Quiz #2

Q1 DOM

8 Points

Assume you are given the following XHTML document:

```
<!DOCTTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Title</title>
  </head>
  <body>
    Body
    <div id="div1">
      Div1
    </div>
    <div id="div2" class="c1">
      Div2
      <div id="div3">
        Div3
        <div id="div4" class="c1">
          Div4
        </div>
      </div>
      <div id="div5" class="c2">
        Div5
      </div>
    </div>
  </body>
</html>
```

Answer the following questions about the JavaScript statements executed in a browser page with the above HTML rendered.

Q1.1

1 Point

```
let element = document.body.firstChild;
try { console.log(element.id); } catch (err) { console.log('ERROR')}
```

Select what the above code will write to the console log.

○ div1

○ div2

○ div3

○ div4

○ div5

○ ERROR

○ None of the above

Save Answer

Q1.2

1 Point

```
let element = document.getElementById('div3').parentNode;
try { console.log(element.id); } catch (err) { console.log('ERROR')}
```

Select what the above code will write to the console log.

○ div1

○ div2

○ div3

○ div4

○ div5

○ ERROR

○ None of the above

Save Answer

Q1.3

1 Point

```
let element = document.getElementsByClassName('c2');
try { console.log(element[0].id); } catch (err) { console.log('ERROR
```

Select what the above code will write to the console log.

○ div1

○ div2

○ div3

○ div4

○ div5

○ ERROR

○ None of the above

Save Answer

Q1.4

1 Point

```
let element = document.getElementById('div3').getElementsByClassName
try { console.log(element[0].id); } catch (err) { console.log('ERROR
```

Select what the above code will write to the console log.

○ div1

○ div2

○ div3

○ div4

○ div5

○ ERROR

○ None of the above

Save Answer

Q1.5

2 Points

```
let element = document.getElementById('div3');
try {element.style.display = "none" } catch (err) { console.log('ERR
```

Select what text will be displayed by the browser rendering the page.

○ Body Div1 Div2 Div3 Div4 Div5

○ Body Div1 Div2 Div4 div5

○ Body Div1 Div2 Div5

○ Body Div1 Div2

○ Body Div1

○ console.log of "ERROR"

Save Answer

Q1.6

2 Points

Consider what would happen to the display if the following JavaScript was executed:

```
document.getElementById('div3').style.position = "absolute";
```

Explain why setting the position to "absolute" without setting anything else will not change the location where 'div3' is rendered. In other words, explain why "div3" doesn't move.

Enter your answer here

Save Answer

Q2 Event handling

6 Points

Assume you have the same XHTML document as Question 1:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Title</title>
  </head>
  <body>
    Body
    <div id="div1">
      Div1
    </div>
    <div id="div2" class="c1">
      Div2
      <div id="div3">
        Div3
        <div id="div4" class="c1">
          Div4
        </div>
      </div>
      <div id="div5" class="c2">
        Div5
      </div>
    </div>
  </body>
</html>
```

Assume you have registered the following event handlers for 'click' events.

```
function handlerCapture(e) {
   console.log("Capture",
      e.currentTarget.nodeName,
      e.currentTarget.id,
      e.target.id);
}
function handlerBubble(e) {
   console.log("Bubble",
      e.currentTarget.nodeName,
      e.currentTarget.id,
      e.target.id);
}
```

on every tag in the HTML document (the `html`, `body`, and `div` tags) with `handleCapture` being registered as a capture mode handler and `handleBubble` being registered as a bubble mode handler.

If the user clicks on the "Div3" text, show what would be outputted to the console log. Provide your answer in the boxes below with each box being a line from a console.log. Leave any extra boxes empty.

The lines output to the console will have four parts:

Item	Possible values
Mode	Capture or Bubble
e.currentTarget.nodeName	`html`, `body`, or `div`
e.currentTarget.id	`div1`..'`div5` or empty string
e.target.id	`div1`..'`div5` or empty string

For example, the output line would look like: `Bubble body div1 div2`

Enter your answer here

Enter your answer here

Enter your answer here

Enter your answer here

Enter your answer here

Enter your answer here

Enter your answer here

Enter your answer here

Enter your answer here

Enter your answer here

Save Answer

Q3 setTimeout

2 Points

Assume you execute the following JavaScript code in a browser.

```
console.log('log1');
setTimeout(() => console.log('log2'), 10);
console.log('log3')
setTimeout(() => console.log('log4'), 1);
```

Select the answer below that best describes the order that the console.log statements will appear in the log:

○ log1, log2, log3, log4

○ log1 log3, log2, log4

○ log1 log3, log4, log2

○ Can't tell - it's timing dependent so could be multiple different ones

Save Answer

Q4 Promises

7 Points

Assume you are given the following code using JavaScript promises:

```
// Returns a promise that resolves to 1
async function One() {
   return 1;
}

// Returns a promise that resolves to 2
async function Two() {
    return 2;
}

// Takes two promises op1,op2 resolves and adds them rogether
async function Sum(op1,op2) {
   return await op1 + await op1
}

let valuePromise = Sum(One(), Two());
valuePromise.then((val) => console.log('val', val));
```

Given the differing opinions on the JavaScript community, some library provided both promise and callback versions of routines. Show your understanding of promises and callbacks by writing the callback versions of the above functions. The callback versions should execute

```
Sum(One, Two, (val) =>  console.log('val',val) );
```

with the same output as the promise version.

Q4.1 Promises

2 Points

Show what a callback version of function One and function Two would look like.

```
function One(doneCallback) {
```

Enter your answer here

```
}

function Two(doneCallback) {
```

Enter your answer here

```
}
```

Save Answer

Q4.2

5 Points

Show what a callback version of function Sum would look like. You can assume that doneCallbacks of `op1` and `op2` will fire within a bounded and short amount of time.

```
function Sum(op1, op2, doneCallback) {
```

Enter your answer here

```
}
```

Save Answer

Q5 Concurrency

5 Points

Q5.1

2 Points

Statements like:

`counter = counter + 1;`

are problematic in many language environments such as Java and C++ in that its implementation will be broken down into multiple primitive steps like:

1. Read counter from memory into a CPU register
2. Increment the CPU register
3. Store the CPU register back into memory

The programmer must ensure that this statement is not executed concurrently. A common approach is to employ locks that ensure the three steps can be executed without another execution of the statement interleaving.

Explain why the JavaScript language does not have this problem.

Enter your answer here

Save Answer

Q5.2

3 Points

In both `fs.ReadFileSync` in Node.js and the DOM's synchronous `XMLHttpRequest` the routine waits until the request has finished before returning rather than using the more standard callback interface. The documentation for both of these routines strongly advises against using them. Explain what problem they cause for the JavaScript runtime?

Enter your answer here

Include a description of how their use would harm a web application running in a browser.

Enter your answer here

Save Answer

Q6 React

4 Points

JSX code looks like the templated HTML that has long been used by web application frameworks. While most templated HTML systems process the HTML templates and output HTML that can be rendered on the browser, JSX does not output HTML. Describe what JSX outputs.

Enter your answer here

Demonstrate your understanding of JSX by showing what `<div id="div1"></div>` would output.

Enter your answer here

Save Answer

Q7 React

5 Points

There are multiple ways code in a React component can update component state. For example, if the component had a state variable `counter`, it could be updated by "direct assignment":

```
this.state.counter = this.state.counter  + 1;
```

or by "calling setState":

```
this.setState({counter: this.state.counter + 1}};
```

Answer the following questions:

Q7.1

3 Points

React allows both forms of updating state but dictates that the "direct assignment" method can only be used in the component constructor and the "calling setState" must be used everywhere else. Explain why React has this requirement.

Enter your answer here

Describe what would happen if you incorrectly used the "direct assignment" method somewhere you were supposed to use the "calling setState" method.

Enter your answer here

Save Answer

Q7.2

2 Points

Another difference between the two approaches to setting component state is what happens if you look at state immediately after the operation. For example:

```
this.state.counter = this.state.counter  + 1;
console.log(this.state.counter);
```

and

```
this.setState({counter: this.state.counter + 1}};
console.log(this.state.counter);
```

will result in different things output to the console log. Explain why.

Enter your answer here

Save Answer

Q8 ServerComm

2 Points

Both REST and GraphQL require the backend developer to export the data as collections of resources with each type of resource having a specified set of properties. For a backend where the resources have a large number of properties, explain the benefit GraphQL would have over REST.

Enter your answer here

Save Answer

Q9 Testing/CSS

2 Points

In lecture we discuss two types of testing frameworks used on web applications:
1. unit testing.
2. end-to-end testing

Assume you start a new frontend job and are given to the assignment of testing the web app's handling of **CSS breakpoints**. Which testing framework or frameworks would be appropriate for the assignment.

○ unit testing

○ end-to-end testing

○ Both unit testing and end-to-end testing

State the rationale behind your answer.

Enter your answer here

Save Answer

Q10 Single Page Applications

2 Points

A web application is called a Single Page Application (SPA) since the browser only loads a single web page from the webserver. The SPA approach is popular for web applications these days.

Explain why in spite of having only a single page, SPA web applications add code to make that single page look to the user like a multiple page web app.

Enter your answer here

Save Answer

Q11 HTTP

2 Points

The HTTP protocol is implemented as a layer on top of the TCP/IP layer. The TCP/IP protocol allows the arbitrary transfer of bytes in both directions from the browser to the webserver and from the webserver to browser).

Although it is implemented on a bi-directional communication substrate, the HTTP protocol works much better for browser to web server communication than it does for web server to browser communication. For example, React components running in the browser can easily fetching model data from the web server anytime they want but there isn't good support for the web server to communicate information it discovers to the browser.
For example, It wouldn't work well to have your web app have an alarm notification that was implemented on the web server and let the browsers know when the alarm has gone off.

Explain what in the HTTP protocol causes this information transfer asymmetric even though it is layered on transport that can send bytes in both directions equally well.

Enter your answer here

Save Answer

Save All Answers

Submit & View Submission ›