

# TW-05 TEAM LEAD VERSION

---



CLARUSWAY  
WAY TO REINVENT YOURSELF

## Meeting Agenda

---

- ▶ Icebreaking
- ▶ Workshop Activities - Tuesday
- ▶ Teamwork Activities - Friday
  - ▶ Questions
  - ▶ Interview Questions
- ▶ Video of the week
- ▶ Retro meeting
- ▶ Case study / project

# Teamwork Schedule

---

## Ice-breaking

90m

- 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.

## Workshop Activities (Tuesday)

10m

- CSS Animation

## Team Work Activities (Friday)

### Ask Questions

20m

#### 1. What is an object in JavaScript?

- A. A function
- B. A data tool
- C. A data structure
- D. An array

Answer: C

#### 2. How do you access a property of an object in JavaScript?

- A. By using curly braces {}
- B. By using the dot notation
- C. By using parentheses ()
- D. By using commas

Answer: B

#### 3. How do you check if a property exists in an object in JavaScript?

- A. By using the exist keyword
- B. By using the contains keyword
- C. By using the hasOwnProperty method
- D. By using the isProperty method

Answer: C

```
const person = {  
  name: "Bob"  
};  
  
console.log(person.hasOwnProperty("name")); // true - Using hasOwnProperty  
console.log("age" in person); // false - Using the 'in' operator
```

#### 4. How do you delete a property from an object in JavaScript

- A. By using the `delete` keyword
- B. By using the `remove` keyword
- C. By setting the property value to null
- D. By assigning an empty string to the property

Answer: A

```
const person = {  
  name: "John",  
  age: 30  
};  
  
delete person.age; // Removing the 'age' property
```

#### 5. How do you add a new property to an existing object in JavaScript

- A. By using the `add` keyword
- B. By using the `insert` keyword
- C. By using the `update` keyword
- D. By assigning a value to a new key

Answer: D

#### 6. How can you create an empty object in JavaScript?

- A. `emptyObject = {};`
- B. `emptyObject = new Empty();`
- C. `emptyObject = Object.empty();`
- D. `emptyObject = new Object();`

Answer: A

## 7. How do you clone an object in JavaScript?

- A. Use the `Object.clone()` method
- B. Use the `Object.assign()` method or the spread operator (...)
- C. Use the `Object.copy()` method
- D. Use the `Object.duplicate()` method

Answer: B

By `Object.assign()` :

```
const originalObject = { name: "John", age: 30 };

// Clone the original object using Object.assign()
const clonedObject = Object.assign({}, originalObject);

// Now, 'clonedObject' is a separate copy of 'originalObject'
console.log(clonedObject); // { name: 'John', age: 30 }
```

By spread operator :

```
const originalObject = { name: "John", age: 30 };

// Clone the original object using the spread operator
const clonedObject = { ...originalObject };

// Now, 'clonedObject' is a separate copy of 'originalObject'
console.log(clonedObject); // { name: 'John', age: 30 }
```

## 8. What is object destructuring in JavaScript?

- A. A way to create objects from strings
- B. A way to concatenate objects
- C. A way to merge objects
- D. A way to extract properties from an object and assign them to variables

Answer: D

## 9. How do you swap the values of two variables without using a temporary variable using array destructuring?

- A. `const a = b; const b = a`
- B. `const [a, b] = [a, b];`
- C. `const [a, b] = [b, a];`
- D. `const [b, a] = [a, b];`

Answer: C

## 10. What happens if you try to destructure an array with more variables than there are elements in the array?

- A. Extra variables are assigned undefined
- B. An error is thrown
- C. The array is automatically resized
- D. Only the first few variables are assigned values

Answer: A

## 11. What does the rest element (...) do in array destructuring?

- A. It spreads elements into multiple arrays
- B. It gathers remaining elements into an array
- C. It removes elements from the array
- D. It reverses the order of elements in the array

Answer: B

```
// Example 1: Collecting remaining elements
const numbers = [1, 2, 3, 4, 5];

// Using rest element to collect the remaining elements
const [first, second, ...rest] = numbers;

console.log(first); // 1
console.log(second); // 2
console.log(rest); // [3, 4, 5]
```

## 12. What is JSON (JavaScript Object Notation)?

- A. A lightweight data interchange format
- B. A JavaScript method for creating objects
- C. A way to define variables in JavaScript
- D. A JavaScript library for animations

Answer: A : JSON is a lightweight data interchange format that is often used to transmit data between a server and a web application. It is based on a subset of JavaScript object literal notation.

```
{
  "name": "John Doe",
  "age": 30,
  "city": "New York",
  "isStudent": false,
  "hobbies": ["reading", "hiking", "cooking"],
  "address": {
    "street": "123 Main St",
```

```
    "zipcode": "10001"  
  }  
}
```

### 13. What is Bootstrap?

- A. A programming language
- B. A front-end framework
- C. A database management system
- D. An operating system

Answer: B

### 14. Which class indicates red colored border in bootstrap?

- A. border warning
- B. border border-danger
- C. border border-warning
- D. border-danger

Answer: B

### 15. Which class indicates lowercased text in bootstrap?

- A. lowercase
- B. text-lowercased
- C. text-lowercase
- D. txt-lowercased

Answer: C

### 16. Which class adds zebra-stripes to a table?

- A. .table-zebra
- B. .table-bordered
- C. .table-striped
- D. .even and .odd

Answer: C

**17. What is the Bootstrap class used to create a responsive navigation bar?**

- A. .navbar-fixed
- B. .nav-collapse
- C. .nav-responsive
- D. .navbar

Answer: D

**18. How can you include Bootstrap in your HTML file?**

- A. Download and link local Bootstrap files
- B. Use a CDN (Content Delivery Network) link
- C. No need to any action you can write BS Code directly in your project
- D. Python should be installed on your system

Answer: A & B

**19. What is the primary function of the Bootstrap modal component?**

- A. To create form elements
- B. To display a message box
- C. To define navigation links
- D. To handle user authentication

Answer: B

**20. The ..... class shapes the image to a circle.**

- A. .img-thumbnail
- B. .img-responsive
- C. .img-circle
- D. .img-rounded

Answer: C



**Coffee Break**

**10m**



## Interview Questions

20m

### 1. In Bootstrap, how do you make navigation elements?

*Answer :* The navigation elements in Bootstrap can be styled in a variety of ways. The markup and base class are the same in all of these .nav. To build tabular navigation or tabs, execute the following steps:

Begin by creating an unordered list using the base class of .nav. The .nav-tabs class should be added.

### 2. How can one create an alert in Bootstrap?

*Answer :* Create a wrapper `<div>` and add a class of .alert and one of the contextual classes to create a basic alert (e.g., .alert-success, .alert-info, .alert-warning, .alert-danger, .alert-primary, .alert-secondary, .alert-light or .alert-dark).

**Success!** This alert box indicates a successful or positive action.

**Info!** This alert box indicates a neutral informative change or action.

**Warning!** This alert box indicates a warning that might need attention.

**Danger!** This alert box indicates a dangerous or potentially negative action.

**Primary!** This alert box indicates an important action.

**Secondary!** This alert box indicates a less important action.

**Dark!** Dark grey alert box.

**Light!** Light grey alert box.

### 3. What is an object in JavaScript?

*Answer :* An object in JavaScript is a composite data type that allows you to store and organize data using key-value pairs. It is a collection of properties where each property is associated with a key (also called a property name) and a value.

### 4. Explain the difference between dot notation and bracket notation when accessing object properties.

*Answer :* Dot notation is used when the property name is known and is a valid JavaScript identifier. Bracket notation is used when the property name is dynamic, contains special characters, or is stored in a variable.



```
let obj = { key: "value" };

// Dot notation
let value = obj.key;

// Bracket notation
let dynamicKey = "key";
let anotherValue = obj[dynamicKey];
```

## 5. What is the difference between `Object.keys()`, `Object.values()`, and `Object.entries()`?

*Answer :* `Object.keys()` returns property names, `Object.values()` returns property values, and `Object.entries()` returns key-value pairs

### Video of the Week

15m

- [Object Destructuring in Javascript](#)

### Coding Challenge

15m

## 1. High Priced Product Categories

- You are given an array of objects representing a collection of products, each with a name, price, and category. Your task is to use map, filter, and reduce to calculate the average price of products in each category, and then return an array of objects containing only the categories that have an average price above 50.
- Sample input :

```
const products = [
  { name: "Product 1", price: 20, category: "Electronics" },
  { name: "Product 2", price: 30, category: "Clothes" },
  { name: "Product 3", price: 40, category: "Electronics" },
  { name: "Product 4", price: 50, category: "Clothes" },
  { name: "Product 5", price: 60, category: "Clothes" },
  { name: "Product 6", price: 70, category: "Electronics" },
  { name: "Product 7", price: 80, category: "Clothes" },
  { name: "Product 8", price: 90, category: "Electronics" },
];
```

- Expected outcome :

```
[
  { category: 'Clothes', average: 55 },
  { category: 'Electronics', average: 55 }
]
```

#### Solution :

```
const products = [
  { name: "Product 1", price: 20, category: "Electronics" },
  { name: "Product 2", price: 30, category: "Clothes" },
  { name: "Product 3", price: 40, category: "Electronics" },
  { name: "Product 4", price: 50, category: "Clothes" },
  { name: "Product 5", price: 60, category: "Clothes" },
  { name: "Product 6", price: 70, category: "Electronics" },
  { name: "Product 7", price: 80, category: "Clothes" },
  { name: "Product 8", price: 90, category: "Electronics" },
];

/* Use map to create an object with category as the key
and an array of products as the value */
const productsByCategory = products.reduce((acc, product) => {
  const category = product.category;
  if (!acc[category]) {
    acc[category] = [];
  }
  acc[category].push(product);
  return acc;
}, {});

// Use map to calculate the average price for each category
const avgPriceByCategory = Object.keys(productsByCategory).map(category => {
  const sum = productsByCategory[category].reduce((acc, product) => acc +
product.price, 0);
  return { category: category, average: sum / productsByCategory[category].length
};
});

// Use filter to only select categories with an average above a certain threshold
const highPricedCategories = avgPriceByCategory.filter(category => category.average
> 50);
console.log(highPricedCategories)
```

## 2. HR VS IT Department

- **Task :** You are given an array of objects representing a collection of employees, each with a name, salary, and department. Your task is to use map, filter, and reduce to calculate the average salary for each department and then return an array of objects containing only the departments that have an average salary above 65000.
- Sample input :

```
const employees = [  
  { name: "John", salary: 50000, department: "IT" },  
  { name: "Jane", salary: 60000, department: "HR" },  
  { name: "Bob", salary: 55000, department: "IT" },  
  { name: "Sophie", salary: 75000, department: "HR" },  
  { name: "Mike", salary: 65000, department: "IT" },  
  { name: "Emily", salary: 80000, department: "HR" },  
  { name: "David", salary: 70000, department: "IT" },  
];
```

- Expected outcome :

```
[  
  { department: 'HR', average: 71666 }  
]
```

### Solution :

```
const employees = [  
  { name: "John", salary: 50000, department: "IT" },  
  { name: "Jane", salary: 60000, department: "HR" },  
  { name: "Bob", salary: 55000, department: "IT" },  
  { name: "Sophie", salary: 75000, department: "HR" },  
  { name: "Mike", salary: 65000, department: "IT" },  
  { name: "Emily", salary: 80000, department: "HR" },  
  { name: "David", salary: 70000, department: "IT" },  
];  
  
/* Use reduce to create an object with department as the key  
and an array of employee objects as the value */  
const employeesByDepartment = employees.reduce((acc, employee) => {  
  const department = employee.department;  
  if (!acc[department]) {  
    acc[department] = [];  
  }  
  acc[department].push(employee);  
  return acc;  
}, {});
```

```
// Use map to calculate the average salary for each department
const avgSalaryByDepartment = Object.keys(employeesByDepartment).map(department =>
{
  const sum = employeesByDepartment[department].reduce((acc, employee) => acc +
employee.salary, 0);
  return { department: department, average: sum /
employeesByDepartment[department].length };
});

// Use filter to only select departments with an average above a certain threshold
const highPaidDepartments = avgSalaryByDepartment.filter(department =>
department.average > 65000);
console.log(highPaidDepartments)
```

## Retro Meeting on a personal and team level

10m

Ask the questions below:

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

## Closing

5m

- Next week's plan
  - QA Session
-