

# Document Object Model

## Preliminary

```
window.onload = function () {  
    (your code here)  
};
```

## Objects and functions

**document** the main object that connects JavaScript to the web page

**document.createElement** function that takes a tag name and returns an object

**document.getElementById** function that retrieves an element by ID

**document.createTextNode** function that creates text from strings

**e.appendChild** function that attaches one element to another

**e.style** object that contains CSS style properties

**e.addClass** function that adds a CSS class to an element

# Handout for Code Your Own Site

## Getting Started

Welcome to the class!

The first step is to download the slides so you can follow along with the lectures: <http://bit.ly/2amZB8z>

Then, navigate to <https://github.com/clarissalittler/websites-for-beginners> and click on the "download" button and then the "download zip" option. Alternately, just use following link to download the zip directly <https://github.com/clarissalittler/websites-for-beginners/archive/master.zip> (or <http://bit.ly/2anatmW>)

By default, the zip file will be downloaded into your Downloads directory. Navigate to your downloads directory then /right/ click on the file **websites-for-beginners-master.zip** and select the option "extract here". Then navigate into the directory and you will find yourself in the right directory for this course.

## Links to resources

- General JavaScript notes: <http://bit.ly/2ahK0Zc>
- Tutorial on HTML and CSS: <http://bit.ly/2af3ifd>
- Tutorial on JavaScript and the Document Object Model: <http://bit.ly/2abKKFa>
- Neocities: <https://neocities.org/>

## Glossary of Terms

**HTML** HyperText Markup Language, the content of a web page

**CSS** Cascading Style Sheets, how pages look

**JavaScript** A programming language that runs in the browser and provides interaction

**DOM** Document Object Model, which connects the JavaScript code to the page

## Common Tags

**h1** heading tag, for headlines, section headings, chapter titles

**p** paragraph tag, holds basic text

**ol** ordered list tag, for numbered or enumerated lists

**ul** unordered list tag, for bulleted—unnumbered—lists

**li** list items, go inside ol or li tags

**style** CSS code goes between the style tags

**script** JavaScript code goes between the script tags

**a** anchor tag is used to make links

## HTML Template

```
<!doctype html>
<html>
  <head>
  </head>
  <body>
  </body>
</html>
```

## HTML Example

```
<!doctype html>
<html>
  <body>
    <h1>This is a heading</h1>
    <p>This is a paragraph</p>
```

```
</body>
</html>
```

## Basic Selectors

**.name** selection by class

**#name** selection by ID

**name** selection by tag name

## Common CSS Properties

**width** the width of the element

**height** the height of the element

**color** the color of the text

**background-color** the color of the background of the element

**display** how the element is displayed: block and none are two possible values

## CSS Example

```
<!doctype html>
<html>
  <head>
    <style>
      p {
        color: red;
      }
    </style>
  </head>
  <body>
    <h1>This is a heading</h1>
    <p>This is a paragraph</p>
  </body>
</html>
```

## Basic JavaScript Syntax

### Creating Variables

```
var variableName = 20;
variableName;
variableName = 30;
```

### Arithmetic

```
10 + 10;
20 - 5;
30 * 3;
30 / 10;
```

### Strings

```
"this is a string";
'as is this';
"as is 'this'";
'and is "this"';
```

You can mix quotation types for typesetting purposes, but otherwise they're the same.

## Functions

### Creating functions

```
function functionName (x) {
  console.log(x);
  return x + 10;
}
```

### Using functions

```
functionName(10);
console.log("thing");
```

## Objects

```
var myObject = {property1 : value1,
                 property2 : value2};
myObject.property1;
myObject.property2 = 100;
```