

TW-007 TEAM LEAD VERSION (Sprint-5 Week-1)



CLARUSWAY
WAY TO REINVENT YOURSELF

Meeting Agenda

- ▶ Icebreaking
- ▶ Questions
- ▶ Interview Questions
- ▶ 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. Consider the following async function and its output. What will be displayed to the console when calling the `f()` function?

```
async function f() {  
  let result = 'first!';  
  let promise = new Promise((resolve, reject) => {  
    setTimeout(() => resolve('done!'), 1000);  
  });  
  
  result = await promise;  
  
  console.log(result);  
}  
  
f();
```

- A. first!
- B. done!
- C. JavaScript error
- D. Something else

Answer: B

2. What is the function to stop an interval timer?

- A. stopTimer
- B. clearInterval
- C. shutdownTimer
- D. clearTimer

Answer: B

3. What are 2 native functions to run code asynchronously in JavaScript ?

- A. timeout - setTimeout
- B. startInternal - setInterval
- C. setTimeout - setInterval
- D. interval - setInterval

Answer: C

4. What is the output of the code below?

```
let x = 0;

async function test() {
  x += await 2;
  console.log(x);
}

test();
x += 1;
console.log(x);
```

- A. 2 3
- B. 0 1
- C. 1 2
- D. 2 2

Answer: C

5. To get the first item from the array ("cooking") using array destructuring, how do you adjust this line?

```
const topics = ["cooking", "art", "history"]
```

- A. `const first = ["cooking", "art", "history"]`
- B. `const [] = ["cooking", "art", "history"]`
- C. `const [, first] = ["cooking", "art", "history"]`
- D. `const [first] = ["cooking", "art", "history"]`

Answer: D

6. What should the console read when the following code is run?

```
const [, , animal] = ["Horse", "Mouse", "Cat"]  
console.log(animal)
```

- A. Horse
- B. Cat
- C. Mouse
- D. undefined

Answer: B

7. Which collection object allows unique value to be inserted only once?

- A. Set
- B. Object
- C. Map
- D. Array

Answer: A

8. Why would you choose an asynchronous structure for your code?

- A. To use ES6 syntax
- B. To ensure that parsers enforce all JavaScript syntax rules when processing your code
- C. To ensure that tasks further down in your code aren't initiated until earlier tasks have completed
- D. To start tasks that might take some time without blocking subsequent tasks from executing immediately

Answer: D

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

- A. GET
- B. DELETE
- C. PATCH
- D. POST

Answer: A

10. What will be logged to the console?

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

A.

```
I  
love  
Javascript!
```

B. The output may change with each execution of code and cannot be determined.**C.**

```
I  
Javascript!  
love
```

D.

```
love  
I  
Javascript!
```

*Answer: C***11. How do you remove the property name from this object?**

```
const foo = {  
  name: 'Noah',  
};
```

A. delete name from foo;**B.** delete foo.name;

- C. del foo.name;
- D. remove foo.name;

Answer: C

Interview Questions

15m

1. What are Promises?

Answer: Promises are one way in handling asynchronous operations in JavaScript. It represents the value of an asynchronous operation. Promises was made to solve the problem of doing and dealing with async code before promises we're using callbacks.

Promises have 3 different states.

Pending - The initial state of a promise. The promise's outcome has not yet been known because the operation has not been completed yet.

Fulfilled - The async operation is completed and successful with the resulting value.

Rejected - The async operation has failed and has a reason on why it failed.

Settled - If the promise has been either Fulfilled or Rejected.

The Promise constructor has two parameters which are functions resolve and reject respectively. If the async operation has been completed without errors call the resolve function to resolve the promise or if an error occurred call the reject function and pass the error or reason to it.

2. What is a Callback function?

Answer: A Callback function is a function that is gonna get called at a later point in time.

```
const btnAdd = document.getElementById('btnAdd');

btnAdd.addEventListener('click', function clickCallback(e) {
  // do something useless
});
```

In this example, we wait for the click event in the element with an id of btnAdd, if it is clicked, the clickCallback function is executed. A Callback function adds some functionality to some data or event. The reduce, filter and map methods in Array expects a callback as a parameter. A good analogy for a callback is when you call someone and if they don't answer you leave a message and you expect them to callback. The act of calling someone or leaving a message is the event or data and the callback is the action that you expect to occur later.

3. What is prototype in javascript?

The prototype is an object that is associated with every functions and objects by default in JavaScript, where function's prototype property is accessible and modifiable and object's prototype property (aka attribute) is not visible.

Every function includes prototype object by default. The prototype object is special type of enumerable object to which additional properties can be attached to it which will be shared across all the instances of it's constructor function.

So, use prototype property of a function in the above example in order to have age properties across all the objects as shown below.

```
function Student() {  
    this.name = 'John';  
    this.gender = 'M';  
}  
  
Student.prototype.age = 15;  
  
var studObj1 = new Student();  
alert(studObj1.age); // 15  
  
var studObj2 = new Student();  
alert(studObj2.age); // 15
```

4. What is JSX? JSX is a syntax extension of JavaScript. It is used with React to describe what the user interface should look like. By using JSX, we can write HTML structures in the same file that contains JavaScript code.

Coding Challenge

20m

- [Coding Challenge: Stopwatch \(JS-024\)](#)



Coffee Break

10m



Video of the Week

5m

- [What Is React \(React js\) & Why Is It So Popular?](#)

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

Case study should be explained to the students during the weekly meeting and has to be completed in one week by the students. Students should work in small teams to complete the case study.

- [Movie Seat Booking App \(JS-07\)](#)
- [Weather-App \(JS-08\)](#)

Closing

5m

-Next week's plan

-QA Session
