IS216 Web Application Development II

Practice Questions

If you spot any errors in the questions and/or source code files, please kindly inform Kyong SHIM (kjshim@smu.edu.sg).

Q1: CSS and JavaScript (10 Points)

Given resources:

- Q1
  - q1.html
  - q1.css

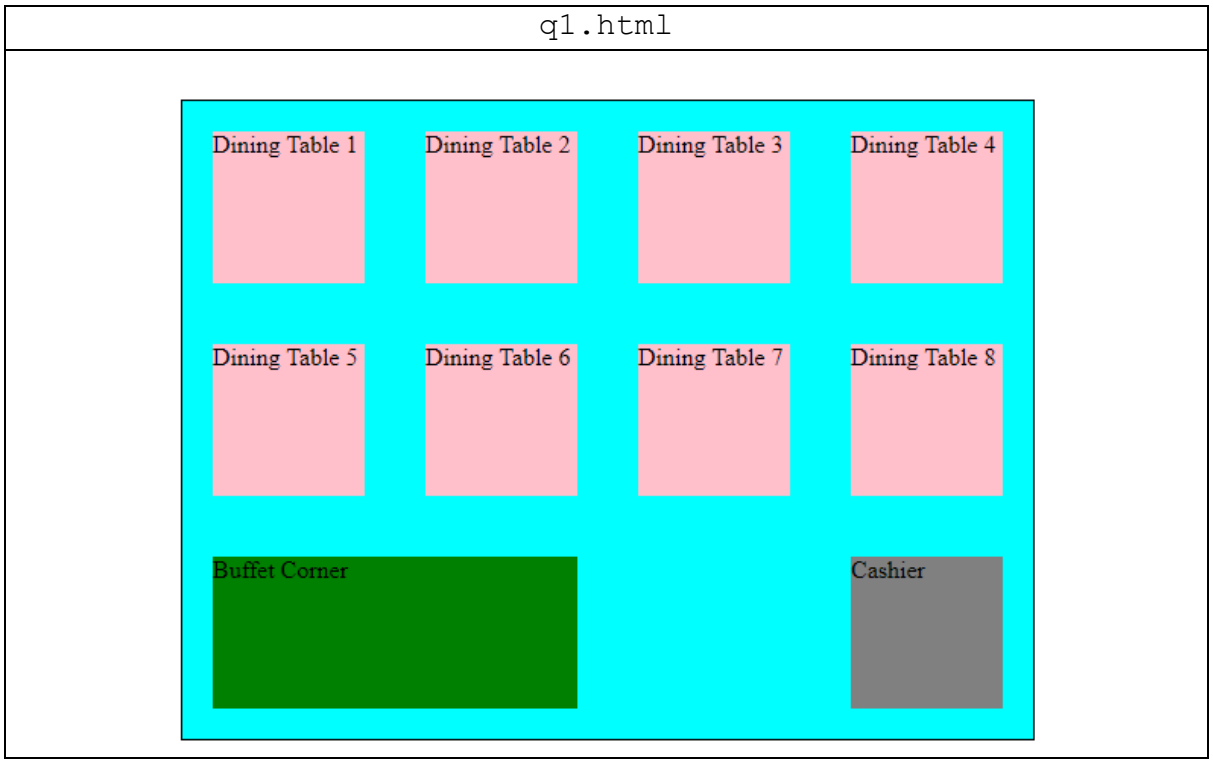
You are to **edit** all 2 files for **Part 1** and **Part 2**.

Part 1 – HTML & CSS

We wish to build a **web page** that displays a restaurant's **floor** with dining tables, buffet corner, and cashier table.

You must use vanilla CSS for this question (NO Bootstrap).

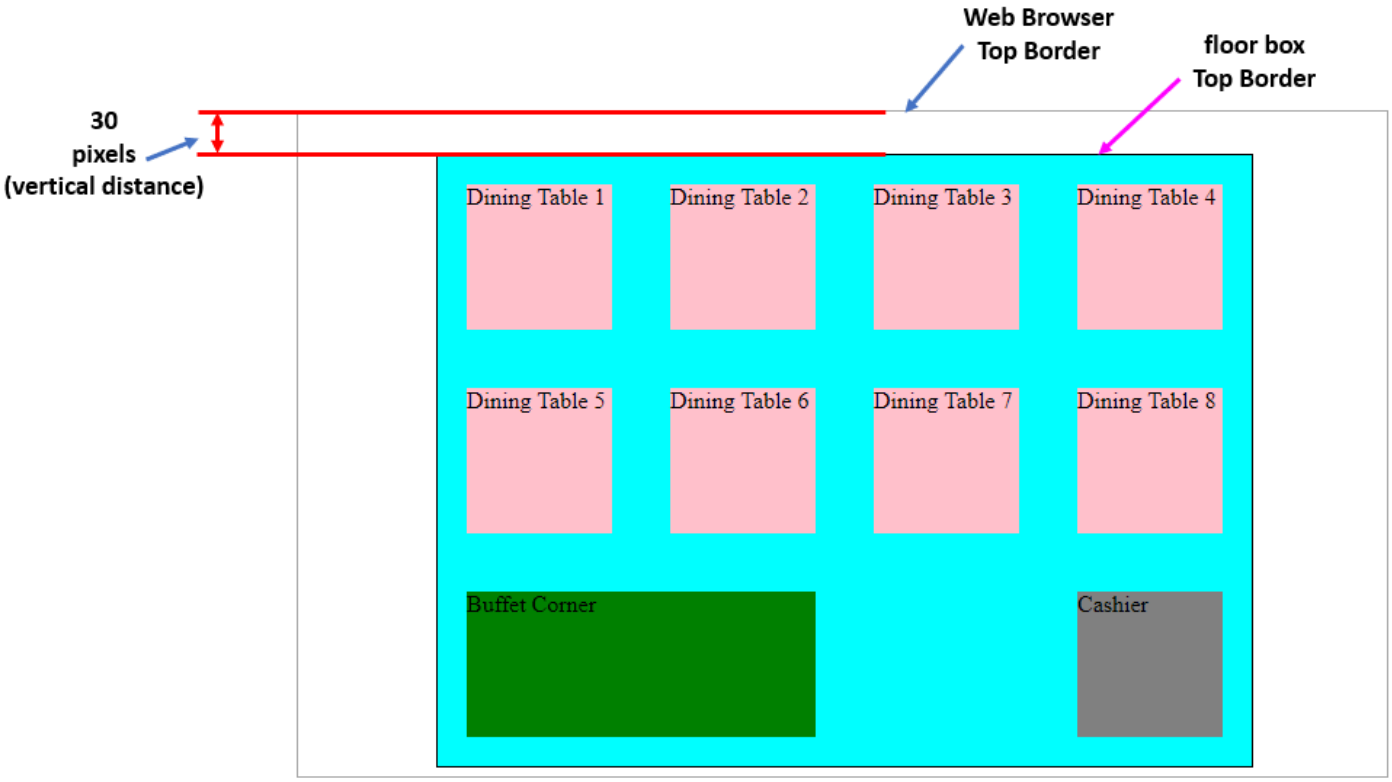
Some skeleton code is given in both **`q1.html`** and **`q1.css`**. Complete both files such that **`q1.html`** displays as shown below. Assume that the user will view **`q1.html`** from a laptop computer screen with screen width > **800 pixels** and screen height > **600 pixels**.

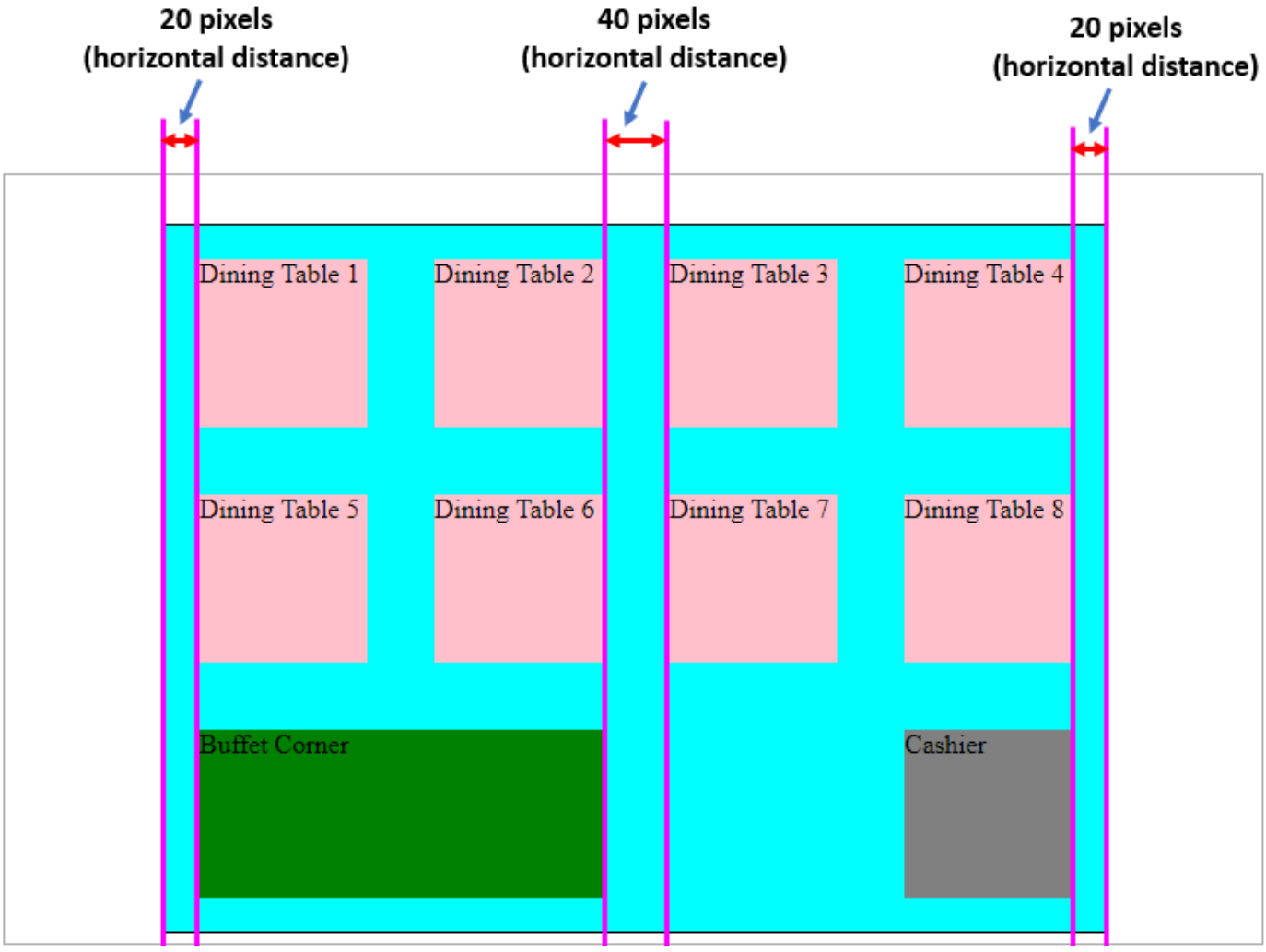


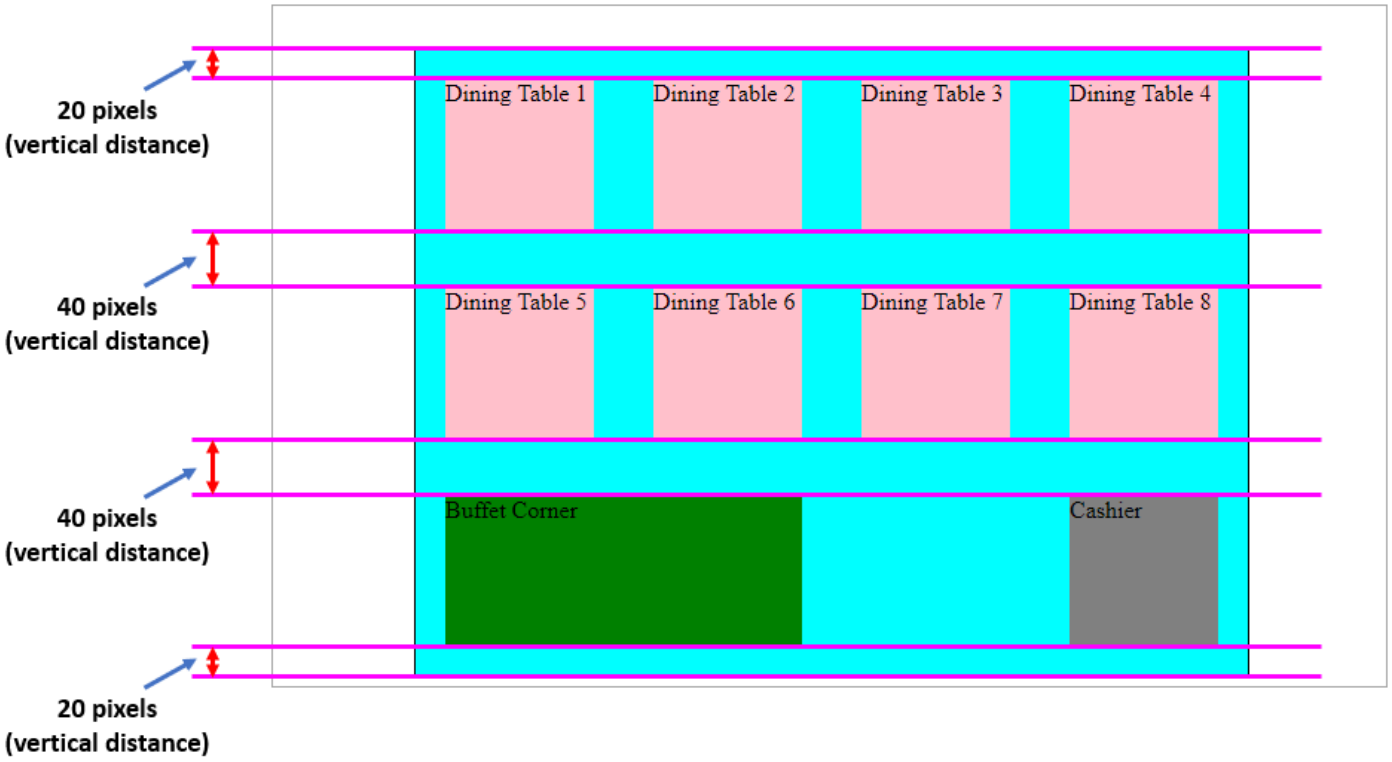
Please ensure:

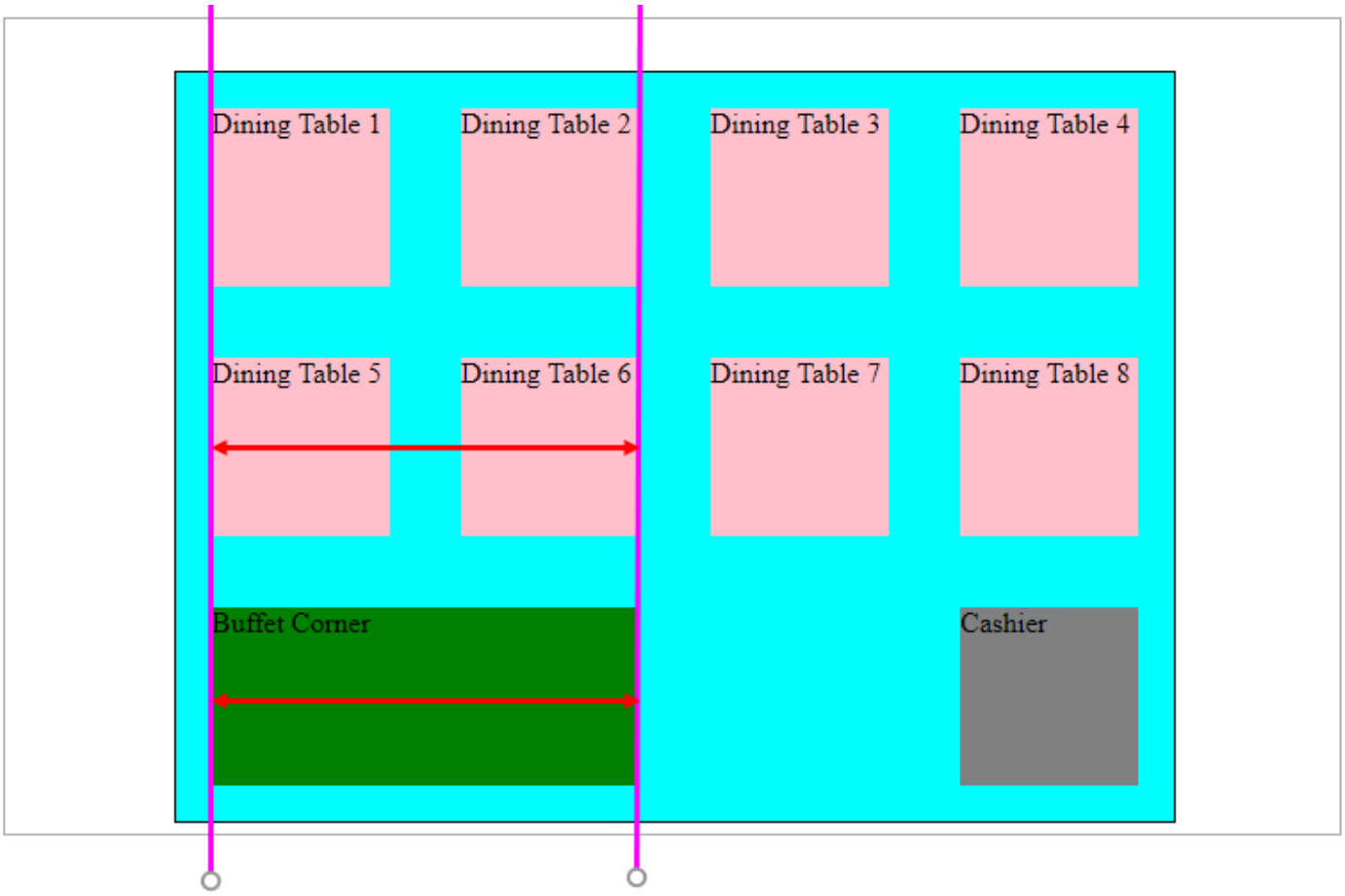
- The **floor** box is horizontally centered in the web page. The **floor** box has fixed width of **560 pixels** and fixed height of **420 pixels**.
- There is a **margin** of **30 pixels** between the web browser's **top border** and the **floor** box's top border.
- Each **dining table** has fixed width of **100 pixels** and fixed height of **100 pixels**.
- The **Cashier** table also has fixed width of **100 pixels** and fixed height of **100 pixels**.
- From the screenshot above, you can infer the dimensions of the **Buffet Corner** table (in relation to the dining table).

Please refer to the next pages for more guidelines.


Web Browser
Top Border
floor box
Top Border
30
pixels
(vertical distance)
Dining Table 1
Dining Table 2
Dining Table 3
Dining Table 4
Dining Table 5
Dining Table 6
Dining Table 7
Dining Table 8
Buffet Corner
Cashier


20 pixels
(horizontal distance)
40 pixels
(horizontal distance)
20 pixels
(horizontal distance)
Dining Table 1
Dining Table 2
Dining Table 3
Dining Table 4
Dining Table 5
Dining Table 6
Dining Table 7
Dining Table 8
Buffet Corner
Cashier





Buffet Corner's width is *(roughly)* about the same as 2 dining tables' widths combined + horizontal distance between them.

Part 2 – JavaScript

In **`q1.html`**, scroll all the way down and see just above `</body></html>`. You will see **`<script>`**…**`</script>`** section. Complete the **JavaScript** section of **`q1.html`** as per the following rules:

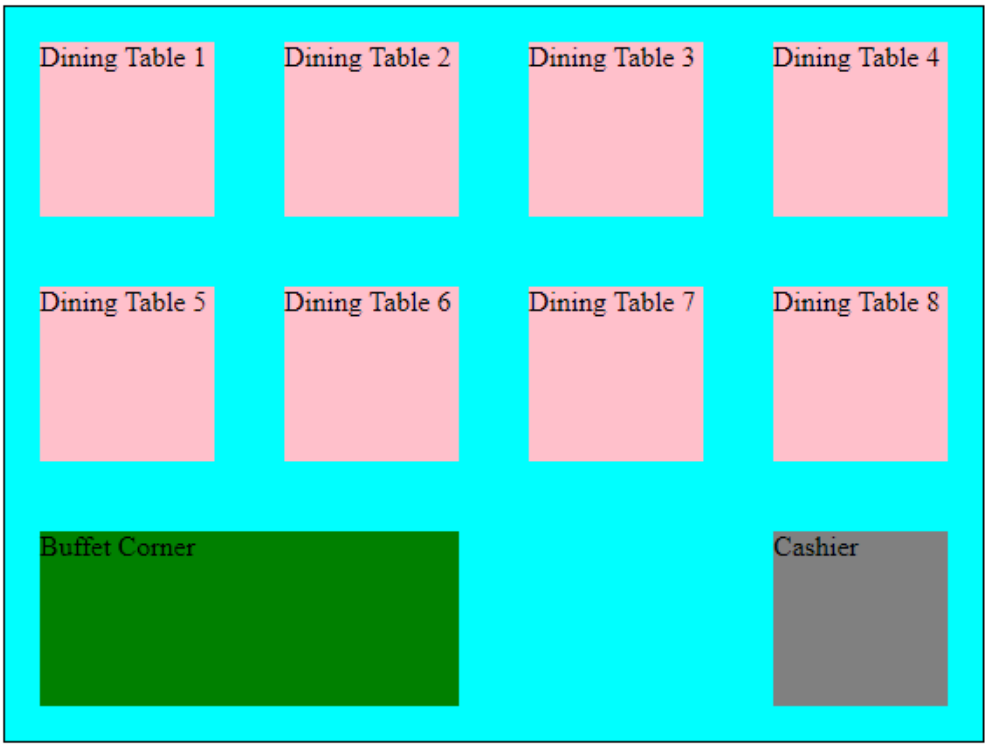
When the user **clicks** on a particular **dining table**:

1. Your JavaScript code will **prompt** the user with **"What is your name?"**
2. If the **user input** is not an empty string, your code will add the **user's name** to the pink **dining table** box.
3. Your code must work correctly for **any dining table**.

For example:

q1.html

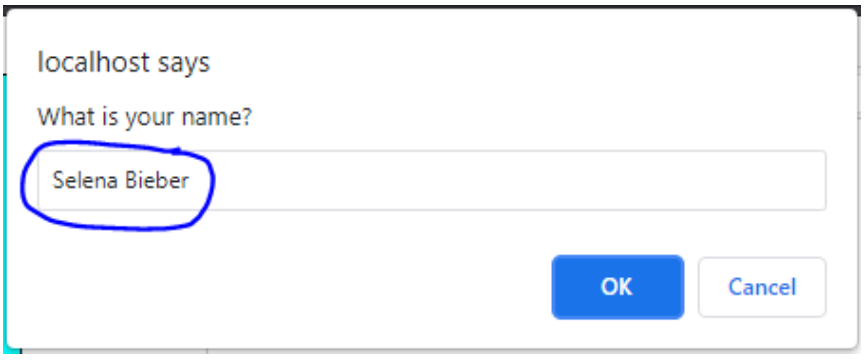
(The web page has just been loaded by the user)



User **clicks** on the pink box that says **Dining Table 7**.

q1.html

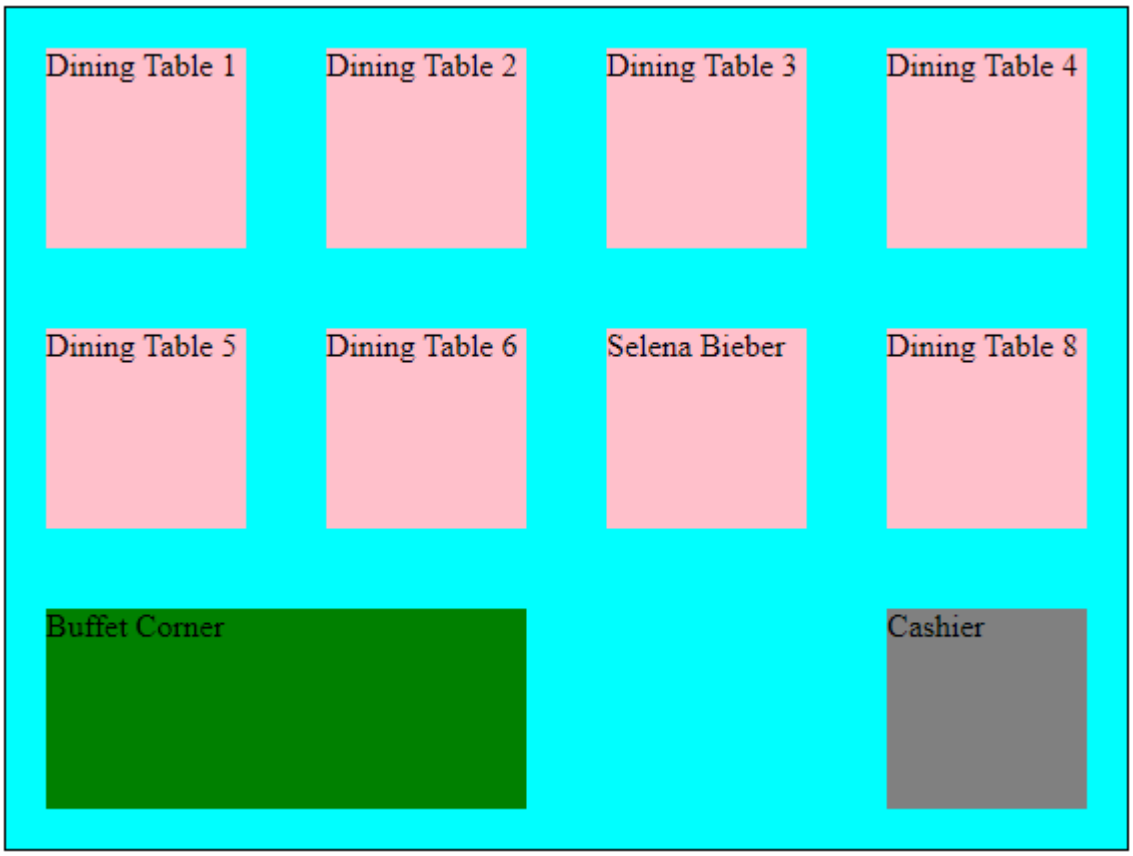
*(After the user clicks on **Dining Table 7**)*



User is **prompted** to enter her **name**.
She keys in **Selena Bieber** and presses **OK**.

q1.html

(After the user keys in her name and presses OK)



Dining Table 7 is now reflecting **Selena Bieber** (user's name).

Q2: CSS and JavaScript (10 Points)

Given resources:

- Q2
  - q2.html
  - q2.css
  - q2.js

You are to **edit** all 3 files for **Part 1** and **Part 2**.

Part 1 – HTML & CSS

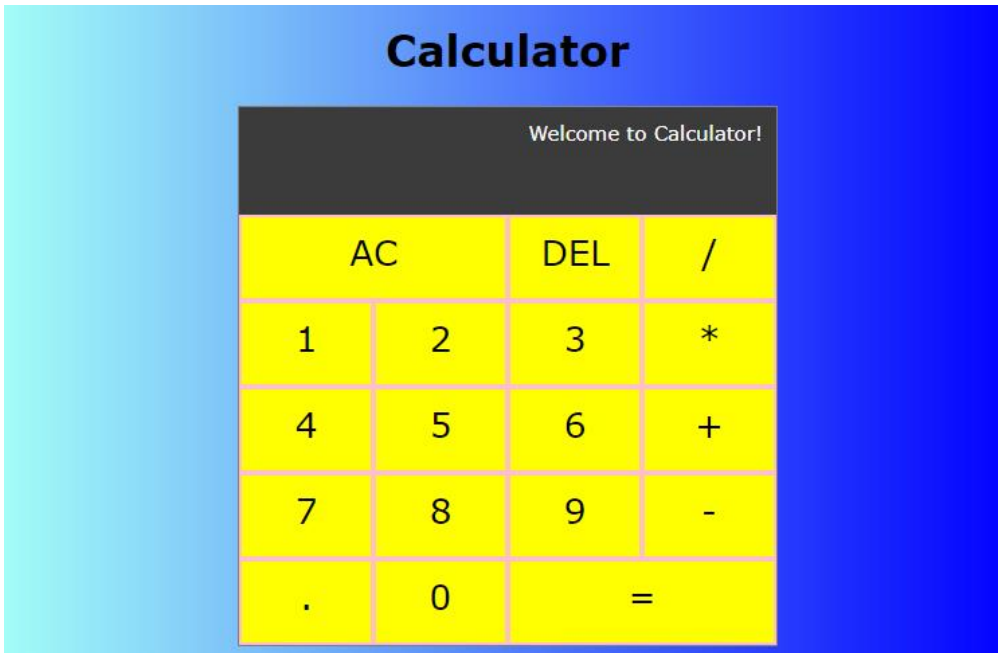
We wish to build a **web page** that displays a calculator.

You are allowed to use Bootstrap for this question (optionally).

See if you can do vanilla CSS first ^_^

Some skeleton code is given in both **`q2.html`** and **`q2.css`**. Complete both files such that **`q2.html`** displays a calculator as shown below. Assume that the user will view **`q2.html`** from a laptop computer screen with screen width > **800 pixels** and screen height > **600 pixels**.

q2.html

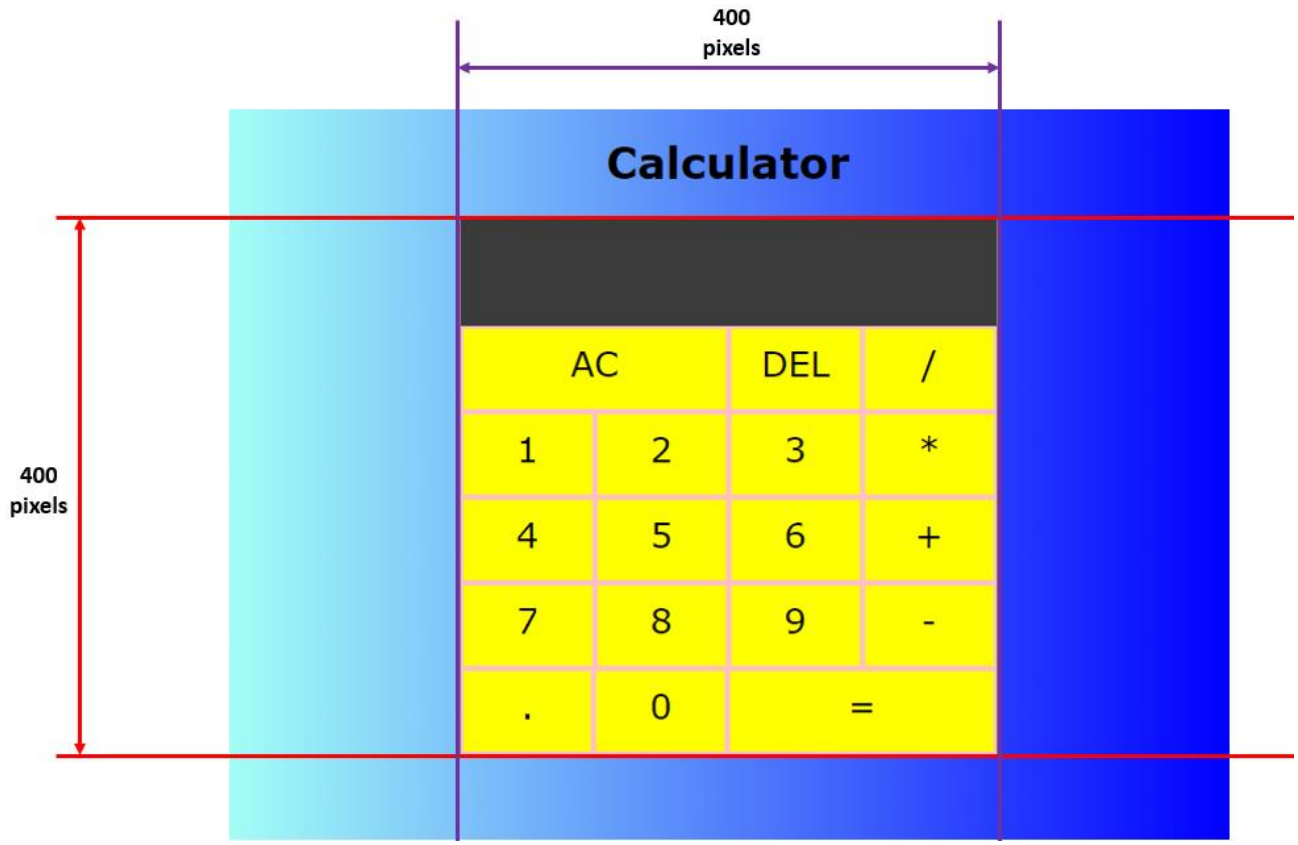


Please ensure:

- **Heading-1** (“Calculator”) is horizontally centered in the web page.
- The calculator **grid** is horizontally centered in the web page. The **grid** has fixed width of **400 pixels** and fixed height of **400 pixels** *(see next page for a visual illustration)*.
- The font size of the keypad options (e.g. AC, DEL, operators and numbers) is **26 pixels**.

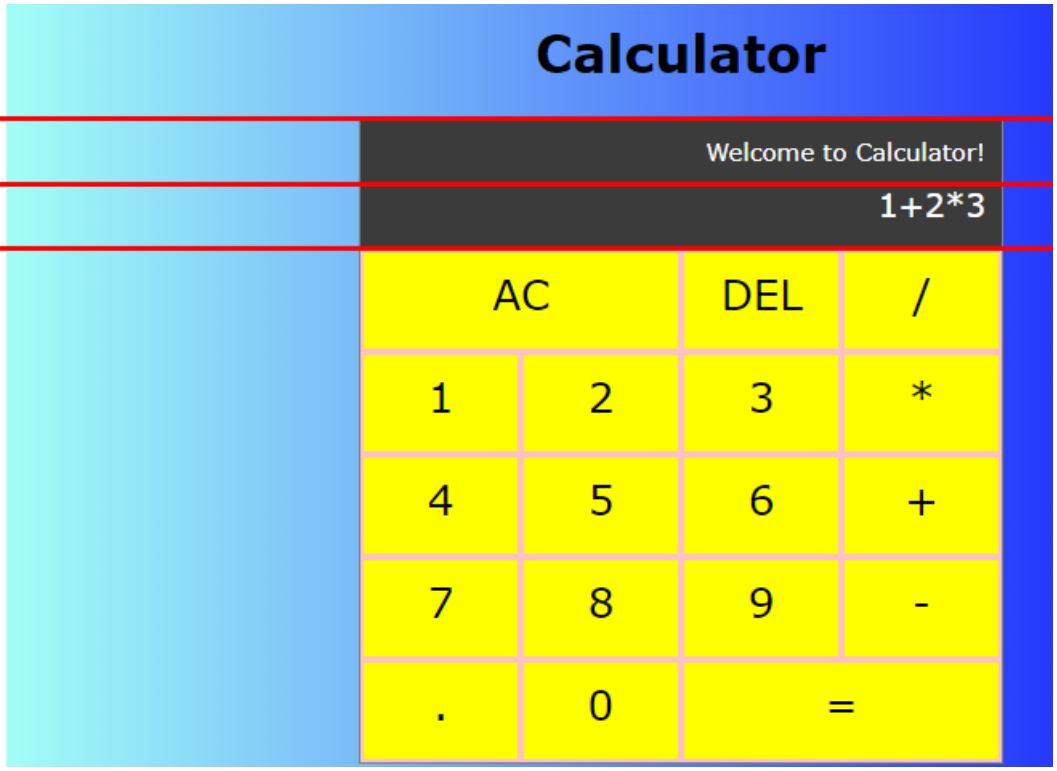
As for **background** (e.g. linear gradient) and **border color/width**, attempt to match the above screenshot as closely as possible (*it does NOT have to be exactly the same*).

Please refer to the next pages for more guidelines.

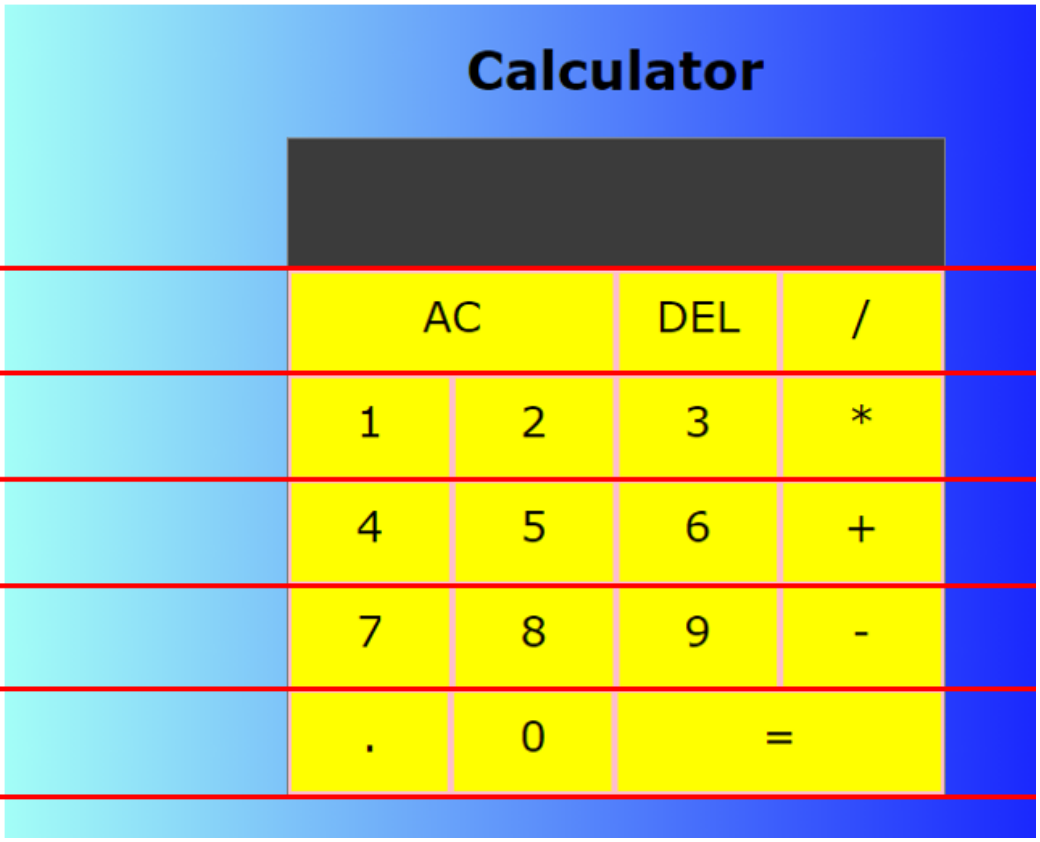


These 2 rows have the same height (40 pixels each)

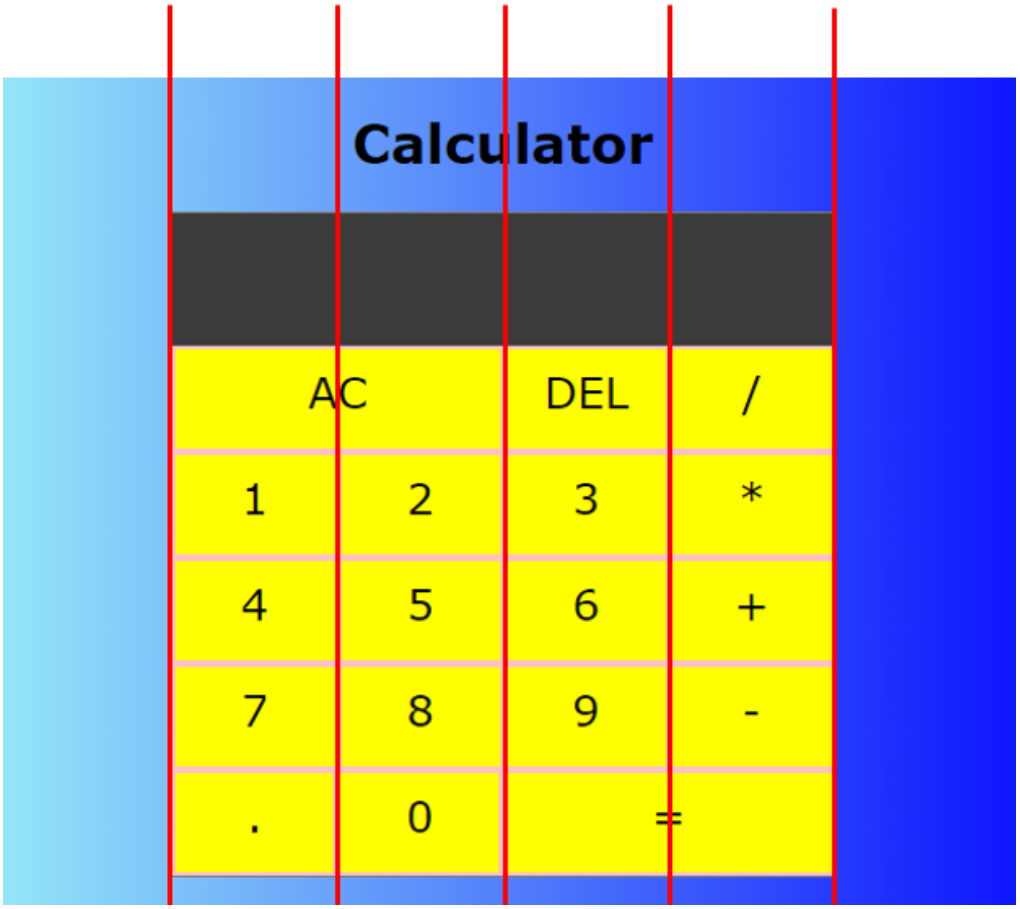
The content in both boxes are RIGHT-indented.



These 5 rows have the same height

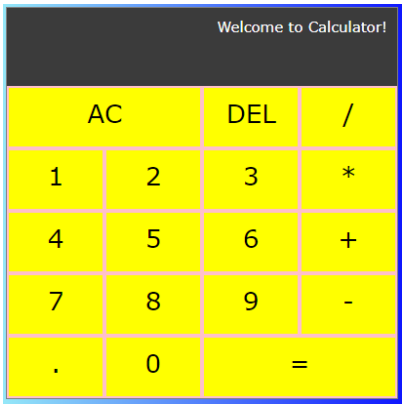


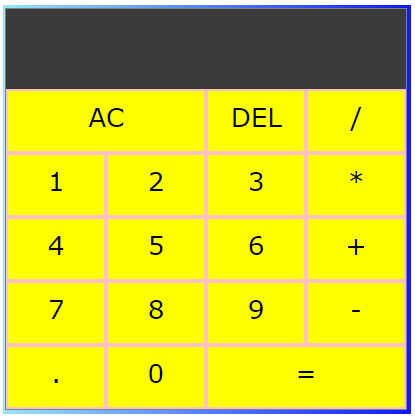
These 4 columns have the same width
Exceptions are "AC" and "=" key options
The "AC" key and "=" key are *(roughly)* twice as wide as "DEL" key (or "0" key)

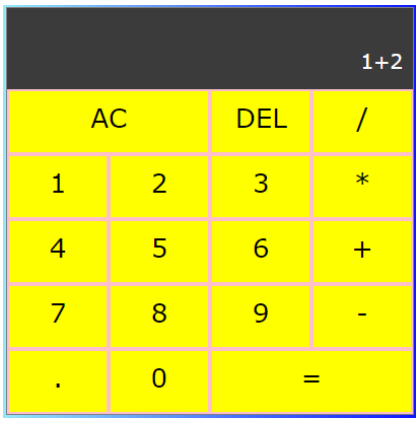
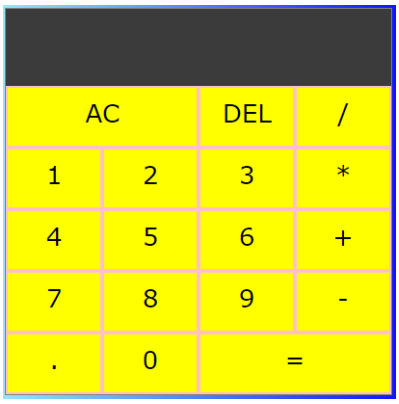


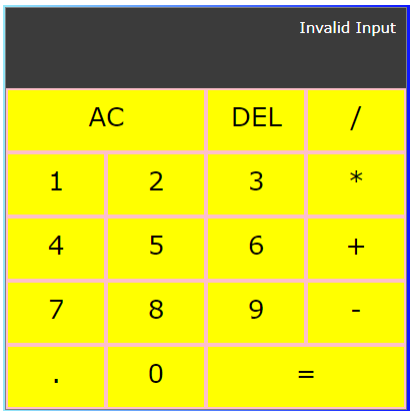
Part 2 – JavaScript

Complete `q2.js` as per the following rules:

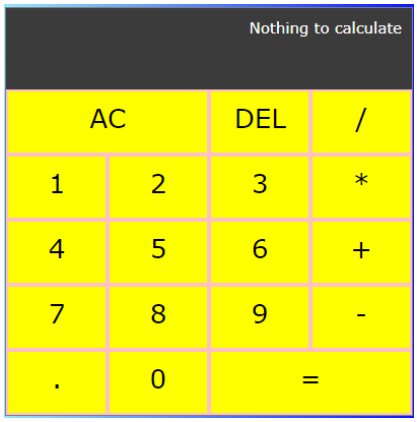
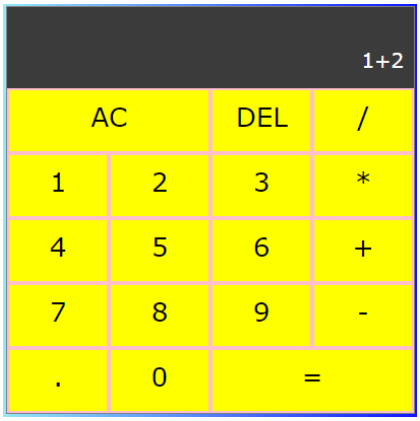
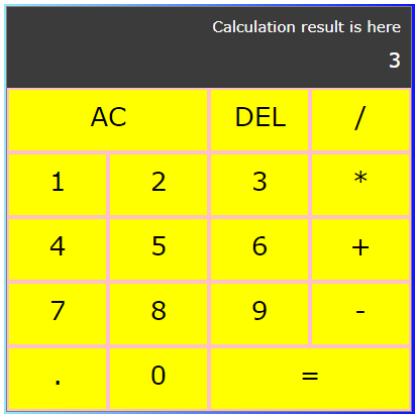
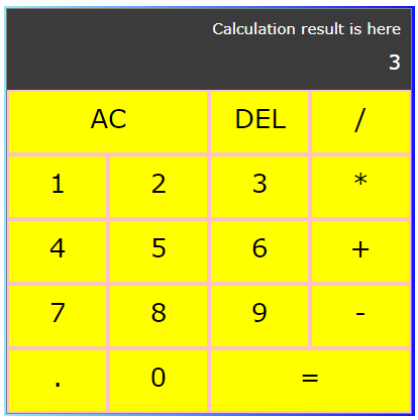
<table>
<tr><th>q2.html</th><th>q2.html</th></tr>
<tr><td> 

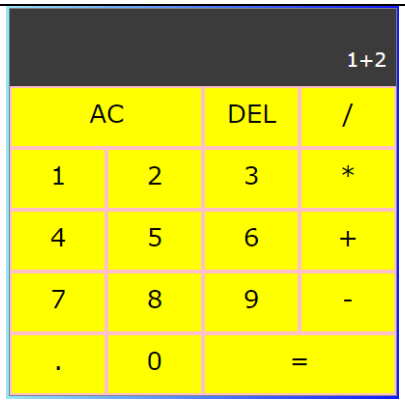
User clicks on AC key option</td><td> 
</td></tr>
<tr><td> 

User presses the following key options (sequentially):
1
+
2</td><td> 
</td></tr>
<tr><td> 

User presses any of the following key options:
/
+
*
-
.</td><td> 

Calculator displays Invalid Input inside message </td></tr>
</table>

 User presses **=**	 Calculator displays **Nothing to calculate** inside **message**
 User presses **=**	 Calculator displays **Calculation result is here** inside **message** Calculator displays the **calculated value** inside **current-equation**
 User clicks on **AC** key option	 **message** is cleared (no text) **current-equation** is cleared (no text)

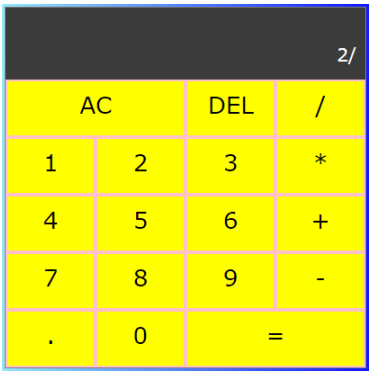
<table>
<tr>
<td>


User presses DEL
</td>
<td>

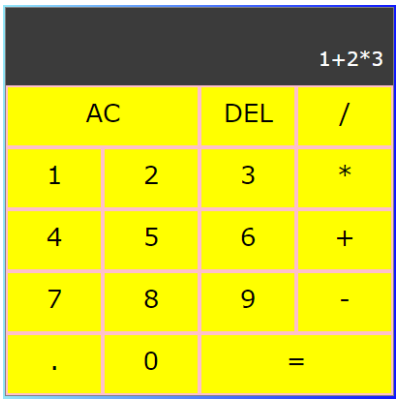

See how 2 has been removed.

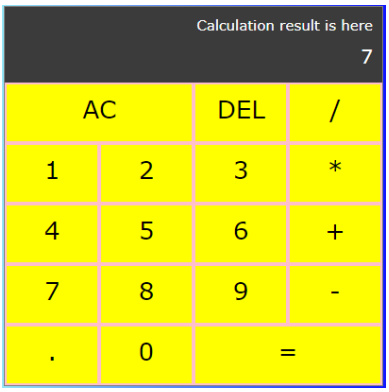
Each click of DEL must remove the right-most character of current-equation 's text

If current-equation 's text is empty, there is no more text left to DELETE.
</td>
</tr>
<tr>
<td>


User presses /
</td>
<td>


See how + has been replaced with /

When the right-most character of current-equation 's text is an operator (+, -, *, /) or period (.), this right-most character is replaced with user's new input.
</td>
</tr>
<tr>
<td>


User presses =
</td>
<td>


</td>
</tr>
</table>

HINT: Consider using **`eval()`** function for calculation.
Ref: https://www.w3schools.com/jsref/jsref_eval.asp

Q3: Asynchronous Request and JavaScript (10 Points)

Given resources:

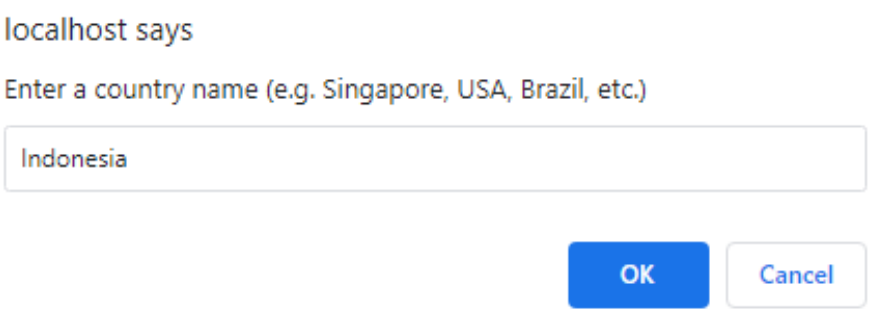
- Q3
  - q3.html
  - axios\
    - axios.min.js (DO NOT MODIFY THIS FILE)
  - covidapi
    - count.php (DO NOT MODIFY THIS FILE)

You are to **edit** `q3.html` **only**. Do not edit the other files.

We wish to build a **web page** that displays global COVID-19 statistics.

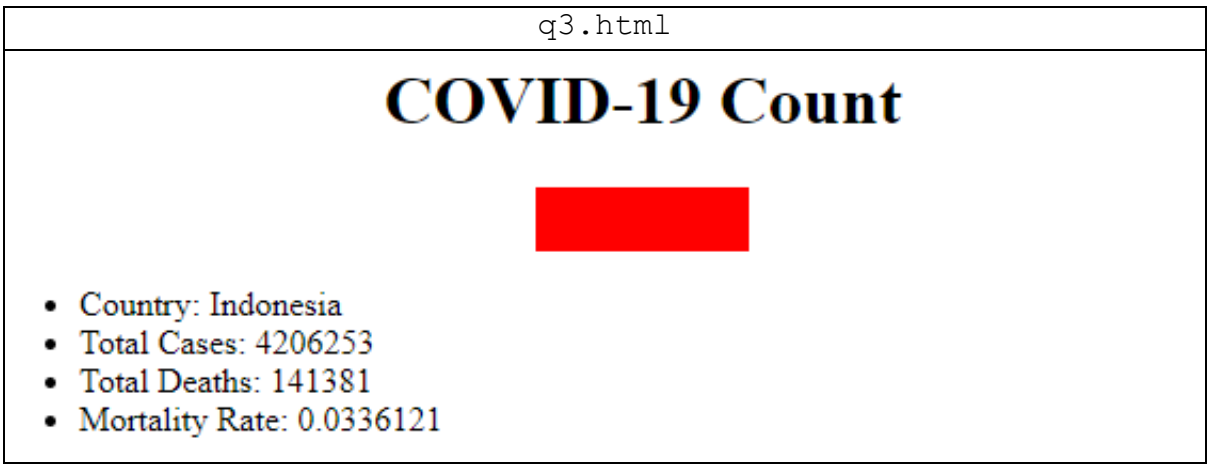
When **`q3.html`** is loaded, the user will be prompted to enter a **country name**.

q3.html



As per the above screenshot, assume that the user has entered **Indonesia** and pressed **OK** button. The web page will then display **Indonesia**'s COVID statistics as shown below.

q3.html



There are 3 parts to this web page:

1. <h1> heading which displays **COVID-19 Count**.
2. A 100px by 30px **box** (it is a DIV element with id='risk') which is **colored**. It is situated below the <h1> heading. *In the above example, the box is colored in red.*
   a. You need to use an appropriate **CSS styling** as specified in **q3.html**'s **Internal CSS** (see <style>…</style>).
   b. The **box** must be colored according to these rules.

Mortality Rate	Box Color
Mortality Rate < 0.01	green
0.01 <= Mortality Rate < 0.02	yellow
0.02 <= Mortality Rate < 0.03	orange
0.03 <= Mortality Rate	red

3. An **unordered list** with 4 **list items**, and they are:
    a. **Country** (e.g. USA, Brazil, Indonesia, France, and so on). *This info comes from the **prompt** user input.*
    b. **Total Cases** indicating the total number of COVID cases. *This info comes from the API response*.
    c. **Total Deaths** indicating the total number of COVID deaths. *This info comes from the API response.*
    d. **Mortality** Rate. You need to calculate this value from **Total Cases** and **Total Deaths** (total deaths divided by total cases) and display in the web page. The calculated value must be in **7 decimal places**.

To retrieve the COVID-19 statistics, your JavaScript code must call the **covidapi** API which is already provided to you ***locally*** inside **Q3** folder. For example, if your **`q3.html`** is currently at:

```
http://localhost/is216/Week7/Review/Q3/q3.html
```

The **covidapi** API endpoint your Q5.html JavaScript code needs to call is at:

```
http://localhost/is216/Week7/Review/Q3/covidapi/count.php
```

It is highly suggested that you use either the web browser or Postman to test the API endpoint and observe the API response before coding JavaScript in **`q3.html`**.

Here is a snippet of the API response:


```
{
    "statistics": {
        "North America": {
            "USA": {
                "total_cases": 43668680,
                "total_deaths": 705293,
                "total_recovered": 33112993,
                "active_cases": 9850394,
                "population": 333389570
            },
            "Canada": {
                "total_cases": 1598800,
                "total_deaths": 27620,
                "total_recovered": 1526156,
                "active_cases": 45024,
                "population": 38148931
            }
        },
```


… more data below..

See the next page for sample test cases and expected outputs.

q3.html

User input country → SINGAPORE

COVID-19 Count

- Country: Singapore
- Total Cases: 84510
- Total Deaths: 73
- Mortality Rate: 0.0008638

q3.html

User input country → canada

COVID-19 Count

- Country: Canada
- Total Cases: 1598800
- Total Deaths: 27620
- Mortality Rate: 0.0172755

q3.html

User input country → arGENtina

COVID-19 Count

- Country: Argentina
- Total Cases: 5248847
- Total Deaths: 114828
- Mortality Rate: 0.0218768