# Lab 7

# JavaScript Functions, Objects, and the DOM

80 points

Purpose/Knowledge/Skills:

Breaking complicated problems into smaller pieces is one of the core skills of professional programmers. Only simple programs can be written as one single piece. Larger programs are broken up into chunks called functions. In this lab, you’ll learn how to call existing functions and create your own.  
  
Most programs model real-world objects in some capacity. Programming objects allow you to combine data and code related to a particular type of real-world thing. Programmers use objects extensively in their code.

Most JavaScript programs alter the displayed HTML page. The Document Object Model (DOM) is one way to add, change, or remove existing HTML elements.

Task 1: Calling functions (5 points)

Create an HTML + JS page from the following code. Please do not copy and paste; Word loves to mangle source code in subtle ways.

<html>

<!-- Student Name Today's Date -->

<head>

<script>

function showAlert(text) {

alert(text);

}

</script>

</head>

<body>

<!-- Demonstrates several ways to call a function -->

<script>

showAlert("This alert came from a script tag");

</script>

<input type="button" value="Button" onClick="showAlert('Called from an event!')" />

<br />

<a href="JavaScript:showAlert('Called from a link')">Click here to show an alert</a>

<!-- When requested, add another script tag down here -->

</body>

</html>

When this works as expected, add code to:

1. Call the function showAlert to display your name when a button is pressed.

2. Add a second call from new <script> tag that says “This alert came from XXXXX’s script tag.”. Replace XXXXXX with your name.

Rubric:

* Student name and date is in a comment on the first line of the program: -5 points if fails
* Original code works: 1 point
* Additional function calls: 2 points each

Please paste a screenshot of a successful program run, and copy-and-paste the source code from your .html and/or .js files here:

## Task 2: w3schools JavaScript review (5 points)

Please read the JavaScript tutorial at w3schools at <https://www.w3schools.com/js/default.asp>. Complete the following sections and paste a screenshot of one “try it yourself” or “Exercise” activity from each.

JS Output – using InnerHTML

Paste a screenshot of a completed “Try It Yourself” or “Exercise” activity here:

JS Functions – all four exercises

Paste a screenshot of all four “Exercise” activities here:   
(One exercise per screenshot, for a total of four screenshots.)

Rubric:

* JS Output -> InnerHTML try it now: 1 points
* JS Functions four screenshots, one per exercise: 1 point each

## Task 3: Idle game questions (10 points).

Please read the additional “Setting and Using Timers” reading for this week before you continue with this task.

An idle clicker game is included in the Lab download for Canvas this week.

Please play the game, review the JavaScript code, and answer the following questions:

3.1. Which of the following are global variables? (Choose four)

ticks span updateUI cookies bakers superBakers clockTick

3.2. Which of the following are functions? (Choose two)

ticks span updateUI cookies bakers superBakers clockTick

3.3. When setInterval is called, the browser begins to call a function repeatedly. How does the browser know how long to wait between function calls?

3.4. In your own words, what is purpose of updateUI?

3.5. In your own words, what is the purpose of clockTick?

Rubric:

* Five questions, 2 point each

Task 4: Expand the idle clicker game from the previous question. (20 points)

4.1. Choose enhancements to the program.

You can include any grouping of options up to a maximum of 20 points.  
For example, if you code 4 options with 5 points each, you’ll earn a full 20 points for this problem.

4.1.1. Calculate and show the current cookies per second made by bakers and super bakers.   
(5 points)

4.1.2. Add a button to stop the clock ticks. (5 points)  
Hint: <https://stackoverflow.com/questions/109086/stop-setinterval-call-in-javascript>

4.1.3. Starting from 5-2, toggle the button so a click will start the clock after it was stopped. (5 points)

4.1.4. Bakers make a cookie every ½ second (500 milliseconds). (5 points)

4.1.5. Add a button, variables, and calculations for MegaBakers. MegaBakers cost 150 cookies and bake 15 cookies every 3 seconds. (10 points)

4.1.6. Add a progress bar that shows how close the user is to having a full jar of 1000 cookies. (10 points)

4.1.7. Ticks happen 10 times a second (100 millisecond delay), with  
all values and calculations updated so the program works as before (i.e. one cookie per second for bakers) (15 points)

4.1.8. Other improvements of your choice, including styling enhancements based on the current number of cookies, bakers, clock ticks, etc. (10 points)

4.3. Code your enhancements.

4.4. Put your name and today’s date in a comment at the top of the program.

4.5. Note what you've done in comments at the top of the program, for ease of grading.  
 Examples:

<!-- Michael Haensel 4/3/2035 -- >

<!-- I implemented show cookies per second, stop/start clock ticks, and changing the bakers to a cookie every 1/2 second -->

or

<!-- Michael Haensel 4/3/2035 -- >

<!-- I added MegaBakers, the current cookies per second, and the number of cookies turns neon green when it's over 500 -->

Rubric:

* Student name and date is in a comment on the first line of the program: -20 points if fails
* Improvements called out in a comment at the top of the program: -20 points if fails
* Improvements per problem statement: 20 points maximum

Please paste a screenshot of a successful program run, and copy-and-paste the source code from your .html and/or .js files here:

## Task 5: Create Objects and Properties in JavaScript (10 points)

Create an HTML + JS page, or use repl.it, to run the following code. Please remember to type this in rather than copy-and-paste; Word loves to mangle source code.

// Student name Today's date

c1 = {

"radius": 25,

"color": "blue",

"round": true

}

console.log(c1.radius);

console.log(c1.color);

console.log(c1.round);

When this works, please make the following changes:

5.1. Create a new property called “units” with the value “inches”. You can either create the property inside the object definition, or assign a value to the new property after the object has been created.

5.2. Create a new property called “fuzzy” with the value “true”.

5.3. Output these two new properties to the console.

5.4. Create a **second** circle object called c2, with the following property values. You do not have to output anything from c2.

|  |  |
| --- | --- |
| c2 | |
| Property | Value |
| radius | 100 |
| color | purple |
| round | false |
| units | centimeters |
| fuzzy | false |

Rubric:

* Student name and today’s date is a comment near the top of the program: -3 points if fails
* Source code, screenshot, and output of a successful program run: 2 points
* Two new properties: 3 points
* Output: 3 points
* Create a second circle: 2 points

Please paste a screenshot of a successful program run, and copy-and-paste the source code from your .html and/or .js files here:

## Task 6: Create reusable objects (5 points)

6.1. Create an HTML + JS page, or use repl.it, to run the following code. Please remember to retype this in rather than copy-and-paste; Word loves to mangle source code.

// Student name Today's date

Circle = function(r) {

this.radius = isNaN(r) ? 0 : r;

this.getArea = function() {

return Math.PI \* this.radius \* this.radius;

};

// Define this.getPerimeter when requested

}

// Once this works, delete the following lines

c1 = new Circle();

c1.radius = 15;

console.log(c1.getArea());

c2 = new Circle(25);

console.log(c2.getArea());

What this works, please make the following changes:

6.2. Delete the code that works with c1 and c2.

6.3. Create code to instantiate three circles, where “XXX” is your initials:

|  |  |
| --- | --- |
| Circle | Radius |
| c1XXX | 30 |
| c2XXX | 50 |
| c3XXX | 150 |

6.4. Add a function getPerimeter() to the definition of the Circle object.   
getPerimeter() calculates Math.PI \* 2 \* this.radius to return the perimeter.

6.5. Output the area and perimeter for each circle.  
As a check, your results could look like:  
  
  
Feel free to use rounding to make the results neater if you’d like.

Rubric:

* Student name and today’s date is a comment near the top of the program: -3 points if fails
* Circles have student initials in their variable names: -10 points if fails
* Source code, screenshot, and output of a successful program run: 1 points
* Instantiate three circles: 2 points
* getPerimeter() definition: 2 points
* Output for three circles: 1 points

Please paste a screenshot of a successful program run, and copy-and-paste the source code from your .html and/or .js files here:

## Task 7: w3schools JavaScript Objects and Document Object Model (DOM) review (5 points)

Please read the JavaScript tutorial at w3schools at <https://www.w3schools.com/js/default.asp>. Complete the following sections and paste a screenshot of one “try it yourself” or “Exercise” activity from each.

JS Objects

Paste a screenshot of a completed “Try It Yourself” or “Exercise” activity here:

JS JSON -> JSON Intro

Paste a screenshot of a completed “Try It Yourself” or “Exercise” activity here:

JS HTML DOM -> DOM Intro, DOM Methods, DOM Document, DOM Elements

Please read all four sections, but several don’t have activities so there’s nothing to screenshot. Paste a screenshot of **one** completed “Try It Yourself” or “Exercise” activity here:

JS HTML DOM -> DOM HTML

Paste a screenshot of a completed “Try It Yourself” or “Exercise” activity here:

JS HTML DOM -> DOM CSS

Paste a screenshot of a completed “Try It Yourself” or “Exercise” activity here:

Rubric:

* Five screenshots, one per section: 1 point each

## Task 8: Modify the DOM with JavaScript (20 points)

Create an HTML + JS page to run the following code. Please remember to retype this in rather than copy-and-paste; Word loves to mangle source code.

<!DOCTYPE html>

<!-- Student name Today's date -->

<!-- Change 1: -->

<!-- Change 2: -->

<!-- Change 3: -->

<!-- Change 4: -->

<html>

<body>

<table border="1">

<tr id="row1">

<td id="dog1">Dog</td>

<td id="dog2">Harley</td>

</tr>

<tr id="row2">

<td id="cat1">Cat</td>

<td id="cat2">Mittens</td>

</tr>

<tr id="row3">

<td id="bird1" onmouseover="birdMouseOver()">Bird</td>

<td id="bird2">Paco</td>

</tr>

</table>

<script>

function birdMouseOver() {

document.getElementById("bird1").style.fontSize = "40px";

};

document.getElementById("row1").style.color = "blue";

document.getElementById("dog1").style.fontFamily = "Arial";

</script>

</body>

</html>

This works, but the page is super ugly.   
Please make at least four improvements the web page styling **without changing the HTML code that creates the table.****Except for option 8.8, your improvements must be done via JavaScript.**   
List your improvements in a comment near the top of the program for ease of grading.

Some possible improvements:

8.1. Change the background color of the rows of the table.  
8.2. Change the text formatting for the first cell of each row.  
8.3. The “Bird” cell changes text size onmouseover, but stays stuck at the larger size even after the mouse leaves the cell. Add another event handler to fix this.  
8.4. Add event handlers to change text color on mouseover or click.  
8.5. Use a WebFont.for some or all of the table cells.  
8.6. Change the table border style.  
8.7. Add a background image to the page.  
8.8. Play a video in the background of the page using something like <https://www.w3schools.com/howto/howto_css_fullscreen_video.asp> . (You will have to edit the HTML and possibly add CSS to make this work.)  
8.9 You are not limited to this list! Feel free to make other style changes as desired. Have fun with this – make it look nice!

Please note: each type of change counts as one for grading purposes. For example: changing the background color of one row, or all rows, or setting each row to a different background color, would all be considered one change for grading.

Changing fontSize and fontColor would count as two changes.

Changing the default background color, and then changing the background color again on mouseover, would count as two changes.

Rubric:

* Four types of changes: 5 points each
* Student name and date is a comment near the top of the program: -5 points if fails

Please paste a screenshot of a successful program run, and copy-and-paste the source code from your HTML and/or JavaScript files here: