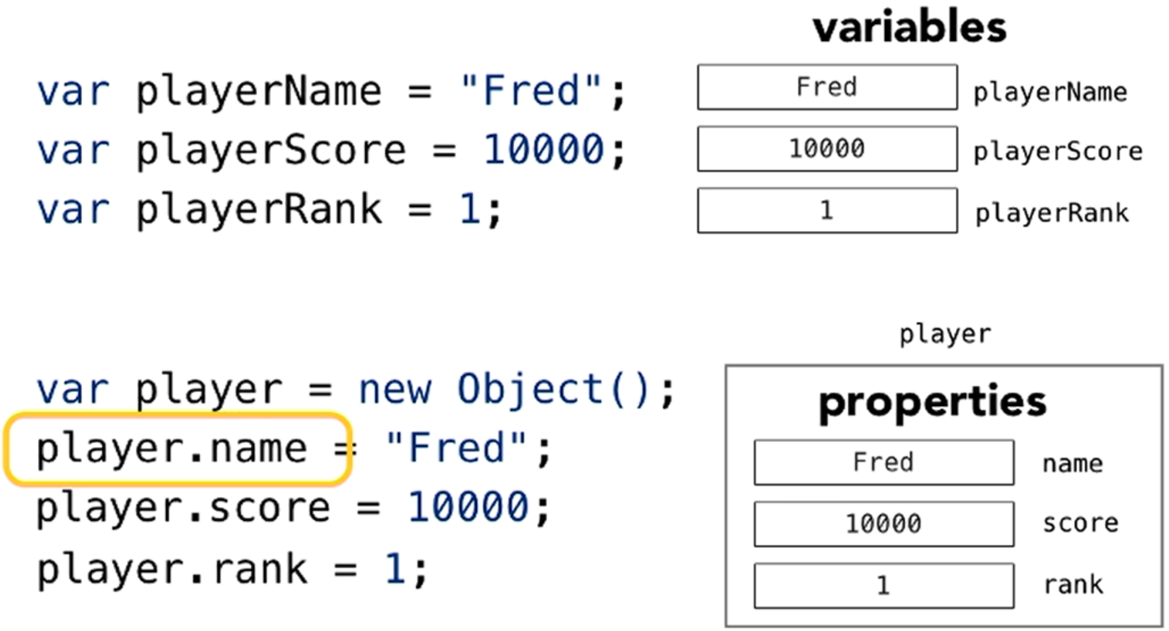


```
var singleValue;
singleValue = 99;

var multipleValues = [] ;

multipleValues[0] = 50;
multipleValues[1] = 60;
multipleValues[2] = "Mouse";


console.log(multipleValues[2]);"Mouse
```

- [Properties & methods]()
- http://www.w3schools.com/jsref/jsref_obj_array.asp

Objects

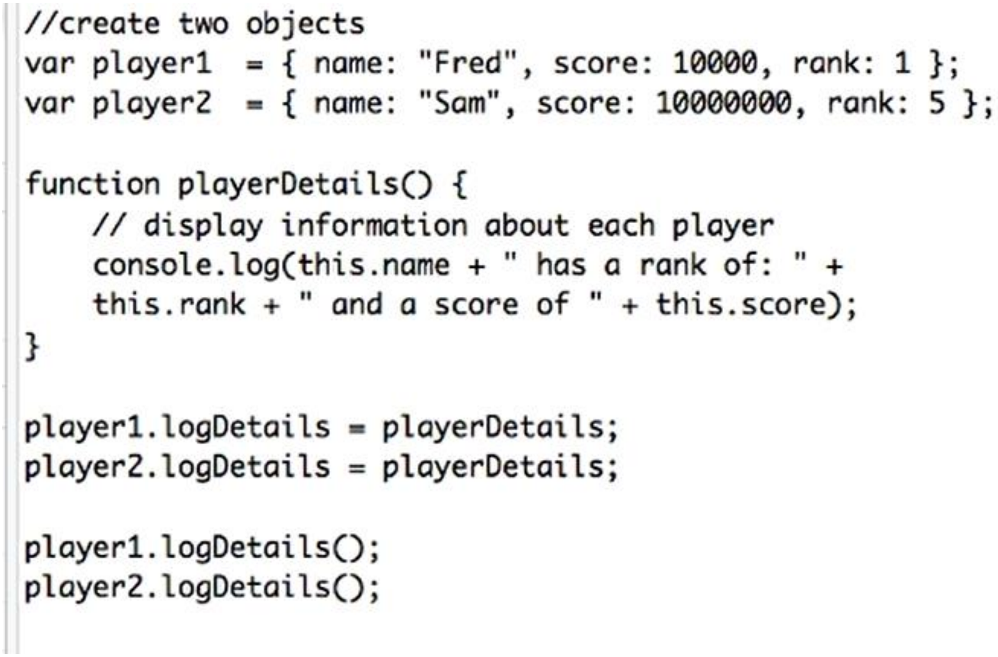
- An object just allows us to gather variables and functions that belong together and give them a name.



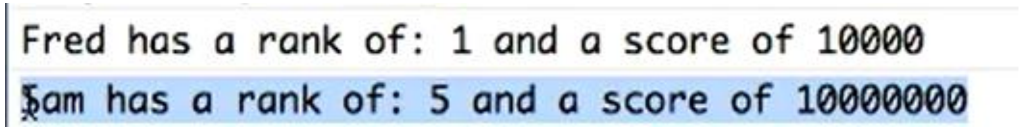
```
var playerName = "Fred";
var playerScore = 10000;
var playerRank = 1;
```

```
var player = new Object();
player.name = "Fred";
player.score = 10000;
player.rank = 1;
```

Example



```
//create two objects
var player1  = { name: "Fred", score: 10000, rank: 1 };
var player2  = { name: "Sam", score: 10000000, rank: 5 };

function playerDetails() {
    // display information about each player
    console.log(this.name + " has a rank of: " +
    this.rank + " and a score of " + this.score);
}

player1.logDetails = playerDetails;
player2.logDetails = playerDetails;

player1.logDetails();
player2.logDetails();
```



```
Fred has a rank of: 1 and a score of 10000
Sam has a rank of: 5 and a score of 10000000
```

Lab Exercise

1. Define a function max() that takes two numbers as arguments and returns the largest of them. Use the if-then-else construct available in Javascript.
2. Define a function maxOfThree() that takes three numbers as arguments and returns the largest of them.
3. Write a function that takes a character (i.e. a string of length 1) and returns true if it is a vowel, false otherwise.
4. Write a function translate() that will translate a text into "rövarspråket". That is, double every consonant and place an occurrence of "o" in between. For example, translate("this is fun") should return the string "tothohisos isos fofunon".
5. Define a function sum() and a function multiply() that sums and multiplies (respectively) all the numbers in an array of numbers. For example, sum([1,2,3,4]) should return 10, and multiply([1,2,3,4]) should return 24.
6. Define a function reverse() that computes the reversal of a string. For example, reverse("jag testar") should return the string "ratset gaj".
7. Represent a small bilingual lexicon as a Javascript object in the following fashion {"merry":"god", "christmas":"jul", "and":"och", "happy":gott", "new":"nytt", "year":"år"} and use it to translate your Christmas cards from English into Swedish.
8. Write a function findLongestWord() that takes an array of words and returns the length of the longest one.
9. Write a function filterLongWords() that takes an array of words and an integer i and returns the array of words that are longer than i.
10. Write a function charFreq() that takes a string and builds a frequency listing of the characters contained in it. Represent the frequency listing as a Javascript object. Try it with something like charFreq("abbabcbdbabdbdbabababcbcbab")

DOM – Document Object Model

DOM

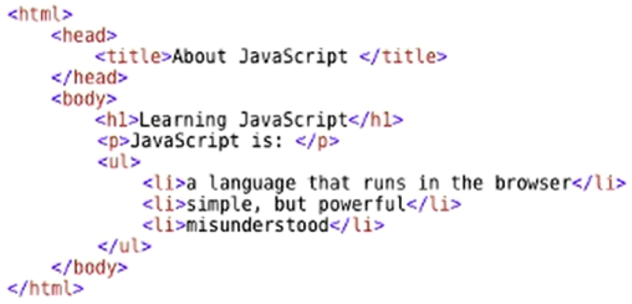
- Document – web page
- Object – thing or container of things
- Eg. <div>, <h1>
- Model – Structure

- It is the way to reach into the page from our script and the way our page can reach into our script

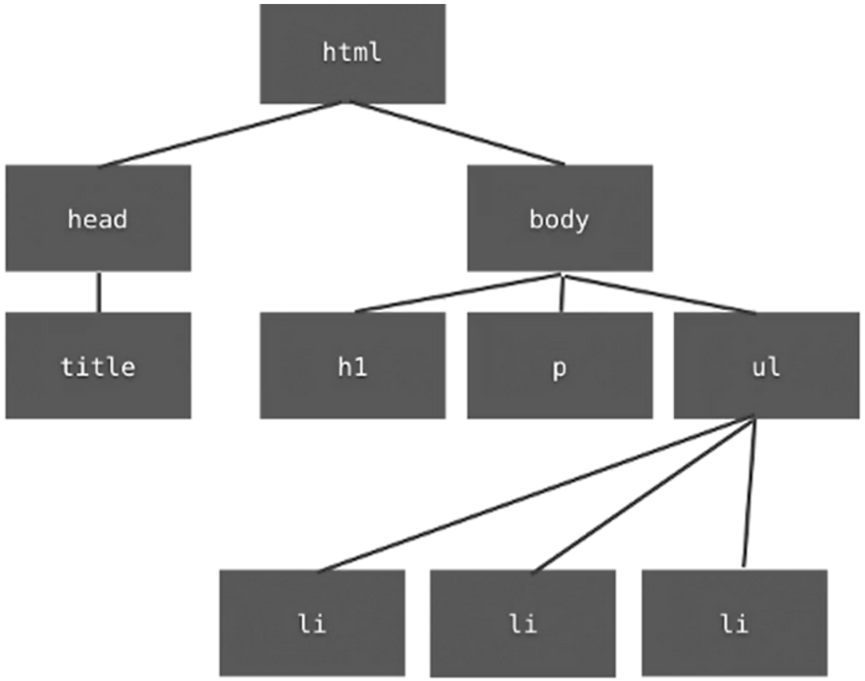
Examples

- Can write JavaScript that navigates around any page
  - get the text of the title
  - get the text of the title
  - get the third link in the menu and change how it's displaying
  - find an image and start to move it, change a link
  - creating new unordered lists, or creating paragraphs, or creating menus and inserting them into the page on the fly

DOM Structure – Nodes & Elements

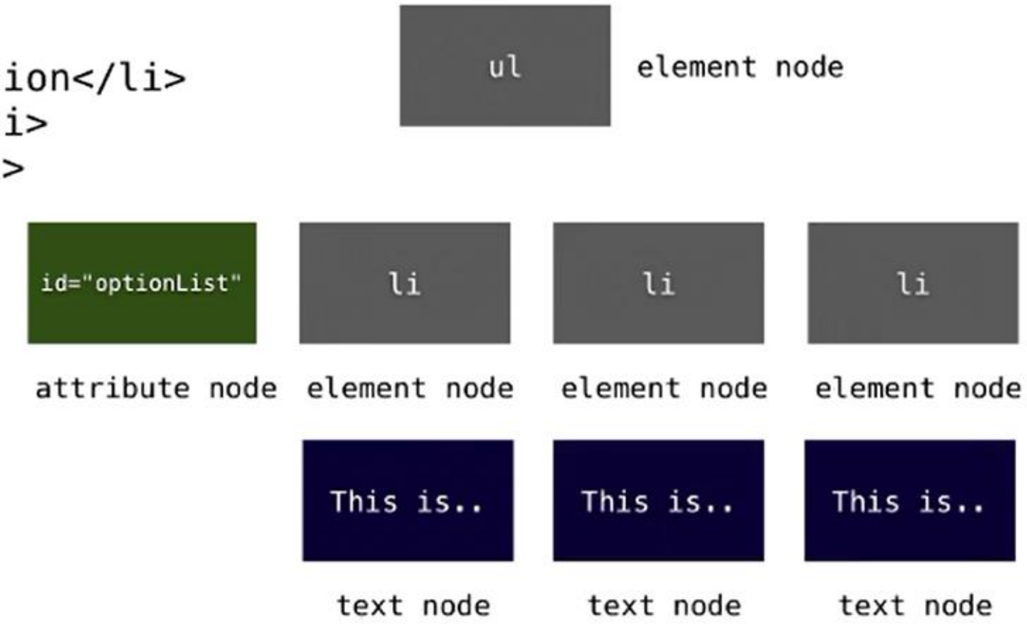


```
<html>
    <head>
        <title>About JavaScript </title>
    </head>
    <body>
        <h1>Learning JavaScript</h1>
        <p>JavaScript is: </p>
        <ul>
            <li>a language that runs in the browser</li>
            <li>simple, but powerful</li>
            <li>misunderstood</li>
        </ul>
    </body>
</html>
```



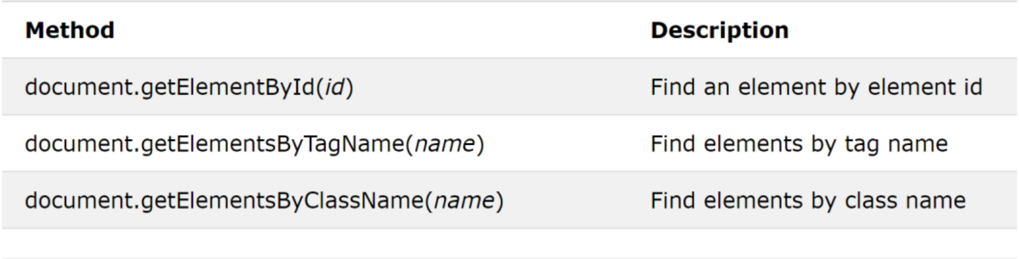
Element, Attributes & Text Nodes

```
<ul id="optionList">
  <li>This is the first option</li>
  <li>This is the second</li>
  <li>This is the third</li>
</ul>
```



Accessing DOM Elements

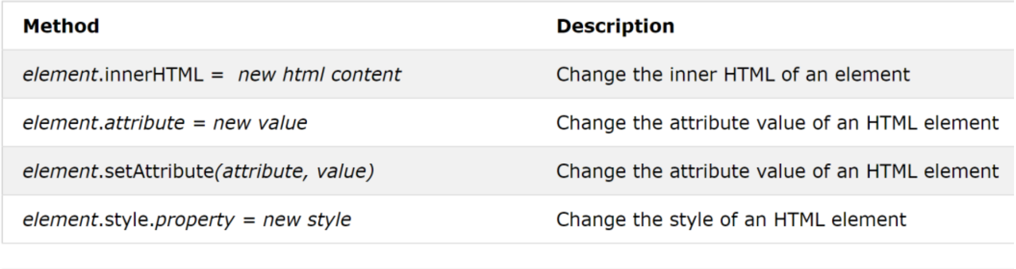
Finding HTML Elements



Method	Description
document.getElementById(*id*)	Find an element by element id
document.getElementsByTagName(*name*)	Find elements by tag name
document.getElementsByClassName(*name*)	Find elements by class name

Changing HTML Elements

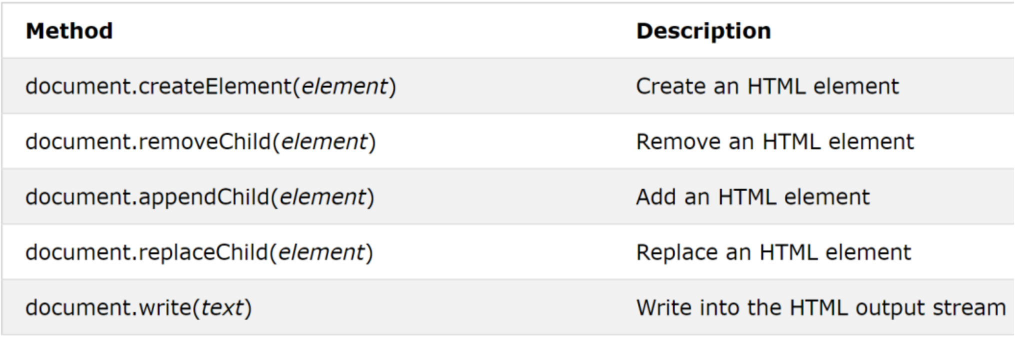
1. Get the element
2. Change the element



Method	Description
element.innerHTML = *new html content*	Change the inner HTML of an element
element.attribute = new value	Change the attribute value of an HTML element
element.setAttribute*(attribute, value)*	Change the attribute value of an HTML element
element.style.*property = new style*	Change the style of an HTML element

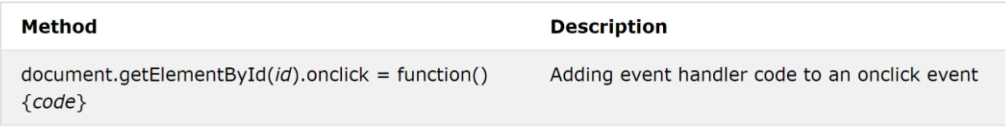
Adding and Deleting Elements

1. Create the element
2. Attach the element



Method	Description
document.createElement(*element*)	Create an HTML element
document.removeChild(*element*)	Remove an HTML element
document.appendChild(*element*)	Add an HTML element
document.replaceChild(*element*)	Replace an HTML element
document.write(*text*)	Write into the HTML output stream

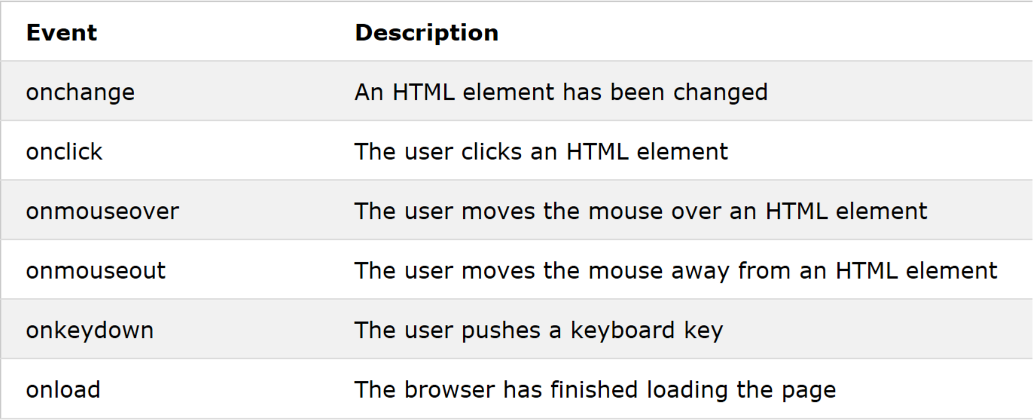
Adding Events Handlers



Method	Description
document.getElementById(*id*).onclick = function(){*code*}	Adding event handler code to an onclick event

Events

- Events are going on all the time – you want to figure out which one to respond to via eventHandler
- Dom object http://www.w3schools.com/jsref/dom_obj_event.asp
- All events - http://www.w3schools.com/tags/ref_eventattributes.asp
- Example- http://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_onclick_dropdown



Event	**Description**
onchange	An HTML element has been changed
onclick	The user clicks an HTML element
onmouseover	The user moves the mouse over an HTML element
onmouseout	The user moves the mouse away from an HTML element
onkeydown	The user pushes a keyboard key
onload	The browser has finished loading the page

Lab Exercise

- Get the time from the user and Implement a counter

- Implement the memory game web app. The game splits the screen into a 4x4 game board. Under each square hides a letter. When a player clicks on a square, that letter is revealed, and the player gets a chance to find its match by clicking on another square. If the two letters match, both letters remain visible; if they don't, they both return to their hidden state. The game ends when all letters are visible. (hint: use setTimeout function)

-

Assignment

- In your responsive portfolio website, add below functionalities-
  - Add a design chooser button in the header or footer, that loads different design for the whole website. Hint: Use javascript to change the link tag's src attribute tag and change css file. The html and js is shared
  - Add a Contact me Form with all the form validation using correct input fields and using js for validation.
  - Add features that make your portfolio interactive (so basically website responds based on the user interaction). Hint: Check out https://dcrazed.com/creative-javascript-examples/ for inspiration. Please make sure to come up with your own innovative idea and use JavaScript to begin with (avoid jquery this week)