


Module 3: Assignment

[Start Assignment](#)

Due Dec 22 by 11:59pm **Points** 39 **Submitting** a file upload **File Types** zip

Assignment 3: Tax Calculator

 "Welcome! In this assignment, you will be creating a tax calculator app. You will have the opportunity to work with functions and event handlers."

Assignment Steps:

1.  Download and extract [assignment03_starter.zip](https://canvas.mccneb.edu/courses/46727/files/7176226?wrap=1)
(<https://canvas.mccneb.edu/courses/46727/files/7176226?wrap=1>) 
(https://canvas.mccneb.edu/courses/46727/files/7176226/download?download_frd=1)
2.  Review the [JavaScript Style Guide](https://canvas.mccneb.edu/courses/46727/files/7176260?wrap=1) (<https://canvas.mccneb.edu/courses/46727/files/7176260?wrap=1>).

Test the application

1. Double-click **index.html** to launch it in your browser; examine the application's behavior. Click on **Calculate Taxes**. Nothing will happen.

Tax Me!

Monthly Salary:

[Calculate Taxes](#)

Modify the application (Part 1)

1. **script.js** has already been linked to **index.html** for you.
2. Add the **Standard Opening Comment** to the top of the script (**-1 if not included or lines missing**).
3. Add **"use strict";** to the top of the script (and below the opening comment) of the JavaScript file (**-1 if not included**).
4. Beneath the comment **//YOUR CODE GOES BELOW** accomplish the following:
 - A. Use the **arrow** syntax to define a function named **\$** (**10 points**)
 - I. This function should accept one parameter, which represents a "selector" and it should return the return value of **document.querySelector(x)** – where **x** is the name of the parameter

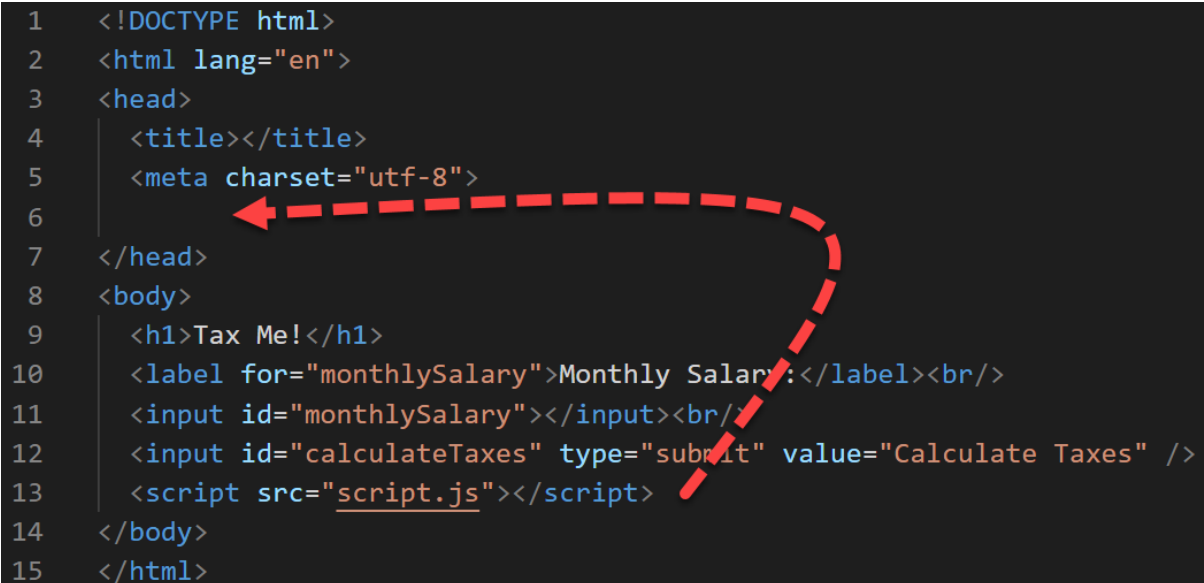
representing the “selector.”

- B. Define an event handler that responds to a **click** event generated by the **input** element that has an **id** of **calculateTaxes**. (5 points)
 - I. The event handler should obtain the value in the text input field with the **id** of **monthlySalary**, which is defined in **index.html** (5 points)
 - II. The event handler should calculate the federal, state, social security, and Medicare tax rates by invoking the **calculateTax** function. Use the following tax rates for this exercise (these aren't real): (8 points)
 - Federal Taxes: 12%
 - State Taxes: 5%
 - Social Security Taxes: 6%
 - Medicare: 1.5%
- C. Calculate the **net** salary by subtracting each of the resulting tax values from the original **monthlySalary**. (2 points).
- D. Invoke the **print** function (2 points).
- E. Once you've accomplished the above steps, run the html file to test the app. Your program should work as shown in the sample output (see below).

Modify the application (Part 2)

1. Open **index.html**.
2. **Move** the existing `<script>` tag from line 13 to line 6:

o



```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <title></title>
5      <meta charset="utf-8">
6
7  </head>
8  <body>
9      <h1>Tax Me!</h1>
10     <label for="monthlySalary">Monthly Salary:</label><br/>
11     <input id="monthlySalary"></input><br/>
12     <input id="calculateTaxes" type="submit" value="Calculate Taxes" />
13     <script src="script.js"></script>
14 </body>
15 </html>
  
```

3. The result should look like the image below. **Keep your script in the <head> for the remaining steps**

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <title></title>
5    <meta charset="utf-8">
6    <script src="script.js"></script>
7  </head>
8  <body>
9    <h1>Tax Me!</h1>
10   <label for="monthlySalary">Monthly Salary:</label><br/>
11   <input id="monthlySalary"></input><br/>
12   <input id="calculateTaxes" type="submit" value="Calculate Taxes" />
13 </body>
14 </html>
```

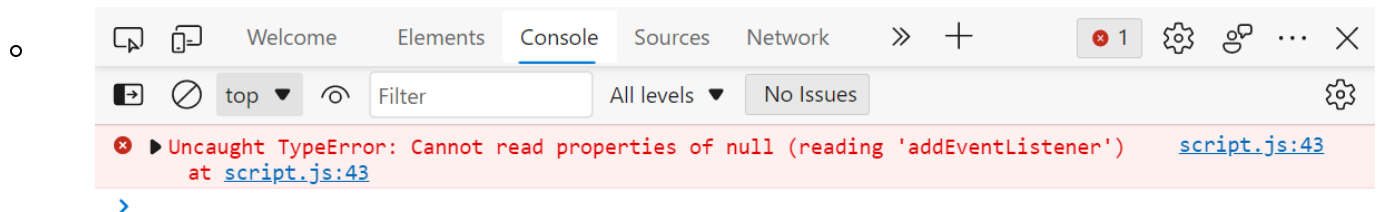
- o 🤖, "Remember, JavaScript will execute as soon as it's loaded into the browser. Up to this point, we have placed our JavaScript at the bottom of our HTML document to ensure all page elements were loaded before the JavaScript loaded and executed."
- o 🤖, "By moving our <script> tag from the bottom of the HTML document to the top of the HTML document, we're going to create problems, because our script will now execute before all page elements have loaded."

4. Save all open files (**html** and **script.js**). Then, launch **index.html** and your browser and test by inputting a salary value of **50** and clicking **Calculate Taxes**. You should get an error in the browser console:

Tax Me!

Monthly Salary:

Calculate Taxes



- o 🤖, "The error appeared because the **addEventListener()** method was called on an element **before** the element loaded in the browser (since we moved **script.js**)."

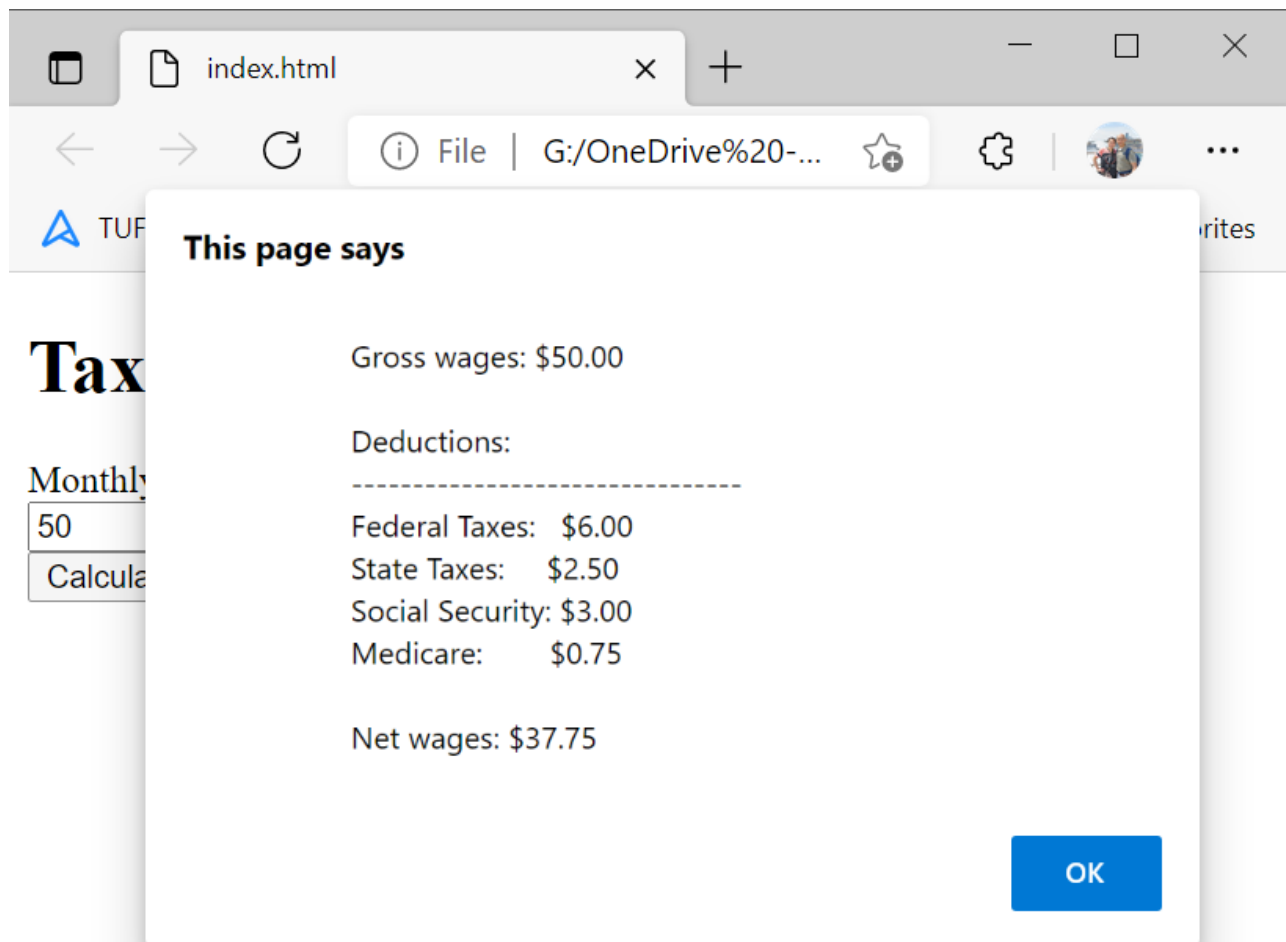
5. Refactor **script.js**: Add an event listener to handle the **document** object's **DOMContentLoaded** (2 points)
6. Refactor **script.js**: Move code that must wait to execute until the HTML has been loaded into the browser and the DOM is ready into the event handler for the **DOMContentLoaded** (2 points)

7. Once you've accomplished the above steps, run the html file to test the app. Your program should work as shown in the sample output (see below).



Wrapping Up

1. Make sure only external JS is present **(-3 points if inline JS is used)**.
2. Double check that you did not use the **var** keyword for declaring variables **(-10 points if var is used)**.
3. Make sure your JavaScript is clean, consistent, and organized. Check the [JavaScript Style Guide \(https://canvas.mccneb.edu/courses/46727/files/7176260?wrap=1\)](https://canvas.mccneb.edu/courses/46727/files/7176260?wrap=1) for guidance on writing clean and consistent code **(3 points)**
4. Add comments to your code to explain *your* understanding of what's going on: The comments must be substantive (meaningful) – when creating your comments, consider explaining the selected aspect of the application to someone who has absolutely no idea what this code does. **(-2 points if no comments or lack thereof)**

Sample Output



Upload Instructions:

1.  Zip your assignment
2.  Upload and submit the zipped assignment.
 - If you make a mistake, you may resubmit the attachment(s) before the assignment due date (there is no need to contact the instructor to do this).

Module 3 Assignment

Criteria	Ratings		Pts
<p>Step 4 A</p> <p>Use the arrow syntax to define a function named \$. This function should accept one parameter, which represents a “selector” and it should return the return value of document.querySelector(x) – where x is the name of the parameter representing the “selector.”</p>	10 pts Full Marks	0 pts No Marks	10 pts
<p>Step 4 B</p> <p>Define an event handler that responds to a click event generated by the input element that has an id of calculateTaxes.</p>	5 pts Full Marks	0 pts No Marks	5 pts
<p>Step 4 B, I</p> <p>The event handler should obtain the value in the text input field with the id of monthlySalary, which is defined in index.html</p>	5 pts Full Marks	0 pts No Marks	5 pts
<p>Step 4 B, II</p> <p>The event handler should calculate the federal, state, social security, and Medicare tax rates by invoking the calculateTax function (2 points each)</p>	8 pts Full Marks	0 pts No Marks	8 pts
<p>Step 4 C</p> <p>Calculate the net salary by subtracting each of the resulting tax values from the original monthlySalary.</p>	2 pts Full Marks	0 pts No Marks	2 pts
<p>Step 4 D</p> <p>Invoke the print function</p>	2 pts Full Marks	0 pts No Marks	2 pts
<p>Part 2: Step 5</p> <p>Refactor script.js: Add an event listener to handle the document object's DOMContentLoaded</p>	2 pts Full Marks	0 pts No Marks	2 pts
<p>Part 2: Step 6</p> <p>Refactor script.js: Move code that must wait to execute until the HTML has been loaded into the browser and the DOM is ready into the event handler for the DOMContentLoaded</p>	2 pts Full Marks	0 pts No Marks	2 pts
<p>Use Strict</p> <p>-1 point if "Use Strict" is not included</p>	0 pts Full Marks	0 pts No Marks	0 pts

Criteria	Ratings		Pts
<p>External JS</p> <p>-3 points if any JS is inline/not external</p>	<p>0 pts Full Marks</p>	<p>0 pts No Marks</p>	<p>0 pts</p>
<p>Standard Opening Comment</p> <p>All five lines of the Standard Opening Comment are included. -1 missing lines or not included</p>	<p>0 pts Full Marks</p>	<p>0 pts No Marks</p>	<p>0 pts</p>
<p>Clean & Consistent Code</p> <p>JavaScript code is clean, consistent, and organized. Refer to the course's JavaScript Style guide. Subtract points if code is not clean and consistent</p>	<p>3 pts Full Marks</p>	<p>0 pts No Marks</p>	<p>3 pts</p>
<p>Comments</p> <p>Include one substantive and meaningful comment to document/explain an aspect of the assignment. The comment should be written in a way that another coder can understand your code. -2 if no comments are included, or not substantial.</p>	<p>0 pts Full Marks</p>	<p>0 pts No Marks</p>	<p>0 pts</p>
<p>var keyword avoided</p> <p>-10 points if var is used anywhere in the program</p>	<p>0 pts Full Marks</p>	<p>0 pts No Marks</p>	<p>0 pts</p>
Total Points: 39			