🌟Assignment 2: A Basic Calculator

Submit Assignment

Due Feb 28 by 11:59pm **Points** 65 **Submitting** a file upload **File Types** zip
Available after Feb 19 at 8:54am

Task: Complete the JavaScript code for a basic calculator in 3 different ways.

Purpose: To assess that you understand how to write inline and external JavaScript programs that respond appropriately to simple user click events and modify the Document Object Model using functions, switch statements, and if-else statements.

Overview

For this Assignment, your group will add JavaScript functionality to a basic calculator that has already been designed for you in HTML.
We will use the **Pair Programming** **(https://en.wikipedia.org/wiki/Pair_programming)** agile collaborative industry practice because software engineers who have team-working skills are more valuable to their organization.
Use ES5 JavaScript for this assignment.
Keep in mind that the completed 3 calculators will all work the same way from the user's perspective and they will not have all of the behaviors and features you get from most modern calculators. **All 3 versions of calculators *must* behave exactly like the calculator in this demonstration video (http://edutek.net/foothill/cs/22a/A2/) .**

This assignment will be *relatively easy*, although it does have some 'busy" work built into it since you are required to do the same thing *three different ways* (of course, there's a method to my madness as this will help you get comfortable using inline JavaScript, an external JavaScript file, creating functions, and working with both if...else if...else and switch elements).

Throughout the project, you will find helpful tips from this **InlineExternalExamples.zip** example archive file.

You will complete the same calculator functionality in 3 different ways:

1. First, by writing completely "**inline**" JavaScript inside the **ABasicCalculator-inline.html** file.
2. Second, by creating a function named **calcu** on an external .js file called **switch.js** that is comprised of a **switch** statement and using some inline JavaScript to call calcu() from the **ABasicCalculator-swtch.html** file.

3. Third, by putting all of your JavaScript in an external .js file called **ifelse.js** (including click event binding code and a function named **calcu** that has an **if...else if...else** statement).

Why bother with External JavaScript files?

Read: ***"Why inline CSS and JavaScript code is such a bad thing"*** **(http://robertnyman.com/2008/11/20/why-inline-css-and-javascript-code-is-such-a-bad-thing/)** .

Why can't we do this assignment using the eval function?

This assignment could have been done using the eval function, but because "eval is ***evil"***

, we are using the **Math.js** **(http://mathjs.org/)** JavaScript math library and calling that library's math.eval() function instead.

Read ***"eval() isn’t evil, just misunderstood"*** ***(http://www.nczonline.net/blog/2013/06/25/eval-isnt-evil-just-misunderstood/)*** .

WARNING:

Do **not** use the **bind()** **(https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Function/bind)** method. It is **not** for setting up event handling.

For HTML elements that are within a form element **(https://www.w3schools.com/html/html_forms.asp)** **that each have a name attribute:** **(https://www.w3schools.com/tags/att_name.asp)**

You can reference them in JavaScript like this: **theNameOfTheForm.theNameOfTheFormElement** Thus, you can access parts of the **calc** form, such as the calculator's display value, like this: **calc.output.value**

Questions:

If you have questions, please review the syllabus and/or contact me, the instructor.

Group Work

- You are required to work in a team (for which Canvas uses the term 'group') for this assignment.
- Before the first team meeting:
  - Make sure you have contacted your group members' via the Canvas Inbox and planned to meet at least weekly for at least 2 hours. **If you have any problems contact me *immediately!* Requests for changing group membership received after Tuesday during Week 7 will likely NOT be approved.** Each team has a team page to share files and info that

can be found by selecting the "Groups" Canvas navigation link.
   - Make sure that you and your team has read and understood the web pages and referenced libraries/plugins that are mentioned in this assignment and in code comments **before** you all start working together on completing the code. That will help you all get the assignment done faster and with higher quality.
- Your team should start, finish, and do **all** of the work together using the Pair Programming agile practice of taking about 20 minute role turns.
- Work together via teleconferencing.
- Do ***not*** split up the work!
- Remember, copying and slightly modifying another person's work, is **plagiarism** and is not acceptable. You can get ideas from students in other teams, but your team must:

Type Your Own Code!

- **Online pair-programming software:** Since we are using pair-programming in our teams, it is best that you use a pair-programming plugin instead, or in addition to, the Zoom teleconferencing software linked from Canvas. Below are a few good options (but there are others).
   - For Brackets or over the Web (https://webdesign.tutsplus.com/articles/real-time-code-collaboration-tools-for-developers--cms-30494) : See the **second** section
   - Teletype for Atom (https://teletype.atom.io/)
   - For Nodeclipse & IntelliJ (https://marketplace.eclipse.org/content/saros-distributed-collaborative-editing-and-pair-programming) (Don't forget to follow the instructions on the "Do: Getting Saros to Work" page.)

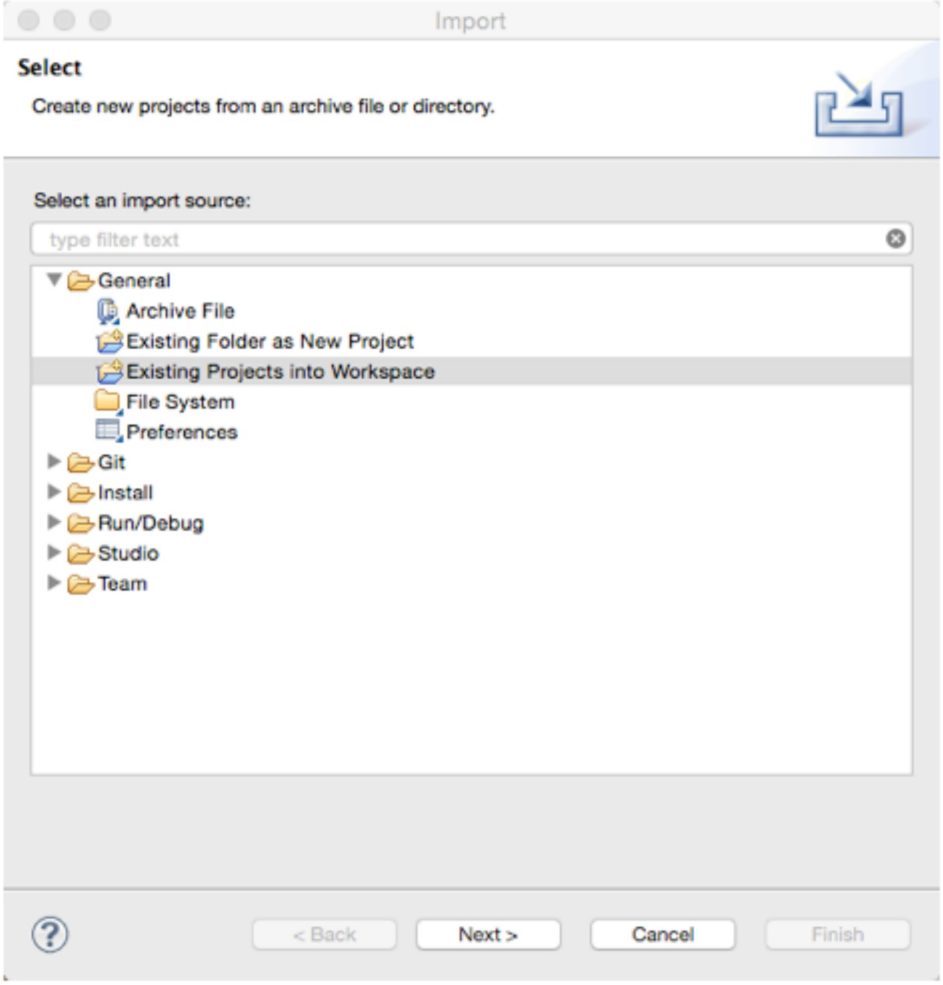
Instructions

Please read is entire page (including the **Overview** and **Group Work** sections above) **before starting your work.**

Step 0: Setting Up the A Basic Calculator Project

1. Download the ABasicCalculator.zip project archive file
2. **Extract** it to where you will be doing your work.
   - It's also a good idea to print out this long page of instructions.
   - If you are using the **Brackets** IDE, you can use in the File menu the "Open folder..." command to open the extracted ABasicCalculator folder.
   - **Make sure your team uses Brackets' Live Preview** (which will be used to evaluate your work) **and then tests the project over the World Wide Web HTTP protocol** (when the URL will start with "**http:**"). For instructions for how to use Live Preview review the page: How to use the Live Preview feature in the Bracket.io IDE
   - If you are using the **Nodeclipse** IDE, choose from the File menu “Import” and then use the

General option called “Existing Projects into Workspace” (shown below) to import the folder you extracted.



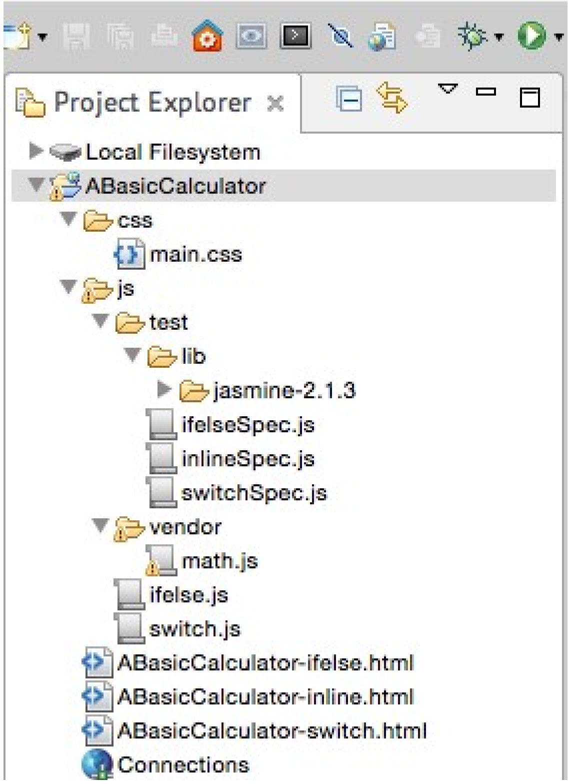
Import

Select

Create new projects from an archive file or directory.

Select an import source:

The project should look like the screenshot below.



Project Explorer

- Local Filesystem
- ABasicCalculator
  - css
    - main.css
  - js
    - test
      - lib
        - jasmine-2.1.3
      - ifelseSpec.js
      - inlineSpec.js
      - switchSpec.js
    - vendor
      - math.js
    - ifelse.js
    - switch.js
  - ABasicCalculator-ifelse.html
  - ABasicCalculator-inline.html
  - ABasicCalculator-switch.html
  - Connections

Step 1: Adding the onclick Elements Inline

1. Open the **ABasicCalculator-inline.html** file in **Brackets** or **Nodeclipse**.
    - *Alternatively,* you can open the **ABasicCalculator-inline.html** file in **Dreamweaver**, **TextEdit**, **Notepad++**, or some other text editor of your choice.
    - You do not have to touch any of the **CSS** inside the <**style**> tags or external stylesheets and you should also not change anything in the contents of the **js/test** & **js/vendor** folders.
2. In the first <**h1**> heading, replace "**Your Name Goes Here**!" with *all of your* **first and last names**.
3. Look at the remaining **comments** on the page and **follow** their **directions** to build a "working" calculator by adding **onclick** event attributes.
4. Save the **ABasicCalculator-inline.html** so later you can submit it to this assignment in Canvas.
5. **Test** your calculator and **fix** any errors, until **everything** works as expected.

HINT: Pay special attention to the **comments** that contain a "**Note**" since they're there to help you.

Step 2: Adding the calcu Function using a switch statement

1. Open **ABasicCalculator-switch.html**. In the first <**h1**> heading, replace "**Your Name Goes Here**!" with *all of your* **first and last names**.
2. Open the external .js page at **js/switch.js** and add **Your Names** in **// comments** at the top.
3. Add a <**script**> link to the **js/switch.js** in the <**body**> of your **ABasicCalculator-swtch.html** file *just **after*** the <**script**> link to js/vendor/math.js.
4. Change the HTML code to make the correct input buttons (inside the form) to inline bind onclick to call the calcu function with the correct parameters.

5. Complete the **calcu** function in the **js/switch.js** file using a **switch** statement.
6. Save the **ABasicCalculator-switch.html** and **js/switch.js** files for later submission to Canvas.
7. **Test** your calculator and **fix** any errors, until **everything** works as expected.

HINT: In the switch, don't forget to include a **break;** after each of the pertinent **case** statements.

Step 3: Adding the calcu Function using if...else if... else

1. Open **ABasicCalculator-ifelse.html**. In the first <**h1**> heading, replace "**Your Name Goes Here**!" with *all of your* **first and last names**.
2. Open **js/ifelse.js** and add **Your Names** in **// comments** at the top.
3. Add a <**script**> link to this **js/ifelse.js** in the <**body**> of your **ABasicCalculator-ifelse.html** file *just **after*** the <**script**> link to js/vendor/math.js.
4. DO NOT change the HTML form code because you will bind the input buttons' onclick event to functions inside **js/ifelse.js** *instead* of using inline JavaScript code. To do that:
    1. In **js/ifelse.js** after the comment "// Binding onclick events" add code to get all the input elements into an array-like HTMLCollection called inputs (using **document.getElementsByTagName** **(https://developer.mozilla.org/en-US/docs/Web/API/Document/getElementsByTagName)**) and use a for-loop to iterate through the array.
    2. During the for-loop you will use an if statement to skip the input element that is not a button
    3. Then set the other input elements' onclick handler to a function that **calls** the calcu function from inside of it.
    4. **Make sure** you pass **this.id** to calcu so calcu will have the correct value for its internal **calcValue** variable.
5. Complete the **calcu** function in the **js/ifelse.js** file using a **if...else if...else** statement.
6. Save the **ABasicCalculator-ifelse.html** and **js/ifelse.js** files.
7. **Test** your calculator and **fix** any errors, until **everything** works as expected.

Step 4: Submission

Submit the following before the due date:

- You will ZIP archive the entire ABasicCalculator folder and submit the resulting **ABasicCalculator.zip** file to this assignment.
- **When you are done, *each* part of this assignment should behave just like this *demonstration video* (http://edutek.net/foothill/cs/22a/A2/) ! Make sure your team tests all 3 parts using FirefoxDeveloperEdition with Brackets' Live Preview** (which will be used to evaluate your work) **and then tests the project over the World Wide Web HTTP protocol** (when the URL will start with "**http:**"). For instructions for how to use Live Preview review the page: **How to use the Live Preview feature in the Bracket.io IDE**

Grading

It is possible to earn some extra credit on certain rubric criteria that have an "Exceeds Expectations" top category.

Peer-Review

It is important to submit this on-time because everyone will peer-review 2 other students' work just after the due date. When you do **your** review of a peer, make sure you leave a comment ***AND*** use the scoring rubric. **Read these pages on how to do your peer-review:**

- **How do I know if I have a peer review assignment to complete? (https://community.canvaslms.com/docs/DOC-3139)**
- **How do I submit a peer review to an assignment? (https://community.canvaslms.com/docs/DOC-3138)**

Original author: **Craig Duckett (http://faculty.cascadia.edu/cduckett/)** Updated by: **Dr. Baba Kofi Weusijana (https://www.linkedin.com/in/babakofi)**

A Basic Calculator Rubric

Criteria	Ratings			Pts
Completed successfully code for ABasicCalculator-inline.html	**15 pts** **Calculator works for users as specified**	**8 pts** **Calculator passes all tests**	**0 pts** **Does Not Meet Expectations**	15 pts
Completed successfully code for ABasicCalculator-ifelse.html	**15 pts** **Calculator works for users as specified**	**8 pts** **Calculator passes all tests**	**0 pts** **Does Not Meet Expectations**	15 pts
Completed successfully code for ABasicCalculator-switch.html	**15 pts** **Calculator works for users as specified**	**8 pts** **Calculator passes all tests**	**0 pts** **Does Not Meet Expectations**	15 pts
Submitted sufficient comments (at-least psuedocode for each part) so peer-reviewers can understand your goals and plans. Even submitting a list of what you don’t know is better than submitting only a note that you don’t know how to start or submitting nothing.	**8 pts** **Meets Expectations**	**0 pts** **Does Not Meet Expectations**		8 pts
Understand the basic syntax of the JavaScript language Your code should be interpreted by Web browsers' JavaScript engines without out any syntax errors (which can be seen in a browser's development console). Warnings reported by the console regarding your code are OK if they are not avoidable with the techniques you have been exposed to in this course or they are not relevant to completing the program as specified. threshold: 3.0 pts	**3 pts** **Your code has no syntax errors**	**0 pts** **Your code has at least one syntax error**		3 pts

Criteria	Ratings			Pts
Produce clearly written code in an industry standard style appropriate for JavaScript. threshold: 3.0 pts	**5 pts** **Exceeds Expectations**	**3 pts** **Meets Expectations**	**0 pts** **Does Not Meet Expectations**	5 pts
Write a JavaScript program that responds appropriately to user events. Students are expected to write JavaScript programs that respond appropriately to user events, such as mouse clicks on buttons. threshold: 3.0 pts	**5 pts** **Exceeds Expectations**	**3 pts** **Meets Expectations**	**0 pts** **Does Not Meet Expectations**	5 pts
Write client-side JavaScript code to modify the Document Object Model. Students are expected change web pages with JavaScript by changing their content by changing, inserting, or removing Document Object Model(DOM) objects. threshold: 3.0 pts	**5 pts** **Exceeds Expectations**	**3 pts** **Meets Expectations**	**0 pts** **Does Not Meet Expectations**	5 pts

Total Points: 71