

⚠ Students have either already taken or started taking this quiz, so be careful about editing it. If you change any quiz questions in a significant way, you may want to consider regrading students who took the old version of the quiz.

Points 45 ✔ Published



Details

Questions

Show Question Details

⋮ Javascript Arrays

3 pts

Assume you are given a JavaScript array named `arr` and function named `getBigRandom()` that returns a random number between 0 and 9000000000000000.

Consider the following code:

```
let arr = [1, 2, 3, 4, "hello", true, false, true, NaN];
let t = typeof (arr[getBigRandom()]);
console.log(t);
```

that prints out the type of an entry in the array chosen by `getBigRandom`;

If you execute the code a million times, what would you expect to be the most frequently returned JavaScript type? Explain your answer.

General answer comments

Most calls to `getBigRandom()` will return an integer that is larger than 8, which will mean that it will be out of bounds for `arr`. In JavaScript, if you try to access an element that is out of bounds, it returns `undefined`. `typeof (undefined)` is `'undefined'`, so that is what we will expect to be printed most frequently.

You have some JavaScript code that contains the expression

```
foo()()()()() === foo
```

Write a function foo that would cause the above statement to return true.

General answer comments

BEST ANSWERS

```
let foo = () => foo;
```

OR equivalently...

```
function foo () {  
  return foo;  
}
```

This works because each successive call to foo returns foo, so it can be called again, and no matter how many times it is chained, the return value will be 'foo', making the comparison:

```
foo()()()()() === foo;
```

evaluate to true. Some submissions were more complicated, but still produced the correct result (e.g. five nested functions).

COMMON MISTAKES

- Many incorrect submissions attempted a more complicated solution and made mistakes in the process, such as having the wrong number of nested functions, or naming all of the nested functions "foo", which caused the comparison to fail.
- Many incorrect submissions also wrote a 'foo' that returned 'this' instead of 'foo'. That answer is incorrect because 'this' is undefined in the global scope.
- Many incorrect submissions had 'foo' return something other than a function, like true or 1 or 0, or even the string "foo". This is incorrect, because in order for foo to be "chained", it has to return a function.

You encounter a large web application in which every class name starts with a prefix encoding the pathname in the source code repository containing the CSS file with rules targeting the class. For example, the CSS classes referred to in `product/module/foo.css` all have names starting with `product-module-foo`. Describe the issue in CSS the developers are trying to mitigate by this naming convention.

General answer comments

CSS classes by default are in the global scope/namespace, so this naming convention is to prevent classes with the same name in different parts of the web app from conflicting.

⋮ HTML

2 pts

Using JavaScript in the browser, it is trivial to write a program that reads input from the user and displays it by writing HTML into the DOM. For example, writing something as simple as

```
<b> characters </b>
```

would display the user's input characters in bold.

The simple approach above works for most characters the user could type but there are several nonalphanumeric characters that don't show up correctly. Explain the problem and what you would have to do to correctly display these characters.

General answer comments

There are a few characters that are used to delimit markup (such as `&`, `<`) in HTML. They will not display correctly because browser will mistake them as part of the markup language.

In order to display the user input correctly in HTML, we need to preprocess the input by replacing those characters with their corresponding escaped version (example: `&` -> `&` ;
).

URLs

2 pts

Identify the parts of the following URL:

`https://en.wikipedia.org/wiki/Internet#Terminology`

- #Terminology [part1]
- /wiki/Internet. [part2]
- https [part3]
- en.wikipedia.org [part4]

Show Answers for

1 answer

Host name

JavaScript Functionss

2 pts

Concisely explain the difference between the return values of `function.bind()` and `function.call()`.

General answer comments

`bind()` returns a copy of the passed in function with its 'this' assigned to whatever was passed in as an argument to `bind()`, but does not actually execute the function. `call()` immediately executes the provided function with 'this' set from the arguments and returns whatever the function normally returns.

URLs

3 pts

Consider a complex web application with many hyperlinks going between different views in the website and fetching many different images, CSS, and JavaScript files from the website. Would it be possible to move the entire web application to run on from a

different hostname in the URL? Take one of the two positions and provide a justification for your answer.

- Yes, the switching could be done without changing any of the URLs in the web application.
- No, some changes would be required, no matter how it was built.

General answer comments

If you constructed a web application where all of the URLs used in it (i.e.. hyperlink, script tags, link tags, img tags, etc.) didn't have a host name part (what we called absolute and relative URLs) it could easily be hosted on some other host. All URLs would end up using the host name of where the web app was loaded from. Such a web application wouldn't need to be changed if moved.

⋮ HTML/CSS

2 pts

A new intern joins the company you are working for and on the first day, they run a script that effectively inlines all the CSS files. The script changed all the tags of the form:

```
<link rel="stylesheet" type="text/css" .. />
```

in the HTML file with the contents of the specified CSS file inside a

```
<style type="text/css"> ... </style>
```

Although the web application continues to work, explain why this is not an improvement with examples of what would be worse off if the change was left in.

General answer comments

- DRY is violated. The same CSS definitions have to be in multiple HTML files now.
- It is harder to maintain. A CSS change that affects multiple HTML files has to be done multiple times.

The key idea here is that a change that one CSS change now involves changing and keeping in sync multiple HTML files.

Full credit is not given for:

- Vague answers. Saying it is bad coding style or harder to maintain without explaining why it is considered that way will not receive full credit.
- Assume that elements must be styled individually. CSS selectors and classes can still be used in inline stylesheets, therefore answers that assume that all styles must be done inline per element are not given credit.
- HTML is not meant for styling, or CSS files are where style should live. This is technically true but it does not explain why it is not considered good form to inline styles.
- Switching of CSS stylesheet becomes harder. This is technically true, but not a huge reason for separating CSS and HTML. It is actually very rare for a website to switch between CSS stylesheets for the same page, outside of CSS styling demos.
- Inline styles will overwrite external styles and be harder to debug. This is not an issue with the scenario because all styles have been added to their HTML pages and there are no external styles.

JavaScript idioms

4 pts

In lecture we talked about two common JavaScript idioms using the "short-circuit evaluation" operators `||` and `&&`. The two examples were:

```
hostname = hostname || "localhost";
```

```
var prop = obj && obj.propname;
```

(a) For each of the examples, explain what useful thing it does for the programmer. Your answer should focus on what the code does and not how JavaScript `&&` and `||` operators work.

(b) For each of the examples, describe what would happen if you accidentally used the other operator in its usage pattern (e.g. `hostname = hostname && "localhost";`)

General answer comments

Part (a): 1. Can assign a default value to hostname 2. Can assign object property if object is defined

Part (b) 1. hostname would assign to "localhost" if hostname is defined, or remain undefined otherwise 2. var prop will be assigned to either obj or raise an exception

⋮ HTML

4 pts

As HTML has gotten older some of the tags useful in the early days and fallen out of favor and their use is discouraged. For each of the following HTML tags, describe if they are in or out of favor and if out of favor describe why?

- `<p>`
- ``
- `<h1>`
- ``

General answer comments

`` and `<p>` have not fallen out of favor but `` and `<h1>` have. `` indicates an image. `<p>` indicates a paragraph. They both have important roles in organizing the document.

On the other hand, `` and `<h1>` define a preset of style on embedded texts, but such style can be more precisely defined through CSS. It is better practice to include style definitions in CSS.

⋮ JavaScript this

5 pts

Consider the following JavaScript code:

```

"use strict";
function MyClass() {
}
MyClass.prototype.myMethod = function (a) {
  let f1 = function () { return this; };
  let f2 = () => this;
  if (a === 0) return f1() === f2();
  if (a === 1) return f1() === f1();
  if (a === 2) return f2() === f2();
  return false;
}
var m = new MyClass;
console.log(m.myMethod(0));
console.log(m.myMethod(1));
console.log(m.myMethod(2));

```

Describe what the three boolean values outputted to the console log would be and explain the reasoning.

General answer comments

The key is to look at the definition of the two functions f1 and f2. Both functions return the **this** keyword. This ends up working differently if the function is declared using array syntax vs function keyword. Later in the function when we directly call f1 and f2 (not as a method function) the this of f1 will be undefined (note without the use strict it would be the window object in the browser). The **this** of the f2 function is inherited from the containing scope. Since myMethod() is invoked as a method on the object "m", the this returned by f2 will be 'm'. From this we can predict the output:

- f1() === f2() undefined === m, false
- f1() === f1() undefined === undefined, true
- f2() === f2() m === m, true

JavaScript has a way of allocating objects (i.e., the new keyword) and modifying objects by adding and remove properties but lacks a way for the programmer to explicitly deallocate/free objects. Explain how long-running JavaScript programs don't run out of memory without a way for the programmer to free the memory allocated for the objects.

General answer comments

Long running programs work because JavaScript has a garbage collector that deallocates objects when there is no reference to them.

⋮ JavaScript Number

2 pts

Explain the reason why code that manipulates large integers that works fine in Java and C++ can break in JavaScript. For example:

```
number1 = 10000000000000000;  
number2 = number1 + 1  
(number1 == number2)
```

has the comparison return false in Java and C++ but true in JavaScript.

General answer comments

Unlike in Java and C++, all numbers in JavaScript are stored as 64-bit floating points. 53 of these bits are used to represent integers, so JavaScript has a maximum safe integer value of $2^{53} - 1$. Integers that are larger than this number are not guaranteed to behave in expected ways during manipulations, partially due to overflow. In the example above, adding 1 to a number larger than the maximum safe integer value results in overflow that causes number 2 to be considered identical to number 1.

In Java and C++, there are types that allow numbers larger than 53-bits to be represented and manipulated as expected.

⋮ CSS capabilities

6 pts

You are given an HTML document containing many paragraphs tags each with a unique id attribute (e.g. `<p id="p1">`). For all the following changes to the display of the document, state if the change could be made using only CSS rules without changes to the HTML:

1. Make some but not all paragraphs displayed in bold red.
2. Remove some but not all paragraphs from the displayed document.
3. Reorder some paragraphs in the displayed document.
4. Make a word that appears in the document show up redacted (i.e. every occurrence of the word covered by a black box).

State and briefly explain your answer for all 4 changes.

General answer comments

1. Can be done with pure CSS by selecting the desired paragraphs by id and using the color and font-weight properties.
2. Can be done with pure CSS by setting some paragraphs to be `display: none`.
3. Can reorder paragraphs using a flex container (perhaps on the body) and setting the order property.
4. Cannot be done using CSS only. Selecting a specific word requires that word to be enclosed in HTML tags, which is not guaranteed in the problem description.

⋮ CSS position

3 pts

Consider an HTML document where every tag has a CSS `"position: absolute;"` rule on it. If you forget to put a position property on a tag it would not be considered an error by the browser. Describe how the browser would determine the position of the item with the forgotten positioning rule?

General answer comments

Full credits will be given to answers that correctly describe the expected rendering output. The element without "position: absolute" should show up either as the child of the <body> or of another element that is absolute positioned. It will be positioned using "position: static" and, in the absence of other rules, be at the top left corner of its parent.

Partial credits for:

- [+1] Mentioning "position: static"
- [+1] Some description of how "position: static" actually positions the element using the normal flow (e.g. by looking at the parent element or the previous non-absolute element)
- [+1] Identifying the two possible parent elements that the position is based on

[+ New Question](#)

[+ New Question Group](#)

[🔍 Find Questions](#)

Notify users this quiz has changed

[Cancel](#)

[Save](#)
