COGS3 JavaScript Module!!

JavaScript Codecademy
part A: rock-paper-scissors
part B: add ‘n subtract
part C: slideshow

Put it in: public_html/hw6
Complete Javascript Codecademy Tutorial

COGS3 – Introduction to Computing Assignment
purpose:
Learn JavaScript

important for interactive webpages and HCI/Design and understanding computer programming

• Make an account on Codecademy
• Use your UCSD account: ---@ucsd.edu
• Do not buy anything!! This assignment does NOT require you to purchase the PRO package.

Turning in your assignment:
1. Use the google form link to submit a screenshot of your completed work on Codecademy

• To receive full credit, you must complete 100% of the JavaScript course.

• Start early, this course will take time. Go to the labs/sections/OH to get help... do not leave this to the last minute. 😊

Academic Integrity
• all work is to be completed individually;
• you may work and help each other in the lab but you cannot have someone else “do/think” your work for you (even if you are doing the typing!)
• if you are repeating this class, you cannot re-submit assignments from a previous quarter.
• please review all ACS and UCSD academic integrity policies – you will be tested on them on the midterm and final exams.
• the best way to learn programming is to do it – give yourself the time to explore and enjoy the process. 😊
Complete codecademy JavaScript course.

Complete the JAVASCRIPT tutorial. START EARLY!!!

Create an account if you don’t already have one. Use your UCSD email.

Go to the bottom of the codecademy.org page and select JavaScript.

Use the link below to submit your JAVASCRIPT screenshot.
COGS3 Spring 2018
https://goo.gl/forms/8ha03ydnOQAQzYon2

Updated link: https://www.codecademy.com/learn
Go to your profile page and take the screenshot to validate your completion.

Use the form to upload it and turn it in.
Do NOT pay!
The course is free.

Introduction To JavaScript
Learn the fundamentals of JavaScript, the programming language of the web.

Select this course on the codecademy website.
Complete 9 sections
FROM: INTRODUCTION TO JAVASCRIPT
TO LEARN JAVASCRIPT CLASSES

Do NOT pay for the PRO!

Start here:

1. Learn JavaScript: Introduction
2. Learn JavaScript: Control Flow
3. Learn JavaScript: Functions
4. Learn JavaScript: Scope
5. Learn JavaScript: Arrays
6. Learn JavaScript: Loops
7. Learn JavaScript: Iterators
8. Learn JavaScript: Objects
9. Learn JavaScript: Classes

Start with these two assignments

Start early!
PART A

purpose:

integrate HTML and JavaScript

use your JS knowledge from Codecademy

• Use your favorite text editor
• Download all part B files from the class website (index.html, style.css and style.js)
• save your files in the following format: cg3xzz_RPS.html
• save your css files in the following format: cg3xzz_RPSStyle.css
• save your js file in the following format: cg3xzz_RPS.js
• cg3xzz = your class account

Turning in your assignment:

1. transfer it to the class server acsweb.ucsd.edu
2. save it in your public_html folder in a folder called hw6.

• To receive full credit your file and directory names must follow the exact specifications outlined the instructions.
• Please pay particular attention to:
  • case, hyphens, underscores, and spaces

Academic Integrity

• all work is to be completed individually;
• you may work and help each other in the lab but you cannot have someone else “do/think” your work for you (even if you are doing the typing!)
• if you are repeating this class, you cannot re-submit assignments from a previous quarter.
• please review all ACS and UCSD academic integrity policies – you will be tested on them on the midterm and final exams.
• the best way to learn programming is to do it – give yourself the time to explore and enjoy the process. 😊
The assignment is to first code the game Rock, Paper, Scissors in JS and then to wrap it into HTML.

- Adapted from an earlier version on Codecademy.
- Rock, paper, scissors (RPS) is a classic 2 player game.
- Each player chooses either a rock, paper or scissors.
- The possible outcomes are:
  - Rock destroys scissors.
  - Scissors cut paper.
  - Paper covers rock.

- For your assignment, you will:
  1. Welcome the user to the game and ask them if they want to play RPS.
  2. Prompt the user to make a choice (choose: rock, paper, or scissors).
  3. The computer will make a random choice (rock, paper or scissors).
  4. Use a compare function to determine who wins.
  5. Ask the user if they want to play again or finish the game.
If you run into trouble... you may want to watch these videos to guide you.

(note: there are many ways to get this done.)

https://www.youtube.com/watch?v=uoaMuxPgQ7c

https://www.youtube.com/watch?v=uvfxxehL6j8
PART B
purpose:
integrate HTML and JavaScript
use your JS knowledge from Codecademy

- Use your favorite text editor
- Download all partB files from the class website (index.html, style.css and style.js)
- save your files in the following format: cg3xzz_jsPartB.html
- save your css files in the following format: cg3xzz_jsPartBStyle.css
- save your js file in the following format: cg3xzz_jsPartBStyle.js
- cg3xzz = your class account

Turning in your assignment:
1. transfer it to the class server acsweb.ucsd.edu
2. save it in your public_html folder in a folder in a folder called hw6.

- To receive full credit your file and directory names must follow the exact specifications outlined the instructions.
- Please pay particular attention to:
  - case, hyphens, underscores, and spaces

Academic Integrity
- all work is to be completed individually;
- you may work and help each other in the lab but you cannot have someone else “do/think” your work for you (even if you are doing the typing!)
- if you are repeating this class, you cannot re-submit assignments from a previous quarter.
- please review all ACS and UCSD academic integrity policies – you will be tested on them on the midterm and final exams.
- the best way to learn programming is to do it – give yourself the time to explore and enjoy the process. 😊
PART C

purpose:
integrate HTML and JavaScript

use your JS knowledge from Codecademy

· Use your favorite text editor
· save your files in the following format:
  cg3xzz_banner.html
cg3xzz_slideshow.html
· save your css files in the following format:
cg3xzz_slideshow.css
· cg3xzz = your class account

Turning in your assignment:

1. transfer it to the class server
   acsweb.ucsd.edu
2. save it in your public_html
   folder in a folder in a folder called 
   hw6.

· To receive full credit your file and
directory names must follow the exact
specifications outlined the instructions.
· Please pay particular attention to:
  · case, hyphens, underscores, and spaces

Academic Integrity
· all work is to be completed individually;
· you may work and help each other in
  the lab but you cannot have someone else “do/think” your work for you (even
  if you are doing the typing!)
· if you are repeating this class, you
  cannot re-submit assignments from a
  previous quarter.
· please review all ACS and UCSD
  academic integrity policies – you will be
  tested on them on the midterm and
  final exams.
· the best way to learn programming is
to do it – give yourself the time to
  explore and enjoy the process. 😊
use w3schools.com as your JS, HTML and CSS resource.

Goals of the assignment:
Create a captioned slide show controlled by navigation buttons.
You will learn to do the following in this assignment:

1. Insert JS code into your HTML documents
2. Create and control forms within HTML
3. Use JS to control and manipulate what your page displays

Before you start the assignment you will need to:
1. collect six images with a common theme – eg. vacation, hobbies, movies, books. (you will be happiest with the outcome if the images are of similar size and quality.)
2. create an images directory in your public_html/hw6 directory and place all of the images in it.
3. For each of the images, you will need to write an appropriate caption. Hint: keep notes about which caption goes with which image.

save your files as: cg3xzz_banner.html, cg3xzz_slideshow.html, and cg3xzz_slideshow.css
Note:

Please read:

- It is great to be creative, but if there is anything that is rude or inappropriate, you will get zero for this assignment.
1. save 6 image files into a folder – note images can be .png, .jpg, .gif ... (called: “images”)
2. start small – get one image to display on your webpage:

```html
<html>
  <head>
    <title>Banner</title>
  </head>
  <body>
    <div align="center">
      <img name="banner" src="images/failure-is-not-falling-down-proverb.gif">
    </div>
  </body>
</html>
```
3. read through the code: (see next slides for larger images!)

4. to **understand** the code read the w3Schools documentation & actually **do** the “try-it” examples:
5. using your six images create your own “banner” page.

You can use the code provided in the assignment 😊

However, don’t just copy without understanding.

Change the parameters and see what happens.
6. Create an interactive slideshow ---

required elements for the slideshow:
• buttons to control the slides (e.g. next, previous, automatic)
• current slide number
• image and caption
• consistent image size

extra credit:
• “fancy” formatting using a linked stylesheet
• going over and above the minimum requirements
<html>
<head>
<title>Banner</title>
<script>
var imgArray = new Array();
var index = 0;
function cycle()
{
    document.banner.src = imgArray[index].src;
    index++;
    if (index > 5)
    {
        index = 0;
    }
    setTimeout("cycle()", 3000);
    return;
}
function startup()
{
    imgArray[0] = new Image;
    imgArray[1] = new Image;

understand this code!
(Banner code)
```html
imgArray[0].src = "images/failure-is-not-falling-down-proverb.gif";
imgArray[1].src = "images/if-you-cant-fly-mlk.gif";
imgArray[2].src = "images/imagination-is-more-important-than-knowledge-einstein.gif";
imgArray[3].src = "images/ive-failed-over-and-over-jordan.gif";
imgArray[4].src = "images/trust-yourself-spock.gif";
imgArray[5].src = "images/you-dont-have-to-be-great-to-start-zagler.gif";

cycle(); return;
```

```html
<body onload="startup()">
  <div align="center">
    <img name="banner" src="images/failure-is-not-falling-down-proverb.gif">
  </div>
</body>
</html>
```
The Array object is used to store multiple values in a single variable.

Try it Yourself - Example

Create an array, and assign values to it:

Example

```javascript
var mycars = new Array();
mycars[0] = "Saab";
mycars[1] = "Volvo";
mycars[2] = "BMW";
```

Try it yourself »
```html
<!DOCTYPE html>
<html>
<body>

<script>
var i;
var mycars = new Array();
mycars[0] = "Saab";
mycars[1] = "Volvo";
mycars[2] = "BMW";

for (i=0;i<mycars.length;i++)
{
document.write(mycars[i] + "<br>");
}
</script>

</body>
</html>
```
Image Object

The Image object represents an embedded image.

For each `<img>` tag in an HTML document, an Image object is created.

Notice that images are not technically inserted into an HTML page, images are linked to HTML pages. The `<img>` tag creates a holding space for the referenced image.
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>align</td>
<td>Not supported in HTML5. Use style.cssFloat instead. Sets or returns the value of the align attribute of an image</td>
</tr>
<tr>
<td>alt</td>
<td>Sets or returns the value of the alt attribute of an image</td>
</tr>
<tr>
<td>border</td>
<td>Not supported in HTML5. Use style.border instead. Sets or returns the value of the border attribute of an image</td>
</tr>
<tr>
<td>complete</td>
<td>Returns whether or not the browser is finished loading an image</td>
</tr>
<tr>
<td>height</td>
<td>Sets or returns the value of the height attribute of an image</td>
</tr>
<tr>
<td>hspace</td>
<td>Not supported in HTML5. Use style.margin instead. Sets or returns the value of the hspace attribute of an image</td>
</tr>
<tr>
<td>longDesc</td>
<td>Not supported in HTML5. Sets or returns the value of the longdesc attribute of an image</td>
</tr>
<tr>
<td>name</td>
<td>Not supported in HTML5. Use id instead. Sets or returns the value of the name attribute of an image</td>
</tr>
<tr>
<td>src</td>
<td>Sets or returns the value of the src attribute of an image</td>
</tr>
<tr>
<td>useMap</td>
<td>Sets or returns the value of the usemap attribute of an image</td>
</tr>
<tr>
<td>vspace</td>
<td>Not supported in HTML5. Use style.margin instead. Sets or returns the value of the vspace attribute of an image</td>
</tr>
<tr>
<td>width</td>
<td>Sets or returns the value of the width attribute of an image</td>
</tr>
</tbody>
</table>
Image width Property

Example
Return the width of an image:

```javascript
var x = document.getElementById("myImg").width;
```

The result of `x` will be:

107

Try it yourself »

Definition and Usage
The width property sets or returns the value of the width attribute of an image.
The width attribute specifies the width of an image.
Tip: Use the `height` property to set or return the value of the height attribute of an image.

http://www.w3schools.com/jsref/prop_img_width.asp
Source Code:

```html
<!DOCTYPE html>
<html>
<body>
<img id="myImg" src="compman.gif" width="107" height="98">
<p>Click the button to return the width of the image.</p>
<p id="demo"></p>
<button onclick="myFunction()">Try it</button>
<script>
function myFunction() {
  var x = document.getElementById("myImg").width;
  document.getElementById("demo").innerHTML=x;
}
</script>
</body>
</html>
```

Result:

Click the button to return the width of the image.

Try it
**Window setTimeout() Method**

**Example**

Display an alert box after 3 seconds:

```javascript
setTimeout(function(){alert("Hello")},3000);
```

**Definition and Usage**

The `setTimeout()` method calls a function or evaluates an expression after a specified number of milliseconds.

**Tip:** 1000 ms = 1 second.

**Tip:** The function is only executed once. If you need to repeat execution, use the `setInterval()` method.

**Tip:** Use the `clearTimeout()` method to prevent the function to run.

http://www.w3schools.com/jsref/met_win_settimeout.asp
setTimeout(function, milliseconds, lang)

### Parameter Values

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>function</td>
<td>Required. The function that will be executed</td>
</tr>
<tr>
<td>milliseconds</td>
<td>Required. The number of milliseconds to wait before executing the code</td>
</tr>
<tr>
<td>lang</td>
<td>Optional. The scripting language: JScript</td>
</tr>
</tbody>
</table>

### Return Value

An integer with the ID value of the timer that is set. Use this value with the clearTimeout() method to cancel the timer.

### More Examples

**Example**

Display a timed text:

```javascript
var x = document.getElementById("txt");
setTimeout(function(){x.value="2 seconds"},2000);
setTimeout(function(){x.value="4 seconds"},4000);
setTimeout(function(){x.value="6 seconds"},6000);
```
Click on the button below. The input field will tell you when two, four, and six seconds have passed.

Display timed text

6 seconds
Open a new window and close the window after three seconds (3000 milliseconds):

```
var myWindow = window.open("","","width=200,height=100");
myWindow.document.write("<p>This is 'myWindow'</p>");
setTimeout(function(){myWindow.close()},3000);
```

Source Code:
```
<!DOCTYPE html>
<html>
<body>

<p>Click the button to open a new window and close the window after three seconds (3000 milliseconds)</p>

<button onclick="openWin()">Open "myWindow"</button>

<script>
function openWin()
{
 var myWindow = window.open("","myWindow","width=200,height=100");
 myWindow.document.write("<p>This is 'myWindow'</p>");
 setTimeout(function(){myWindow.close()},3000);
}
</script>

</body>
</html>
```
Window `setInterval()` Method

**Window Object**

**Example**
Alert "Hello" every 3 seconds (3000 milliseconds):

```javascript
setInterval(function() {alert("Hello"), 3000);
```

**Definition and Usage**
The `setInterval()` method calls a function or evaluates an expression at specified intervals (in milliseconds).
The `setInterval()` method will continue calling the function until `clearInterval()` is called, or the window is closed.
The ID value returned by `setInterval()` is used as the parameter for the `clearInterval()` method.

**Tip:** 1000 ms = 1 second.

**Tip:** To execute a function only once, after a specified number of milliseconds, use the `setTimeout()` method.

http://www.w3schools.com/jsref/met_win_setinterval.asp
onclick Event

Event Object

Example
Execute a JavaScript when a button is clicked:

```
<button onclick="myFunction()">Click me</button>
```

Definition and Usage
The onclick event occurs when the user clicks on an element.

Syntax
In HTML:

```
<element onclick="SomeJavaScriptCode">
```
Syntax

In HTML:

```html
<element onclick="SomeJavaScriptCode"> 
```

In JavaScript:

```javascript
object.onclick=function(){SomeJavaScriptCode};
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SomeJavaScriptCode</td>
<td>Required. Specifies a JavaScript to be executed when the event occurs</td>
</tr>
</tbody>
</table>

Browser Support

The onclick event is supported in all major browsers.

onclick In HTML

The onclick attribute can be used within ALL HTML elements, EXCEPT: `<base>`, `<bdo>`, `<br>`, `<head>`, `<html>`, `<iframe>`, `<meta>`, `<param>`, `<script>`, `<style>`, and `<title>`. 

Event Object
<DOCTYPE html>
<html>
<head>
    <script>
        function myFunction()
        {
            document.getElementById("demo").innerHTML="Hello World";
        }
    </script>
</head>
<body>
    Click the button to trigger a function.
    <button onclick="myFunction()">Click me</button>
    <p id="demo">Click me</p>
</body>
</html>