TW-12 TEAM LEAD VERSION

TEAMWORK

CLARUSWAY
WAY TO REINVENT YOURSELF

Meeting Agenda

- Icebreaking
- Questions
- Interview Questions
- Coffee Break
- Coding Challenge
- Video of the week
- Retro meeting
- Case study / project

Teamwork Schedule

Ice-breaking 5m

- Personal Questions (Study Environment, Kids etc.)
- Any challenges (Classes, Coding, studying, etc.)
- Ask how they're studying, give personal advice.
- Remind that practice makes perfect.

Team work 5m

- Ask what exactly each student does for the team, if they know each other, if they care for each other, if they follow and talk with each other etc.

Ask Questions 15m

1. What is the purpose of the async keyword in JavaScript?

A. To define an asynchronous function.
B. To specify a function that runs synchronously.
C. To create a generator function.
D. To indicate a callback function.

Answer: A

2. In an async function, what does the `await` keyword do?

A. To forcefully stop the function.
B. To pause the function execution until a promise is resolved.
C. Rejects a promise with an error.
D. Cancels the async function.

Answer: B Explanation: The await keyword is used to pause the function's execution until the promise it awaits is resolved.

3. Which of the following is a valid use case for using async/await in JavaScript?

A. Iterating through an array of numbers.
B. Declaring constants.
C. Defining a switch statement.
D. Fetching data from an API.

Answer: D Explanation: Async/await is commonly used when fetching data from APIs or performing other asynchronous operations.

4. What is the purpose of the Fetch API in JavaScript?

A. To create an array from a function.
B. To manipulate the DOM.
C. To send HTTP requests and receive responses.
D. To define JavaScript functions.

Answer: C

5. When using the Fetch API, what method is used to handle a response and extract JSON data?

A. .text()
B. .json()
C. .response()
D. .xml()

Answer: B

6. Which statement is true about error handling in async/await?

A. Errors are automatically handled by async/await.
B. Use .then() and .catch() to handle errors.
C. Errors in async/await cannot be caught.
D. Use try-catch blocks to catch errors.

Answer: D

7. What is the primary benefit of using async/await over traditional callbacks?

A. Simpler and more readable code
B. Easier handling of errors
C. Smaller bundle size
D. Improved performance

Answer: A

8. What does the setInterval() function do in JavaScript?

A. Halts the execution of the JavaScript code.
B. Delays the execution of a function for a specified time.
C. Executes a function repeatedly at a specified time interval.
D. Sets a timeout for a function to run only once.

Answer:C

9. How do you stop the execution of a function scheduled with setInterval()?

A. Using the return statement in the function.
B. By calling the clearInterval() function with the interval ID.
C. By setting the interval time to 0.
D. By using the break statement.

Answer: B

10. When using clearInterval(), what is the required argument?

A. The function to be cleared.wo
B. The time interval.
C. The DOM element
D. The interval ID returned by setInterval().

Answer: D Explanation: clearInterval() requires the interval ID returned by setInterval() as its argument to clear the interval.

11. What is the primary advantage of using Axios over the Fetch API?

A. Simplicity and ease of use.
B. Wider browser support.
C. Smaller bundle size.
D. Better performance.

Answer: A Explanation: Axios is often favored for its simplicity and ease of use compared to the Fetch API.

12. When would 'results shown' be logged to the console?

```
let modal = document.querySelector('#result');
setTimeout(function () {
  modal.classList.remove('hidden');
}, 10000);
console.log('Results shown');
```

A. after 10 second
B. immediately
C. after results are received from the HTTP request
D. after 10000 seconds

Answer: B

13. Why might you choose to make your code asynchronous?

A. to ensure that tasks further down in your code are not initiated until earlier tasks have completed
B. to make your code faster
C. to ensure that the call stack maintains a LIFO (Last in, First Out) structure
D. to start tasks that might take some time without blocking subsequent tasks from executing immediately

Answer: D

14. What is the HTTP verb to request the contents of an existing resource?

A. DELETE
B. PATCH
C. GET
D. CALL

Answer: C

15. Which method call is chained to handle a successful response returned by `fetch()`?

A. done()
B. catch()
C. then()
D. finally()

Answer: C

16. Which statement is applicable to the defer attribute of the HTML <script> tag?

A. defer allows the browser to continue processing the page while the script loads in the background.
B. defer causes the script to be loaded from the backup content delivery network (CDN).
C. defer blocks the browser from processing HTML below the tag until the script is completely loaded.
D. defer lazy loads the script, causing it to download only when it is called by another script on the page.

Answer: A

17. What will be logged to the console?

```
console.log('I');
setTimeout(() => {
  console.log('love');
}, 0);
console.log('Javascript!');
```

- [x] A.

```
I
Javascript!
love
```

- [ ] B.

```
love
I
Javascript!
```

- [ ] C. The output may change with each execution of code and cannot be determined.
- [ ] D.

```
I
love
Javascript!
```

Interview Questions 15m

1. What is 'callback hell' in the context of using callbacks, and what are its disadvantages?

Answer :
"Callback hell" is used when you have many nested callbacks in our code. Think of it as a layer cake of functions where each layer depends on the one above it. This makes our code look like a pyramid, often called the "Pyramid of Doom." The downside? It makes our code hard to read and even harder to debug. Plus, each new layer adds complexity, making future changes a headache.

2. Can you explain what asynchronous programming is?

Answer :
Asynchronous programming is a form of programming that allows for tasks to be completed out of order. This means that a program can start a task and then move on to other tasks before the first one is completed. This can be helpful in situations where a task might take a long time to complete, but the program doesn't need to wait for it to finish before moving on

3. How does async/await help with performance and scalability?

Answer :
Async/await can help improve performance and scalability by allowing your application to do other work while it is waiting for a task to complete. This can help avoid bottlenecks and keep your application responsive. Additionally, using async/await can help reduce the overall amount of code needed to be written, making your application easier to maintain.

4. What is the purpose of the Promise constructor in JavaScript?

Answer :
The Promise constructor is used to create a new Promise object. A Promise object represents an asynchronous operation, and can be used to track the status of that operation. A Promise can be in one of three states: pending, fulfilled, or rejected.

Coding Challenge 20m

- Random User Data Display

☕

Coffee Break 10m

☕

Video of the Week 5m

- Asynchronous Vs Synchronous Programming

Retro Meeting on a personal and team level 5m

Ask the questions below:

- What went well?
- What went wrong?
- What is the improvement areas?

Case study/Project 15m

- Weather App (JS-04)

Closing 5m

- Next week's plan
- QA Session
