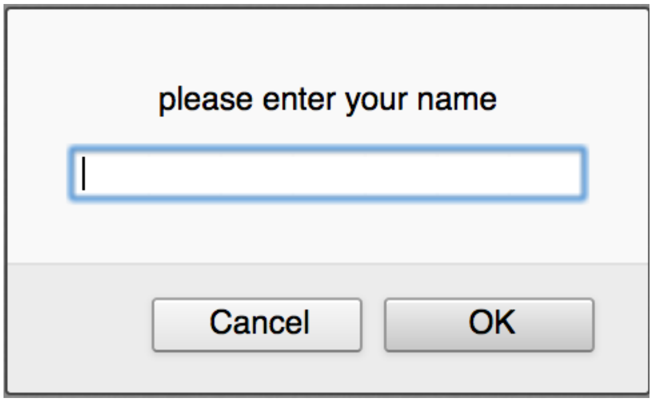
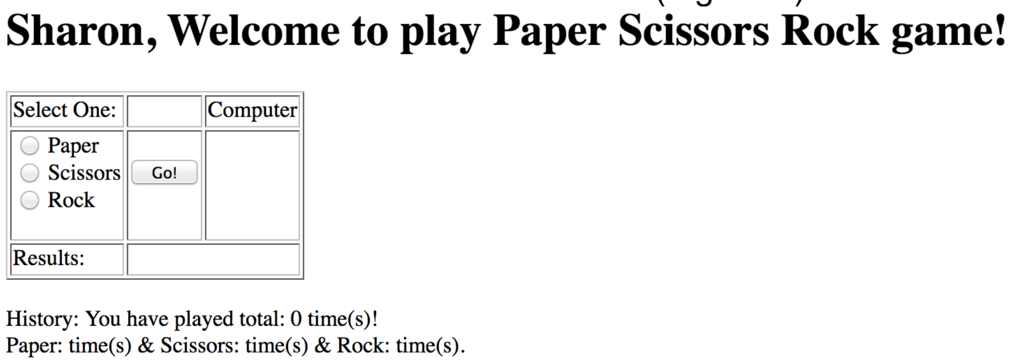
Learning Goal: Practice JavaScripts and create dynamic webpages.

Tasks:
Create a paper-scissors-rock game on dynamic webpages. On loading the page, should prompt windows to solicit user’s name (Figure 1), then the page will display user’s input to welcome him/her on a HTML page (Figure 2). User may now start playing the game by selecting one out of three (paper/scissors/rock) options to compete with a random generated paper/scissors/rock by the computer. The result should be displayed immediately on the same page, and the page should update user dynamically how many times s/he has been playing, and how many of papers/scissors/rocks that the computer generates. (per sample output).



please enter your name

(Figure 1)

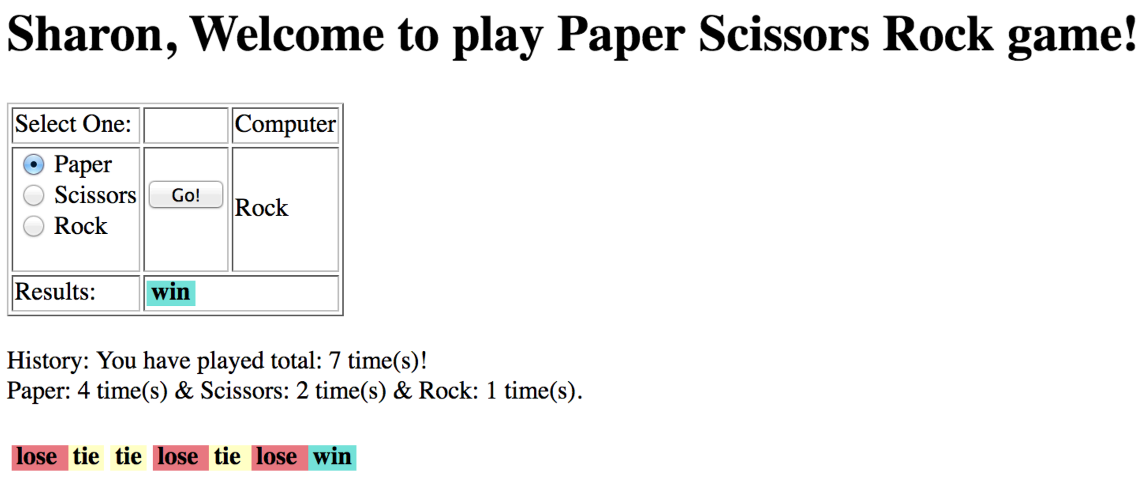


Sharon, Welcome to play Paper Scissors Rock game!

Select One:		Computer
○ Paper ○ Scissors ○ Rock	Go!	
Results:		

History: You have played total: 0 time(s)!
Paper: time(s) & Scissors: time(s) & Rock: time(s).

(Figure 2) ps. No selections, no play!



Sharon, Welcome to play Paper Scissors Rock game!

Select One:		Computer
● Paper ○ Scissors ○ Rock	Go!	Rock
Results:	win	

History: You have played total: 7 time(s)!
Paper: 4 time(s) & Scissors: 2 time(s) & Rock: 1 time(s).

lose tie tie lose tie lose win

(Figure 3)

Requirement:

You will need at least 1 html, 1 CSS file and 1 js file (if you can condense your scripts to less than 15 lines, go ahead to just use inline scripts.)

1. A assgin3 home page (named **assign3.html**). This page should include:

- a HEAD section with links and scripts linkages
- a BODY section with
  - Radio buttons for users to select paper/scissors/rock.
  - A button to initiate computer random generated paper/scissors/rock.
  - Text to display the results of win/lose/tie.
  - Text to display total play counts.
  - Text to display each selection counts (of computer generates output count)

2. A JavaScript (named **game.js**). Again, if you can condense all your functions and statements less than total 15 lines, you don’t need to submit a separate js file, you can use inline scripts.

- A `play()` function: will auto-generate paper/scissors/rock randomly, and display the generated paper/Scissors/Rock under Computer column. Hint: *declare an array, and use* *`Math.random()`*
- A `compare(yourInput, computerGen)` function that returns win/lose/tie result, and display on Results row and update all counters and logs in the history section.
- You should not let user play before they select one option, you may prompt instruction to user.

3. A CSS style sheet (named **game.css**). You can style any fonts, even images on your website. Minimally, you should include:

   - Set result text font-weight, paddings, and background color accordingly.

Note: You must use a text editor or IDE to write your HTML/CSS/JS.

Evaluation:
0: no submission.
60-70: missing multiple implementation.
80: erroneous implementation.
90: minor logic error, no user prompts.
95: minor errors (html or css criteria not complied)
100: all the criteria comply with requirements per requirement.

Bonus: There will be 5 points bonus for this assignment: if you can compute and display win% for each option, you get 3pt; exercise your creativity to make this game “look” more appealing, you get another 2 pt.

Submission:

1. Bundle everything into a zip file. Name it *YourFirstName_LastName_Assignment3* (i.e. Sharon_Hsiao_Assignment3.zip) Submit it through Camino.
2. Due on **2021/11/18 (Thursday) 10am**.