

# TW-005 TEAM LEAD VERSION (Sprint-4 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. In the code below, what is the purpose of the *id* attribute?

```
<p id="warning">Be careful when installing this product.</p>
```

- A.** It establishes that *id* is a unique identifier in the document, used for styling CSS, scripting, and linking within a webpage.
- B.** It establishes that *id* is a unique identifier in the document, used for styling CSS and with Javascript code.
- C.** It establishes that *id* may be used for styling CSS several times per page.
- D.** It establishes that *id* is a unique identifier in the website, used for styling CSS, scripting, and linking within a webpage.

Answer: A

### 2. What do media queries allow us to do?

- A.** Stream video on our site.
- B.** Use different CSS based on screen size.
- C.** Do responsive design.
- D.** Changes all document sizes and feature.

Answer: B

**3. What is the difference between `<input type="submit" value="click me">` and `<button type="submit">Click me</button>`?**

- A. There is no difference. Both will render a button that submits a form.
- B. Both will submit a form. However, the `<button>` can have content other than text, like an image or nested HTML elements, while the `<input>` cannot.
- C. `<input type="button">` has been deprecated in HTML5. You should use the `<button>` tag instead.
- D. Both will submit a form. However, the `<input>` can have content other than text, like an image or nested HTML elements, while the `<button>` cannot.

Answer: B

**4. Which Object method returns an iterable that can be used to iterate over the properties of an object?**

- A. Object.get()
- B. Object.loop()
- C. Object.each()
- D. Object.keys()

Answer: D

**5. What is the value of `dessert.type` after executing this code?**

```
const dessert = { type: 'pie' };  
dessert.type = 'pudding';
```

- A. pie
- B. The code will throw an error
- C. pudding
- D. undefined

Answer: C

**6. Which of the following operators can be used to do a short-circuit evaluation?**

- A. ++
- B. --
- C. -+
- D. ||

Answer: D

**7. What does the following expression evaluate to?**

```
[] == []
```

- A. True
- B. undefined

C. False

D. []

Answer: C

**8. How many prototype objects are in the chain for the following array?**

```
let arr = [];
```

A. 3

B. 2

C. 0

D. 1

Answer: B

**9. An ..... is a JavaScript function that runs as soon as it is defined.**

A. Generator function

B. Arrow function

C. Regular Function

D. Immediately Invoked Function Expression

Answer: D

**10. What type of scope does the end variable have in the code shown?**

```
var start = 1;
if (start === 1) {
  let end = 2;
}
```

A. conditional

B. block

C. global

D. function

Answer: B

**11. Using ..... lets you share a set of CSS properties from one selector to another.**

A. @media

B. @inheritance

C. @extend

D. @include

Answer: C

**12. Which of the following directive displays the SassScript expression value as fatal error?**

- A. @error
- B. @warn
- C. @at-root
- D. None of the above

Answer: A

## Interview Questions

15m

### 1. What is Object Destructuring?

Answer: Object destructuring is a new way to extract elements from an object or an array.

Object destructuring:

Before ES6 version:

```
const classDetails = {  
  strength: 78,  
  benches: 39,  
  blackBoard:1  
}  
  
const classStrength = classDetails.strength;  
const classBenches = classDetails.benches;  
const classBlackBoard = classDetails.blackBoard;
```

The same example using object destructuring:

```
const classDetails = {  
  strength: 78,  
  benches: 39,  
  blackBoard:1  
}  
  
const {strength:classStrength, benches:classBenches,blackBoard:classBlackBoard} =  
classDetails;  
  
console.log(classStrength); // Outputs 78  
console.log(classBenches); // Outputs 39  
console.log(classBlackBoard); // Outputs 1
```

As one can see, using object destructuring we have extracted all the elements inside an object in one line of code.

If we want our new variable to have the same name as the property of an object we can remove the colon:

```
const {strength:strength} = classDetails;  
// The above line of code can be written as:  
const {strength} = classDetails;
```

## 2. What is the rest parameter?

Answer: It provides an improved way of handling parameters of a function.

Using the rest parameter syntax, we can create functions that can take a variable number of arguments. Any number of arguments will be converted into an array using the rest parameter. It also helps in extracting all or some parts of the arguments. Rest parameter can be used by applying three dots (...) before the parameters.

**Note- Rest parameter should always be used at the last parameter of a function.**

```
function extractingArgs(...args){  
  return args[1];  
}  
  
// extractingArgs(8,9,1); // Returns 9  
  
function addAllArgs(...args){  
  let sumOfArgs = 0;  
  let i = 0;  
  while(i < args.length){  
    sumOfArgs += args[i];  
    i++;  
  }  
  return sumOfArgs;  
}  
  
addAllArgs(6, 5, 7, 99); // Returns 117  
addAllArgs(1, 3, 4); // Returns 8
```

## 3. Explain what is the difference between Sass and SCSS?

Answer: The difference between Sass and SCSS is that,

Sass is a CSS pre-processor with syntax advancements and an extension of CSS3. Sass has two syntaxes. The first syntax is "SCSS" and it uses the .scss extension. The other syntax is indented syntax or just "Sass" and it uses the .sass extension. While Sass has loose syntax with white space and no semicolons, the SCSS resembles more to CSS. Any CSS valid document can be converted to Sass by simply changing the extension from .CSS to .SCSS.

#### 4. Can you tell us some limitations of JavaScript?

Answer: Everything has its own pros and cons. Likely, JavaScript has some limitations also. They are:

- Although JavaScript offers a client-side service, well, client-side JS doesn't approve users to write or read files. This policy has been established due to some security concerns.
- JS can sometimes be translated in different ways by several browsers. While the server-side JS will offer the same result all the time, client-side scripting language may vary slightly from time to time.
- JavaScript is not applicable for network applications because it doesn't offer any support for such applications.
- JavaScript doesn't come with any multiprocessor essentials.

#### 5. Explain how to define a variable in Sass?

Answer: Sass variables are simple: you assign a value to a name that begins with (\$), and then you can refer to that name instead of the value itself.

```
$base-color: #c6538c
$border-dark: rgba($base-color, 0.88)

.alert
  border: 1px solid $border-dark
```

#### 6. What is the spread operator?

Answer: Although the syntax of spread operator is exactly the same as the rest parameter, spread operator is used to spread an array, and object literals. We also use spread operators where one or more arguments are expected in a function call. Note- Key differences between rest parameter and spread operator: Rest parameter is used to take a variable number of arguments and turns into an array while the spread operator takes an array or an object and spreads it. Rest parameter is used in function declaration whereas the spread operator is used in function calls.

```
function addFourNumbers(num1,num2,num3,num4){
  return num1 + num2 + num3 + num4;
}

let fourNumbers = [5, 6, 7, 8];

addFourNumbers(...fourNumbers);
// Spreads [5,6,7,8] as 5,6,7,8

let array1 = [3, 4, 5, 6];
let clonedArray1 = [...array1];
// Spreads the array into 3,4,5,6
console.log(clonedArray1); // Outputs [3,4,5,6]
```

```
let obj1 = {x:'Hello', y:'Bye'};
let clonedObj1 = {...obj1}; // Spreads and clones obj1
console.log(obj1);

let obj2 = {z:'Yes', a:'No'};
let mergedObj = {...obj1, ...obj2}; // Spreads both the objects and merges it
console.log(mergedObj);
// Outputs {x:'Hello', y:'Bye',z:'Yes',a:'No'};
```

## 7. How to use a prompt box in JS?

Answer: The prompt box facilitates entering text in JavaScript – essentially, it's a dedicated field for typing in text symbols.

### Coding Challenge

20m

- [Coding Challenge: CC-03 CSS Grid](#)



### Coffee Break

10m



### Video of the Week

5m

- [What's new in Bootstrap 5?](#)



## Retro Meeting on a personal and team level

5m

Ask the questions below:

- What went well?
- What could be improved?
- What will we commit to do better in the next week?

## Case study/Project

15m

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

- [Checkout Page \(JS-05\)](#)
- [Sass Website Page Design \(HC-06\)](#)

## Closing

5m

-Next week's plan

-QA Session

---