

# cs002 Lab 6

## JavaScript

Assigned: November 7,2016

Due: November 18,2016

---

### Introduction

So far, you have learned how to lay out a webpage using HTML, and then style it with CSS. In this lab, you'll learn how to give your webpage interactivity using JavaScript.

### Lab Goals

- Learn how to change HTML using JavaScript
- Learn simple JavaScript events: onclick, onmouseover, onmouseout
- Understand variables and functions
- Learn image rollover and color change

### Lab Assignment

#### Getting Started

**Note: You cannot copy-and-paste from this PDF or else some characters will not copy correctly.**

You will be writing your JavaScript in a plain text editor. Once you have finished, you will open the document in a web browser, which will take care of interpreting what you have written and turn it into an interactive web page.

1. Download the resources for the Javascript lab and extract the files. You should save your work in this folder.
2. Open the text editor of your choice (recommended: Sublime or Notepad++)
3. Once you have your text editor open, save the file as a JavaScript file by saving it as lab6.js. Make sure to save it in the same folder as the resources you extracted in Step 1.
4. Now, you'll need to have an html file to work with. We've provided a template, **homepage.html**. Open this html file.
5. Just like you needed to link your css file to html, you will also need to link your JavaScript file to your html. To do this, add a link between anywhere between the `<head>` `</head>` tag in homepage.html. Remember: **you cannot copy-and-paste from this PDF or else some characters will not copy correctly:**

```
<script src="lab6.js"></script>
```

*\*Make sure your .js and .html files are in the same folder or else your HTML won't be able to find the JavaScript file using the code above!*

Note: The html file would be able to find the JavaScript file if specified the correct relative path (ie: "../lab6.js" would be the path of a JavaScript file located in the directory/folder above your where your HTML file is located).

6. Before adding any JavaScript, you will need to add this to your JavaScript file:

```
window.addEventListener('load', function(){  
    // put all JavaScript code here  
});
```

This makes sure that your code runs after the window loads in your web browser. Another important note is that the // indicates the beginning of a comment. Anything on the same line will not be written as code run by your computer. This is useful to make notes about your code, making it easier to understand.

**Task: For each task, leave a comment above it to explain what your code is doing.**

## **Hello World!**

1. **Add the following line of code into your JavaScript file within the window.addEventListener function:**

```
console.log("Hello World");
```

Your file should now look like the following:

```
window.addEventListener('load', function(){  
    console.log("Hello World");  
    // put the rest of your Javascript code here  
});
```

2. Now, let's look at your web page by right-clicking your html file in the file explorer.

Right-Click File → Open with → Google Chrome.

3. Right-click anywhere in the browser that's not an image → Inspect (Ctrl+ Shift + i)
4. Something should pop up at the bottom or to the right. In the Elements tab, you should be able to see all your html. Note that you can change the html here, which will change what pops up in the browser. However, this will not permanently change your html file.
5. Switch to the Console tab. Here, you should see Hello World printed. The Console tab is useful because if there are errors in your JavaScript, they will show up in the console. You should use this tool to help you figure out what is going on when your code does not function the way you expect it to.

## **JavaScript Display Changes**

### **Changing HTML**

We can actually change our html files using JavaScript using something called `getElementById()`. **Add the following to your JavaScript file under your `console.log`("Hello World"):**

```
document.getElementById("title").innerHTML = "The  
Incredibles";
```

Now you should see “The Incredibles” in the browser instead of “The Parr Family”.

What this code does is go to your HTML, finds the element with the ID “title”, and changes the original text (“The Parr Family”) to “The Incredibles”. Note that capitalization and spacing between the quotes is important. The IDs need to match exactly.

**Task: Make a new HTML element and change its HTML using JavaScript.**

*Hint:* p, h1, and div are all examples of HTML elements. Refer back to the HTML lab for a refresher on creating HTML elements. (*Example:* `<p></p>`)

### **Alert!**

We can also add an alert to our web page. When we open the browser, this alert will pop up. **Add this to your code:**

```
alert ("Syndrome is coming");
```

## JavaScript Events




By using JavaScript, we can make web pages react to ‘events’ that occur on the webpage. This allows for web pages to be more interactive. For example, some events include onclick, onmousemove, and onmouseover.

## Variables

Variables are very important in Computer Science because they store data in a location. When you store this data, you can come back to it and use it later, which can also make doing tasks much more convenient and make your code neater.

## Functions

Functions are used in computer science to perform a specific action/task. We can block out this task from the rest of the code. **Add this to your code:**

<code>function add(x,y){</code>		This is the function definition
<code>    console.log(x+y);</code>		This is the function body
<code>};</code>		
<code>add(3,4);</code>		This is the function call

Check your console tab again, you should see 7 printed there.

## Clicking Event

Let’s start by adding something to the body of our html file. **Add the following to your homepage.html file** (Note: make sure to save after updating the file!):

```
<p id = "click"> Click me! </p>
```

Now, we want to tell the computer that when this element is clicked, the text will change to “You clicked me!”. Let’s break down the best way to do this.

1. Create a variable myClick:

```
var myClick;
```

2. Now, we want to give data for the variable myClick to store.

```
var myClick = document.getElementById("click");
```

What this does is save the “click” element as the variable myClick, so when we want to access this element again, we don’t need to type the whole “document.getElementById(“click”)” out, we can just use myClick.

3. Now how do we tell the computer to change this element when it is clicked?  
JavaScript has a function `onclick`.

```
myClick.onclick =
```

We want what comes after the `=` to be to take the `myClick` element and change its `innerHTML` to be “You clicked me.” We can do this by using a function.

```
myClick.onclick = function(event){  
    myClick.innerHTML = "You clicked me!";  
};
```

This says that when we click on `myClick`, which is an element in your html, we want to run a function. What the function does depends on what you put between the `{ }` brackets. In this particular function, we tell it to change the text of `myClick`.

Note: in javascript the semicolon (`;`) is like a period and signifies the end of a line of code. Make sure to finish each line of code with a semicolon unless it ends with an open bracket (`{ }`).

**Task: Change your code so that when you click `myClick`, there is an alert.**

### Other Cool JavaScript Events

Besides `onclick`, JavaScript has some other cool events you can try. You can use them similar to how we used `onclick` except you will replace `onclick` with the following event name:

`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

**Task: Create a new HTML element and change it when one of the above events occur**

### Image Rollover

Let's try changing an image when we moves the mouse over it and changes back when we move the mouse out.

1. Create a variable that stores the `img` element in your html

```
var image1 = document.getElementById("image");
```

2. Then, on the mouseover event, you want to change the image's src, or its source location, to a different src:

```
image1.onmouseover = function(event){  
    this.src= "img2.jpg";  
};
```

3. Now, the image changes when we move our mouse over it since in the resources we provided we have another image titled "img2.jpg". To change it back to the original when we move the mouse out, the code we write is very similar.

```
image1.onmouseout = function (event){  
    this.src = "img1.jpg";  
};
```

**Task: Choose 2 images of your own and try changing the image on mouseover and changing it back on mouse out.**

### Color Changes

In your HTML template, we created a with the ID "color". Next, we want to change the color of this div using a JavaScript function.

1. Define a **function** to change the background color of any element with an id. As parameters, we will take in the **elementId** of the element we want to change, its **color**, and the **timeout** we will wait until we change the color. It will look something like:

```
function colorChange(elementId, color, timeout){  
    // body of function, put code for function here!  
}
```

2. The code below will set that div's background color to be blue after one second (the number is in milliseconds). Put this code in the body of your function:

```
setTimeout(elementId + ".style.backgroundColor = '" + color  
+ "'" , timeout);
```

What this is doing is it is taking the `elementId` passed into the function and changing its `backgroundColor` to `color` after `timeout` milliseconds have passed.

To call on this function to change the background color of our div, we can write the following code below (outside) the colorChange function we created earlier:

```
colorChange("color", "blue", 1000);
```

This will change the background color of the element with id “color” to blue after 1000 milliseconds.

3. You can continue changing the color by calling on this function with different parameters (you will need to make the next timeout longer than the previous timeout, or else it will skip the previous color).

**Task: Make the div change to at least 3 different colors.**

### Check-off Requirements

- Make an HTML element and change its text using JavaScript
- Create myClick alert function
- Create another HTML element that uses a mouseover, mouseout, or keydown event
- Image rollover on own images
- Color change function, change the div to at least 3 colors
- Have comments above the code for each task explaining which task the code is associated with

### Submission

To submit this lab, please raise your hand so a TA can come check your work. Make sure you tell the TA to check you off their list, or else you will not receive credit. If you do not finish your lab in the allotted time, please either come to office hours or another lab section to finish your work. If you are unable to complete this lab due to sickness or injury, please contact BOTH Don and the cs002 TAs via email. Please contact the cs002 TA's if you have any further questions.

**cs002 TA Email:** [cs002tas@cs.brown.edu](mailto:cs002tas@cs.brown.edu)

**Don Stanford's Email:** [don.stanford@gmail.com](mailto:don.stanford@gmail.com)