Quiz #1

Q1 Markup Languages

4 Points

Which of the following programming language features would you expect to **not** appear in a language that billed itself as pure **Markup Language**:

□ Variables

□ Subroutines

□ Condition statement (e.g. if)

□ Iterative statement (e.g. for/while)

Briefly explain your answer:

Enter your answer here

Save Answer

Q2 HTML

3 Points

Explain the problem and the workaround when writing an HTML document where the content is "**Examples of how to write an HTML document**". Include an example in your answer.

Enter your answer here

Save Answer

Q3 New HTML versions

3 Points

Explain why it is common for a web app developer to follow the latest ECMAScript standards for their JavaScript code yet not use the features in the latest HTML standards? In other words, why would they need to be careful about using new HTML features but not new JavaScript features?

Enter your answer here

Save Answer

Q4 CSS

6 Points

Consider a simple CSS rule like:

```
p {
 color: green
}
```

Q4.1

3 Points

Explain how it is possible for some `<p` elements in an HTML document to not be colored green even with the above rule.

Enter your answer here

Save Answer

Q4.2
3 Points

Explain how it is possible for some elements using a tag other than `<p` to be colored green by the above rule.

Enter your answer here

Save Answer

Q5 URLs
4 Points

Assume you point a browser at the website:

```
http://www.example.com/a/b/c.html
```

and it downloads an HTML document.

The HTML document loaded contains four hyperlinks. For each of the hyperlinks, clicking on the link immediately after the initial download will direct the browser to a URL. State that URL for each case.

```
<a href="http://www.example.com/a/b/c.html#d">Link</a>
```

Enter your answer here

```
<a href="/a/b/c.html#e">Link</a>
```

Enter your answer here

```
<a href="a/b/c.html">Link</a>
```

Enter your answer here

```
<a href="#f">Link</a>
```

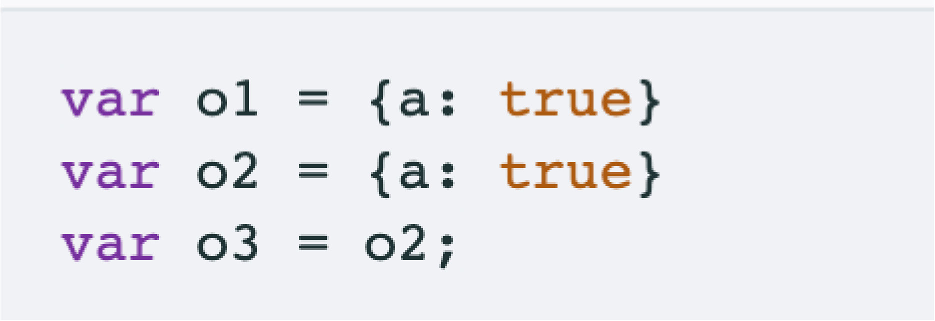
Enter your answer here

Save Answer

Q6 JavaScript objects

5 Points

Assume you have a JavaScript file that begins with the following definitions:



```
var o1 = {a: true}
var o2 = {a: true}
var o3 = o2;
```

Q6.1

3 Points

What does the following JavaScript print:

```
console.log(o1 === o2);
```

○ true

○ false

```
console.log(o1 === o3);
```

○ true

○ false

```
console.log(o2 === o3);
```

○ true

○ false

```
console.log(JSON.stringify(o1) == JSON.stringify(o2));
```

○ true

○ false

```
console.log(JSON.stringify(o1) == JSON.stringify(o3));
```

○ true

○ false

```
console.log(JSON.stringify(o2) == JSON.stringify(o3));
```

○ true

○ false

Save Answer

Q6.2

2 Points

Assume you execute the following assignment state to the above code:

```
o3.a = false;
```

What does each of the following statements print:

```
console.log(o2 === o3);
```

○ true

○ false

```
console.log(JSON.stringify(o1) == JSON.stringify(o2));
```

○ true

○ false

```
console.log(JSON.stringify(o1) == JSON.stringify(o3));
```

○ true

○ false

```
console.log(JSON.stringify(o2) == JSON.stringify(o3));
```

○ true

○ false

Save Answer

Q7 Javascript scopes

8 Points

Consider the following JavaScript code fragments. When executed they will execute five of the `console.log` statement.

Q7.1

3 Points

```
try { console.log(a); } catch(err) { console.log("ERROR"); }
try { console.log(b); } catch(err) { console.log("ERROR"); }
{
   let a = 1;
   let b = 2;
   console.log(a + b);
}
try { console.log(a); } catch(err) { console.log("ERROR"); }
try { console.log(b); } catch(err) { console.log("ERROR"); }
```

For each console.log prints select the output:

console.log 1

○ undefined

○ 1

○ 2

○ 3

○ ERROR

console.log 2

○ undefined

○ 1

○ 2

○ 3

○ ERROR

console.log 3

○ undefined

○ 1

○ 2

○ 3

○ ERROR

console.log 4

○ undefined

○ 1

○ 2

○ 3

○ ERROR

console.log 5

○ undefined

○ 1

○ 2

○ 3

○ ERROR

Save Answer

Q7.2

3 Points

```
try { console.log(a); } catch(err) { console.log("ERROR"); }
try { console.log(b); } catch(err) { console.log("ERROR"); }
{
   var a = 1;
   var b = 2;
   console.log(a + b);
}
try { console.log(a); } catch(err) { console.log("ERROR"); }
try { console.log(b); } catch(err) { console.log("ERROR"); }
```

For each console.log prints select the output:

console.log 1

○ undefined

○ 1

○ 2

○ 3

○ ERROR

console.log 2

○ undefined

○ 1

○ 2

○ 3

○ ERROR

console.log 3

○ undefined

○ 1

○ 2

○ 3

○ ERROR

console.log 4

○ undefined

○ 1

○ 2

○ 3

○ ERROR

console.log 5

○ undefined

○ 1

○ 2

○ 3

○ ERROR

Save Answer

Q7.3

2 Points

The following code ends up polluting the namespace of the enclosing scope. In the space above and below add a line of code that would remove this namespace pollution problem yet still execute the code.

Enter your answer here

```
{
    var a = 1;
    var b = 2;
    console.log(a + b);
}
```

Enter your answer here

Save Answer

Q8 JavaScript arrays

3 Points

Q8.1

2 Points

Consider the following JavaScript code:

```
var v = [];

v[0] = 0;
v[1] = 1;
v[1.5] = 1.5;
v['foo'] = 'foo';

console.log(v[0] + v[1] + v[1.5], v.foo);
```

The console.log statement will output `2.5 foo`. Explain the properties of JavaScript arrays that allow this code to work even though most other programming languages would consider the last two lines some kind of error.

Enter your answer here

Surprisingly the same output will occur if the declaration of `v` is changed to be:

```
var v = {};
```

so `v` is on an object. Explain how this works like an array when v is an object.

Enter your answer here

Save Answer

Q8.2

1 Point

Arrays in JavaScript are described as being **polymorphic**.   Are objects in JavaScript also polymorphic? Justify your answer.

Enter your answer here

Save Answer

Q9 JavaScript classes

3 Points

Consider the following JavaScript code that creates an object of type Rectangle which is a subclass of Shape (not shown)

```
class Rectangle extends Shape {
  constructor(height, width) {
     super(height, width);
     this.height = height;
     this.width = width;
  }
  area() {
    return this.width * this.height;
  }
  static countRects() {
    // ...
  }
}
var r = new Rectangle(10,20)
```

At least three objects are created by this code and populated with properties. For each of the objects listed below, select which properties are added to the object:

The newly allocated instance of Rectangle

□ countRects

□ area

□ height

□ width

□ None of the above

The prototype object of the newly allocated instance of Rectangle

□ countRects

□ area

□ height

□ width

□ None of the above

The Rectangle class function object

□ countRects

□ area

□ height

□ width

□ None of the above

Save Answer

Q10 JavaScript closures

2 Points

For each of the following primitive JavaScript types, would it be possible for a JavaScript **closure** to hold a variable of that type.

□ undefined

□ number

□ string

□ boolean

□ function

□ object

Briefly explain your answer:

Enter your answer here

Save Answer

Q11 JavaScript this

2 Points

`this` variable is declared in every JavaScript function. Frequently it is either an object (for methods) or undefined (for non-method functions). Can it be other JavaScript types like numbers or strings or functions? If so, explain how this could happen. If not, explain why not.

Enter your answer here

Save Answer

Save All Answers

Submit & View Submission ❯