

# WWW Technologies and Applications 2021

## Assignment 2 – JavaScript, DOM, and JSON

TA: Humphrey (humphrey0925@alum.ccu.edu.tw)

Deadline: 11:59pm, April 8, 2021

Demo Video: <https://youtu.be/RMNaR1wJVtQ>

### 1. Task Description

In this assignment, you will build a calculator with the required layout using JavaScript, DOM, and JSON. **You should link your homework2 button in the page you completed for assignment 1 to this assignment. If your homework2 button could not link to this assignment, this assignment will NOT be graded.** The features should be restricted as follows:

#### 1.1 The layout of the page:

Figure 1 shows the layout of the calculator. As for a standard calculator, there is a screen to display the number that the user inputs and an operation area to operate the numbers. You can find a tutorial here (<https://freshman.tech/css-grid-calculator/>). Additionally, for a JSON calculator, you need to implement a drop-down list and a Load button next to it in the operation area for loading a JSON file that contains the input numbers and operators for calculation.

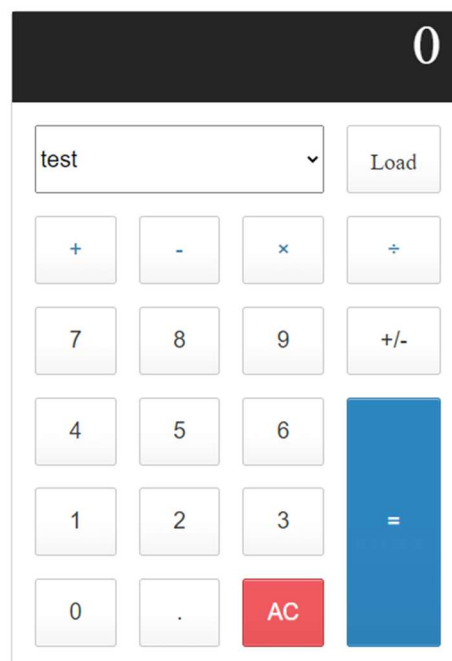


Figure 1. Calculator layout

#### 1.2 The standard calculator

##### 1.2.1 Four operators

Your calculator should implement four major operators: addition, subtraction, multiplication, and division. Users can click on the digit buttons to input numbers. For example, if the user wants to input "235", the user can click the digits as follows: 2 > 3 > 5, and your display area should be "2>23>235" after each click on a digit. For more information, you can check our demo video.

### 1.2.2 UI and UX

The calculator should compute the result of the previous operation at the display area right after clicking on the next operator button. Your AC button should clean the output of the display area. The number in the display area should change the sign after clicking the "+/-" button for converting a positive number into a negative. For example, the number "123" should be "-123" after clicking the "+/-" button.

### 1.2.3 Float numbers

A float number should round off to the 4<sup>th</sup> decimal place after operations. For example, if the result of calculation is 3.14159, you should display 3.1416 as a result. If the result of calculation is 3.14144, you should display 3.1414 as a result.

### 1.2.4 Error exception

Your calculator should handle the following exceptions:

- (1) The division should output Err after the user inputs zero as the divisor.
- (2) The user should not input "." after inserting a float number.
- (3) The font size of the display area should be smaller when the number of digits is above 13.  
For example, your font size should change from 5 rem to 3 rem when the number of digits is above 13.

## 1.3 JSON calculator

A drop-down list of the calculator should be the list from our document website (<http://www2021.csie.io:82/json>). A user can choose a JSON file that contains the input numbers and operators and click on the Load button to conduct operations. Your list MUST include two JSON files called hidden1.json and hidden2.json for grading. To access your JSON file from our website, you can use the following code in your code:

```
fetch('http://www2021.csie.io:82/api/v0/json?name=' + YOUR_FILE_NAME)
  .then(function(response) {
    return response.json()
  }).then(function(res_json) {
    //Your code in here.
    //res_json is the json from our website according to which file you want
    //to use.

  })
```

Your calculator should calculate the operations, given a JSON file. Each item in the JSON file includes types and values. A type is either a number or an operator. Your result should output in the display area. hidden1.json will be available after the deadline of the submission, and hidden 2.json will be available after the deadline of the late submission. **All functions in 1.2 should include into JSON calculator.**

## 2. How to submit your assignment and what should be submitted

### 2.1 How to submit your assignment

The procedure of submitting your assignment can be found at our document website (<http://www2021.csie.io:82/homework/submit>).

### 2.2 What should be submit

You should **hand in your assignment to the GitLab** and deploy your website on your server. You should record a demo video to show each function of your assignments. Your demo video should upload to YouTube and paste your link into eCourse2. **Your assignment will not be graded without a demo video.**

## 3. Grading policy

**Graders will test your homework only on Google Chrome in 1920\*1080. Homework submitted late will be accepted for up to 7 days after the due date, and will receive an automatic 30% penalty. Homework submitted more than 7 days after the due date will not be accepted. Only one final submission (either one on-time or one late submission) is accepted. Your assignment will not be graded without a demo video. You should explain all function of your assignment in the demo video. The link of your demo video should be submitted to eCourse.** The TA(s) will mark and give points according to the following grading policy:

Calculator	Layout	5%
	Four operators with correct number display	20%
	UI and UX	15%
	Float number	10%
	Error detection (Each exception: 10%)	30%
JSON	JSON calculator	20%