

TW-02 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

1- Get Season

- Create a program that determines the season based on the month and day entered by the user. (Do not use loops or functions) :
- Take month and day from the user.
- Season conditions:
 - Spring: February 21 - May 31
 - Summer: June 1 - September 21
 - Fall: September 22 - December 20
 - Winter: December 21 - February 20
- Inputs should 1-12 for month and 1-31 for day. If it is outside these values, an error message should be displayed on the console.

Expected Outputs :

If month is 9 and day is 20, output should be:

```
`The season for 9/20 is Summer.`
```

If month is 5 and day is 32, output should be:

```
'Invalid month or day. Please enter valid values.'
```

Solution :

```
const month = prompt('Enter a month (1-12)') // Enter a month (1-12)
const day = prompt('Enter a day (1-31)'); // Enter a day (1-31)
```

```
const isValidMonth = month >= 1 && month <= 12;
const isValidDay = day >= 1 && day <= 31;

if (!isValidMonth || !isValidDay) {
  console.log('Invalid month or day. Please enter valid values.');
```



```
} else {
  let season;

  if ((month >= 3 && month <= 5) || (month === 2 && day >= 20 && day <= 29)) {
    season = 'spring';
  } else if ((month >= 6 && month <= 8) || (month === 9 && day <= 21)) {
    season = 'summer';
  } else if ((month >= 9 && month <= 11) || (month === 12 && day <= 20)) {
    season = 'fall';
  } else {
    season = 'winter';
  }

  console.log(`The season for ${month}/${day} is ${season}.`);
}
```

2- Leap Year Checker :

- Write a JavaScript program that checks if a given year is a leap year. Leap years have a special property where they have an extra day, February 29th. The program uses conditional statements to determine whether the input year is a leap year or not. (Do not use loops or functions)
- Your program checks for leap years based on the following conditions:
 - If the year is divisible by 4 and not divisible by 100, it's a leap year.
 - If the year is divisible by 400, it's a leap year.

Expected Outputs :

- input : 2023 output: 2023 is not a leap year.
- input : 2024 output: 2024 is a leap year.

Solution :

```
const year = prompt('Enter year');

const isLeapYear =
  (year % 4 === 0 && year % 100 !== 0) ||
  (year % 400 === 0);

if (isLeapYear) {
  console.log(`${year} is a leap year.`);
} else {
  console.log(`${year} is not a leap year.`);
}
```

3 - Parallax Web Page

- [Project - 02 : Parallax Web Page \(HC-02\)](#)

Team Work Activities (Friday)

Ask Questions

20m

1. What is the main purpose of CSS Flexbox?

- A. To create two-dimensional layouts
- B. To style text and fonts
- C. To create responsive single-dimensional layouts
- D. To create animations

Answer: C

2. Which CSS property is used to turn an element into a flex container?

- A. display: block;
- B. display: flex;
- C. position: relative;
- D. float: left;

Answer: B

3. What is the default value for the flex-direction property?

- A. column
- B. column-reverse
- C. row-reverse
- D. row

Answer: D

4. Which CSS property is used to align items along the main axis in a flex container?

- A. justify-content
- B. align-items
- C. align-content
- D. flex-align

Answer: B

5. What is the purpose of the `fr` unit in CSS Grid?

- A.** It represents a fixed unit of measurement.
- B.** It defines a font size unit in CSS.
- C.** It stands for "fraction" and represents a fraction of available space in the grid container.
- D.** It is used for text formatting.

Answer: C

6. What is the key difference between CSS Flexbox and CSS Grid?

- A.** Flexbox is one-dimensional, while Grid is two-dimensional.
- B.** Flexbox is only for text styling, while Grid is for layout.
- C.** Flexbox can only be applied to block-level elements, while Grid can be applied to inline elements.
- D.** Flexbox and Grid have identical functionality.

Answer: A

7. How do you center an item both horizontally and vertically within a flex container?

- A.** `margin: auto;`
- B.** `text-align: center;` and `vertical-align: middle;`
- C.** `center: true;`
- D.** `justify-content: center;` and `align-items: center;`

Answer: D

8. What is the purpose of the `grid-area` property in CSS Grid?

- A.** It controls the visibility of grid items.
- B.** It sets the background color of grid items.
- C.** It defines the area within a grid item where content should be placed.
- D.** It specifies the gap between grid rows and columns.

Answer: C

9. What is the purpose of the `else` statement in JavaScript?

- A.** It defines a loop.
- B.** It provides an alternative code block to execute when the "if" condition is false.
- C.** It specifies the condition.
- D.** It terminates the program.

Answer: B

10. What is the result of the following JavaScript code snippet?

```
var x = 10;
var y = 5;
if (x > y) {
  console.log("x is greater than y");
} else if (x < y) {
  console.log("x is less than y");
} else {
  console.log("x is equal to y");
}
```

- A. "x is greater than y"
- B. "x is less than y"
- C. "x is equal to y"
- D. None of the above

Answer: A

11. In a JavaScript "if-else if-else" statement, which block of code is executed if none of the conditions are true?

- A. The "if" block.
- B. The "else if" block.
- C. The "if else" block.
- D. The "else" block.

Answer: D

12. Which loop is guaranteed to execute at least once in JavaScript?

- A. for loop
- B. do-while loop
- C. while loop
- D. for...in loop

Answer: B

13. Which loop is best suited for situations where you don't know how many times the loop needs to run in advance?

- A. while loop
- B. do-while loop
- C. for loop
- D. for each loop

Answer: A

14. In a "for" loop, what are the three statements enclosed in parentheses separated by semicolons?

- A. Condition, iteration, and execution
- B. Initialization, execution, and condition
- C. Initialization, condition, and iteration
- D. Condition, initialization, and execution

Answer: C

15. What is the purpose of the "continue" statement in a loop?

- A. It exits the loop prematurely.
- B. It restarts the loop from the beginning.
- C. It skips the current iteration and moves to the next one.
- D. It is not a valid statement in JavaScript.

Answer: C

16. What is the purpose of the "default" case in a JavaScript "switch" statement?

- A. It provides a code block to execute when none of the cases match.
- B. It defines the initial condition.
- C. It specifies the condition.
- D. It terminates the switch statement.

Answer: A

17. What happens if you forget to include an exit condition in a "while" loop?

- A. The loop will exit immediately.
- B. The loop will throw an error.
- C. The loop will only run once.
- D. The loop will run forever (infinite loop).

Answer: D

Interview Questions

20m

1. What are the escape characters in JavaScript?

*Answer: In JavaScript, we use escape characters, typically backslash (\) while working with special characters, such as ampersands (&), apostrophes ('), double quotes (" "), and single quotes (' '). Whatever enclosed within the escape characters gets displayed by the JavaScript.

Six additional escape characters are also available in JavaScript:

- \b – Backspace
- \f – Form feed
- \n – New line
- \r – Carriage return
- \t – Horizontal tabulator
- \v – Vertical tabulator

2. What is JavaScript 'Strict Mode'?

Answer: 'Strict mode' is a restricted variant of JavaScript. Usually, this language is 'not very strict' in throwing errors. But in 'Strict mode' it will throw all types of errors, even the silent errors. Thus, the process of debugging becomes easier. And the chances for making a mistake for the developer is reduced.

3. What is the difference between **var** and **let** keyword?

Answer: From the very beginning, the **var** keyword was used in JavaScript programming whereas the keyword **let** was just added in 2015.

The keyword **var** has a function scope. Anywhere in the function, the variable specified using var is accessible but in **let** the scope of a variable declared with the **let** keyword is limited to the block in which it is declared. Let's start with a Block Scope.

In ECMAScript 2015, let and const are hoisted but not initialized. Referencing the variable in the block before the variable declaration results in a ReferenceError because the variable is in a "temporal dead zone" from the start of the block until the declaration is processed

4. What is JavaScript Hoisting?

Answer: While using the **JavaScript Hoisting** method, when an interpreter runs the code, all the variables are hoisted to the top of the original /current scope. If you have a variable declared anywhere inside the code, then it is brought to the top.

This method is only applicable to the declaration of a variable and is not applicable for the initialization of a variable. Functions are also hoisted to the top, whereas function explanations are not hoisted to the top.

Basically, where we declared the variable inside the code doesn't matter much.



Coffee Break

10m



Video of the Week

15m

- [What You Can Do with JavaScript](#)

Case study/Project

15m

The case study will be solved by the students during the week and by the team on Friday Team Work.

- [Project - 03 : Home Furnishing Page \(HC-03\)](#)

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
-