Contents

2019 Past Paper

1. AJAX

AJAX stands for `Asyncronous Javascript And XML`

- The X part of this is no longer true due to the fact that most people have switched from using XML to JSON.

2. The three standard technologies for an SPA are the following

- HTML for presentation
- CSS for styling
- Javascript for computation

3. Address main differences between SPA's and static multipage websites

- SPA's allow for updating one component and not having to redraw the entire page, this leads to more efficient websites and quicker usage for end users
- SPA's send data between components rather than between pages, a webpage can be made up of multiple components
- Components in an SPA are loaded as different URL links over the template HTML (these URL's can contain multiple components).
- Data is fetched using AJAX/JSON.

4. Why might we choose SPA rather than static website?

- Because they want there users to have a faster and better user experience due to the fact that we can update small parts of the webpage without updating the whole webpage.
- To contain the client side rendering of information and UI updating all within the browser.

5. Is the event loop for Javascript in the browser single or multi-threaded?

- Single threaded

6. Give output of following code

```
1 console.log("1");
2 setTimeout(() => console.log("2"), 0);
3 console.log("3");
```

> Output: 1 3 2

7. Complete DOCTYPE

```
<!DOCTYPE HTML>
```

8. Provide css rule in line 8 to set color of input field to yellow

```
1  <style>
2    #score {
3      background-color: yellow;
4    }
5  </style>
```

9. If the user did not enter a value into the form, and simply pressed the 'Go' button, state the value that would be passed by default:

> 3 would be parsed in

10. `use strict` at the top of a function will enable strict mode, meaning that certain things JS will treated with more strict rules, such as undefined variables within the function. Leads to nicer code and conforms to more best practices.

11. Identify the part or parts of the code fragment that indicate that this is ES6 code

> Line 18: const addScore = () => {}

12. Complete the onclick attribute in the button element in line 15:

```
1  <button onclick='addScore()' name='button'>Go</button>
```

13. Complete the onScore() function, displays on line 10

```
1  function onScore() {
2    const a = document.getElementById("score").value;
3    document.getElementById("result").innerHtml = a;
4  }
```

14. Please tick or mark beside each URL which is considered to be the same origin under SOP as the URL for the ExperienceThis! app:

| http://experiencethis.co.nz:4941
| https://experiencethis.com,:4941
| https://experiencethis.co.nz:80

|x| https://experiencethis.co.nz/awesomevenus:4941

15. What are legitimate reasons for an application to make a cross-site request?

You may want to make a cross origin request to fetch some information from a certain domain, such as fetching a country from the rest countries API.

16. What is the primary purpose of CORS to safeguard the user of an application or the backend server?

CORS primary purpose is to prevent cross origin requests from being made to the domain by unfriendly services, *we may want to stop a service making large amount of requests as this could be detrimental to the service*.

17. Which of browser, server, or client app responsible for enforcing the restrictions imposed by the CORS headers?

The browser is responsible for enforcing CORS headers, the server specifies what headers to be checked

18. A request is blocked by cors, what data is received by the web app?

Error provided by the browser, the data received by the client side error will be blocked request.

19. Access-Control-Origin is an example of a forbidden header. What is a forbidden header in CORS?

A forbidden header in CORS is an HTTP error that the server specifies cannot be modified

20. Give a brief reason why CSRF is no longer included in the latest OWASP top 10

idk

21. provide a one line description of the XSS vulnerability

A XSS vulnerability is when an attacker can use script injection into the webpage with malicious intent

22. which are safe?