April 4, 2018

UCF /Orlando Coding Bootcamp

Week 3 home work assignment –The Trivia Game

On it’s face, this should not have been a terribly difficult. Plus we were handed several suggestions:

+Set up the questions & answers as an array of objects.

+++However, it turns out there are still many different ways to do this & obviously picking the “right” one dictates the commands best suited to manipulate it.

+++The examples found in the course work showed a specific & simple way of accessing elements in these arrays but I could never get them to work.

+++On top of this, despite looking at many other solutions & code examples on-line, I was unable to output array elements to the screen. Thinking it was a mis-match between the command chosed & the array structure, I changed the structure but still couldn’t make it work.

+++ I could have simply hard-coded the array elements into output statements & perhaps technically met the rubric. However, I felt it would be a waste of time & tuition to defy standard programming practices to this degree.

I spent to much time on the above that I never dig in & developed the rest of the code. I figured that since it was so simple, I could even wait until the last minute but the last minute is here & the arrays & outputs for them still do not work.

Pseudo code of the logic would simply be:

Output banner, introduction & rules –probably in a Bootstrap-driven /JumbotronCSS with some HTML.

Add some color, change the font text with CSS

The logc would be written in a combination of Javascript & JQuery…

Use banner at top of page to show total questions correct, total questions lost, percentage correct

1. Use “For loop” to increment through each array & pace each array element in a corresponding button;
2. If maximum questions not reached
   * 1. Output question
     2. Output buttons filled in with a., b., c., d.
     3. Output instructions to the user to pick one of the choices
     4. Start a 30 second countdown timer
     5. Show the timer counting down.
   1. If the user fails to click before time reached 0
      1. Stop the timer,
      2. display (:00) time remaining
      3. Disable the buttons
      4. Output the correct answer (use a different color &/or button shape)
      5. End of decision tree, return to #1
   2. Else if the user clicks before the timer expires
      1. Stop the timer,
      2. display time remaining
      3. disable the buttons
      4. output the correct answer (use a different color &/or button shape)
      5. Compare the guess to the correct answer
         1. If equal,
            1. Output the correct answer
            2. print winner
            3. wins++
            4. clear the question from the page
            5. update the display of wins vs losses
            6. return to #1
         2. if not equal
            1. Output the correct answer
            2. losses++
            3. clear the question from the page
            4. update the display of wins vs losses
         3. return to #`1

Lines 1-6 could be a function (since it needs to run once at the beginning & then again at the start of each iteration of the loop

Lines (b.i.) –(b.iv.) could be a function

Lines (c.v.1.a) –(c.v.1.e) could be a function

Additional logic if you put all the questions on one page vs the above assumes one question on the page at a time.