Quiz Submissions - Midterm

Top of Form

**Dan Singer (username: djs5435)**

**Attempt 1**

Written: Mar 7, 2019 11:01 AM - Mar 7, 2019 11:38 AM

**Submission View**

released: Apr 23, 2019 11:32 AM

|  |
| --- |
| Canvas and Bitmap Manipulation |

7 points

|  |  |  |
| --- | --- | --- |
| **Question 1** |  | 1 / 1 point |

Write code that *fills* a 200px wide, 200px high rectangle at (100,150) path in canvas - do NOT use the ctx.fillRect() convenience method. Assume the canvas drawing context object is in scope and named ctx

ctx.beginPath();

ctx.rect(100, 150, 200, 200);

ctx.closePath();

ctx.fill();

|  |  |  |
| --- | --- | --- |
| **Question 2** |  | 1 / 1 point |

A canvas \_\_\_\_\_\_\_\_\_\_ specifies a starting color, an ending color, and an area over which the color changes.

Question options:

|  |  |  |
| --- | --- | --- |
|  | context | |
|  | state property | |
|  | gradient | |
|  | current transformation matrix | |
|  | clipping region | |
| **Question 3** |  | 1 / 1 point | |

Write a single line of code that rotates the canvas context by 180 degrees. Assume the canvas drawing context object is in scope and named ctx

|  |  |  |
| --- | --- | --- |
| Answer: | ctx.rotate(Math.PI); |  |
| **Question 4** | | |  | 1 / 1 point |

When the following code is executed, what color will the stroke of the circle be?  
  
let canvas = document.querySelector('canvas');   
let ctx = canvas.getContext('2d');  
  
ctx.strokeStyle="green";  
ctx.save();  
ctx.strokeStyle="blue";  
ctx.lineWidth=10;   
ctx.save();  
ctx.strokeStyle="red";  
ctx.strokeRect(20,20,100,100);  
ctx.restore();  
ctx.arc(470, 70, 25, 0, Math.PI\*2, false);  
ctx.stroke();

Question options:

|  |  |  |
| --- | --- | --- |
|  | white (because all the stroke colors were added) | |
|  | black (because all the stroke colors were subtracted) | |
|  | red | |
|  | green | |
|  | blue | |
| **Question 5** |  | 1 / 1 point | |

Describe how to implement a*darkness filter* on an imageData array

Loop through the array in groups of four, and for each group, subtract a value from the first three elements of the group. For example, if the first four bytes were 255,255,255,1, then you could subtract 55 to end up with 200,200,200,1.

|  |  |  |
| --- | --- | --- |
| **Question 6** |  | 1 / 1 point |

\_\_\_\_\_\_\_\_\_\_ is the combining of visual elements from separate sources into single images.

Question options:

|  |  |  |
| --- | --- | --- |
|  | Compositing | |
|  | Effectuating | |
|  | Inheriting | |
|  | Delegating | |
|  | Splining | |
| **Question 7** |  | 1 / 1 point | |

Suppose your canvas is has dimensions of 500 x 500 pixels. How many elements will be in the typed array you get back from ctx.getImageData(0, 0, 500, 500) ?

Question options:

|  |  |
| --- | --- |
|  | 500 |
|  | 1000 |
|  | 250,000 |
|  | 750,000 |
|  | 1,000,000 |
| Web Audio | |

5 points

|  |  |  |
| --- | --- | --- |
| **Question 8** |  | 1 / 1 point |

Give 3 differences between a typical JS Array created with [ ], and a JS typed array such as a Uint8Array

Regular JS arrays can contain any data type, support adding elements, and support removing elements.

Typed arrays can only contain one data type, do not support adding elements, and do not support removing elements.

|  |  |  |
| --- | --- | --- |
| **Question 9** |  | 0 / 1 point |

**Normal Speech** falls in lower frequencies, roughly between 500 Hz to 2 kHz

Question options:

|  |  |  |
| --- | --- | --- |
|  | True | |
|  | False | |
| **Question 10** |  | 1 / 1 point | |

The primary paradigm used by WebAudio is that of a(n) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, where AudioNode objects are connected together to define the overall audio rendering.

Question options:

|  |  |  |
| --- | --- | --- |
|  | analyser node | |
|  | linear time domain | |
|  | audio context | |
|  | audio routing graph | |
|  | sequencer | |
| **Question 11** |  | 1 / 1 point | |

How did we use an *Analyser Node* in our Web Audio HW and project?

The Analyser Node is created from the audio context object and is connected to by the MediaElementSource object. The analyser node gives us access to the waveform and frequency data of an audio clip via getByteTimeDomainData and getByteFrequencyData, which can then be displayed through canvas shapes which change in appearance based on the audio data.

|  |  |  |
| --- | --- | --- |
| **Question 12** |  | 1 / 1 point |

Give an example of a web audio Effect Node, and describe what your example does.

The BiquadFilter node allows lowpass and highpass filters to be routed into the output of an audio source, allowing the highs or lows of an audio clip to be removed.

|  |
| --- |
| Variables, Scope, & Mutabilitiy |

1/2 point each = 3 points total

|  |  |  |
| --- | --- | --- |
| **Question 13** |  | 0.5 / 0.5 points |

What is the scope of variable myNum below?

<script>  
               let myNum = 0;

               function init(){  
                              console.log(myNum);  
               }  
</script>

Question options:

|  |  |  |
| --- | --- | --- |
|  | block | |
|  | function/local | |
|  | global | |
|  | property | |
|  | script | |
| **Question 14** |  | 0.5 / 0.5 points | |

What is the scope of the init function below?

<script>  
               function init(){  
                              console.log("Hi there!");  
               }  
</script>

Question options:

|  |  |  |
| --- | --- | --- |
|  | block | |
|  | function/local | |
|  | global | |
|  | property | |
|  | script | |
| **Question 15** |  | 0.5 / 0.5 points | |

What will be logged out when this code runs?

const x = {};  
x.name = "fred";  
console.log(x.name);

Question options:

|  |  |  |
| --- | --- | --- |
|  | "fred" | |
|  | A blank line | |
|  | undefined | |
|  | This code will produce an error before the console.log() executes | |
| **Question 16** |  | 0.5 / 0.5 points | |

What is the scope of variable myNum below?

<script>  
               function init(){  
                              if(true){  
                                             for(var i=0;i<5;i++){  
                                                            let myNum = 0;  
                                             }  
                              }  
               }  
</script>

Question options:

|  |  |  |
| --- | --- | --- |
|  | block | |
|  | function/local | |
|  | global | |
|  | property | |
|  | script | |
| **Question 17** |  | 0.5 / 0.5 points | |

What is the scope of variable myNum below?

<script>  
               var myNum = 0;

               function init(){  
                              console.log(myNum);  
               }  
</script>

Question options:

|  |  |  |
| --- | --- | --- |
|  | block | |
|  | function/local | |
|  | global | |
|  | property | |
|  | script | |
| **Question 18** |  | 0.5 / 0.5 points | |

What will be logged when this code runs?

const y = 1;  
y++;  
console.log(y);

Question options:

|  |  |
| --- | --- |
|  | This code will produce an error before the console.log() executes. |
|  | 2 |
|  | 1 |
|  | undefined |
| Browser DOM | |

|  |  |  |
| --- | --- | --- |
| **Question 19** |  | 1 / 1 point |

Write code that selects the first paragraph on an HTML page and changes its text content to “Hello!”

document.querySelector("p").innerHTML = "Hello!";

|  |
| --- |
| The button in the HTML below will do nothing when clicked |

<!DOCTYPE html>  
<html>  
        <head>  
                <title>Example 2</title>  
        </head>  
        <body>  
                <button>Click Me</button>  
                <script>  
                        let button = document.querySelector("button");  
                        button.onclick = doStuff();  
                        function doStuff(){  
                                console.log("Do Stuff!");  
                        }  
                </script>  
        </body>  
</html>

Answer the questions below about this situation.

|  |  |  |
| --- | --- | --- |
| **Question 20** |  | 0.5 / 0.5 points |

Why?

button.onclick is assigned the result of doStuff, rather than the actual function doStuff. Therefore, onclick will be assigned to null and not do anything when clicked.

|  |  |  |
| --- | --- | --- |
| **Question 21** |  | 0.5 / 0.5 points |

Describe how to fix the code.

Replace button.onclick = doStuff() with button.onclick = doStuff;

|  |  |  |
| --- | --- | --- |
| **Question 22** |  | 1 / 1 point |

Describe the difference between the HTML we see in the Chrome web browser when we "View Page Source" versus when we "Inspect" a page with the Web Inspector

With "View Page Source," the content is exactly as it was when retrieved from the web server. "Inspect" gives us a live view of the DOM and any changes that may have occured because of JavaScript code.

|  |
| --- |
| The following code throws an error: |

Error:

Uncaught TypeError: Cannot set property 'innerHTML' of null.

Code:

<!DOCTYPE html>  
<html>  
           <head>  
                     <title>Example 1</title>  
                      <script>  
                               function init(){  
                                      let myH1 = document.querySelector("#main");  
                                               myH1.innerHTML="Welcome to my favorite music page";  
                                       }  
                                      init();  
                      </script>  
           </head>  
           <body>  
                    <h1 id="main">This is my header</h1>  
           </body>  
</html>

Answer the questions below regarding this situation.

|  |  |  |
| --- | --- | --- |
| **Question 23** |  | 0.5 / 0.5 points |

Why does this error occur?

The page has not finished loading when myH1 is defined, so it will be null.

|  |  |  |
| --- | --- | --- |
| **Question 24** |  | 0.5 / 0.5 points |

Describe how to fix the error.

Replace init() with window.onload = init;

|  |
| --- |
| JavaScript Objects & Classes |

5 points

|  |  |  |
| --- | --- | --- |
| **Question 25** |  | 1 / 1 point |

The JavaScript class syntax introduces a new object-oriented inheritance model to JavaScript.

Question options:

|  |  |
| --- | --- |
|  | True |
|  | False |
| Consider this code as the initial value of obj: | |

<script>

let obj = {  
        x: 5,  
        y: 10,  
        move(){  
                this.x++;  
                this.y++;  
        }  
};

</script>

|  |  |  |
| --- | --- | --- |
| **Question 26** |  | 1 / 1 point |

What will be the output of this code?  If there is a JS error, then write ERROR

console.log(obj.speed);

|  |  |  |
| --- | --- | --- |
| Answer: | undefined |  |
| **Question 27** | | |  | 2 / 2 points |

Convert obj above to an ES6 class named **Mover**

* give it a constructor that takes x and y as values
* initialize those values as x and y properties
* implement a move() method (that causes x and y to increase by 1)
* implement a moveBack() method (that causes x and y to decrease by 1)

class Mover{

    constructor(x, y) {

        this.x = x;

        this.y = y;

    }

    move() {

        this.x++;

        this.y++;

     }

    moveBack() {

        this.x--;

        this.y--;

    }

}

|  |  |  |
| --- | --- | --- |
| **Question 28** |  | 1 / 1 point |

Create a new instance of **Mover** named car, with x and y equal to 0

let car = new Mover(0, 0);

|  |
| --- |
| More JavaScript |

6 points

|  |
| --- |
| JavaScript Functions |

|  |  |  |
| --- | --- | --- |
| **Question 29** |  | 1 / 1 point |

Declare a JavaScript function named multiply that accepts two numbers as arguments. This function will multiply the two numbers together and return the result.

function multiply(a, b) {

    return a \* b;

}

|  |  |  |
| --- | --- | --- |
| **Question 30** |  | 1 / 1 point |

Now re-write the multiply function as an ES6 arrow function.

let multiply =  (a, b) => a \* b;

|  |  |  |
| --- | --- | --- |
| **Question 31** |  | 2 / 2 points |

Given the following **odds** array, write JS code that will iterate over the **odds** array, add all even numbers to the **evens** array and remove the even numbers from the **odds** array. The final version of the **odds** array should only contain odd numbers. The final version of the **evens** array should only contain the even numbers.

let odds = [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ];  
let evens = [ ];

for (let i = 0; i < odds.length; ++i) {  
    if (odds[i] % 2 == 0) {

        evens.push(odds[i]);

        odds.splice(i, 1);

        i--; // avoid skipping an element

    }

}

|  |
| --- |
| IIFE |

|  |  |  |
| --- | --- | --- |
| **Question 32** |  | 0.5 / 0.5 points |

What does IIFE stand for?

|  |  |  |
| --- | --- | --- |
| Answer: | Immediately Invoked Function Expression |  |
| **Question 33** | | |  | 0.5 / 0.5 points |

What's the primary benefit of using an IIFE in your JS code?

It keeps variables and functions outside of the global scope, leaving the data more protected.

|  |  |  |
| --- | --- | --- |
| **Question 34** |  | 1 / 1 point |

Add the code necessary for the myLib() function to “export” the showCounter() function (ONLY) so that it is visible and usable from the outside of the myLib() function

function myLib(){  
            const secret = 42;  
            let counter = 10;

            function showCounter(){  
                        return counter;  
            }

            // your code will go here.

}

let lib = myLib();  
console.log(lib.showCounter());

return { showCounter };

|  |
| --- |
| Other |

3 points

|  |  |  |
| --- | --- | --- |
| **Question 35** |  | 1 / 1 point |

When we look for common keywords in a text document or web page, we will often want to exclude words such as exclamations, personal pronouns and prepositions because they don’t really tell us much about the contents of a page or document. What is the term for this category of words?

|  |  |  |
| --- | --- | --- |
| Answer: | stopWords |  |
| **Question 36** | | |  | 1 / 1 point |

The following quote represents which creative art movement?

"The official belief in the infallibility of reason, logic and causality seemed to us senseless - as senseless as the destruction of the world and the systematic elimination of every particle of human feeling. This was the reason why were were forced to look for something that would re-establish our humanity. What we needed to find was a balance between heaven and hell. A new unity combining chance and design.”

Question options:

|  |  |  |
| --- | --- | --- |
|  | Pop | |
|  | Art Nouveau | |
|  | Minimalism | |
|  | Dada | |
|  | Rococo | |
| **Question 37** |  | 1 / 1 point | |

Describe the JavaScript *prototype chain*

Each object contains a prototype property which links to an object containing functions. Whenever an object method is invoked, it's first checked if that object contains the method. If it doesn't, the compiler will check the prototype object, the prototype's prototype, and so on until that method is found. Once it's found, that method is invoked.

|  |
| --- |
| Extra Credit (Value types vs. Reference Types) |

1/2 point each

|  |  |  |
| --- | --- | --- |
| **Question 38** |  | 0.5 / 0.5 points |

See the code below. What will be logged for the values of x and y?   
  
let x = "1";  
let y = x;  
x = "2";  
console.log(x);  
console.log(y);

"2"

"1"

|  |  |  |
| --- | --- | --- |
| **Question 39** |  | 0.5 / 0.5 points |

See the code below. What will be logged for the values of x and y?

let x = [1,2,3];  
let y = x;  
x.push(4);  
y.push(5);  
console.log(x);  
console.log(y);

[1,2,3,4,5]

[1,2,3,4,5]

|  |
| --- |
| **Attempt Score:**  33 / 33 |
| **Overall Grade** (highest attempt)**:**  33 / 33 |

Bottom of Form