TW-10 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 10m

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

Ask Questions 15m

1. What is the new avengers array after this code block?

```
let avengers = ["Iron Man", "Captain America", "Black Widow", "Hulk"];

avengers.splice(2, 1, "Thor", "Hawkeye");
```

A. ['Black Widow']
B. index error
C. ['Iron Man', 'Captain America', 'Thor', 'Hawkeye', 'Hulk']
D. ['Iron Man', 'Captain America', 'Thor', 'Hawkeye','Black Widow', 'Hulk']

Answer: C

2. Write a loop to iterate languages?

```
let fullStack = {
  languages: ["JavaScript", "React", "HTML"],
  jira: true,
  gitHub: true,
  difficulty: 8,
};

for (let i = 0; i < fullStack.languages.length; i++) {
  console.log(fullStack.languages[i]);
}

// alternatively
fullStack.languages.forEach((item) => console.log(item));

//output : JavaScript, React, HTML
```

3. Write a code for get `fullStack` object's keys

```
let fullStack = {
  languages: ["JavaScript", "React", "HTML"],
  jira: true,
  gitHub: true,
  difficulty: 8,
};

// old way, not advised anymore
for (let key in fullStack) {
  console.log(key);
}

// modern way
Object.keys(fullStack).forEach((item) => console.log(item));

//output : languages, jira, gitHub, difficulty
```

4. Write a method to get myCar's age

```
const myCar = {
  make: "ford",
  model: "Mustang",
  year: 1965,
  color: "Black",
};

myCar.age = function (current) {
  console.log(current - myCar.year);
};

myCar.age(2023); //Output: 58
```

5. What is the output of this code snippet?

```
(function (data) {
  return (function () {
    console.log(data);
    data = "Full Stack";
  })();
})("Clarusway");
```

A. `Full Stack`
B. `Clarusway`
C. `undefined`
D. `Error`

Answer: B

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

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

Answer:B

7. What will this code print?

```
var v = 1;
var f1 = function () {
  console.log(v);
};

var f2 = function () {
  var v = 2;
  f1();
};

f2();
```

A. 2
B. 1
C. Nothing - this code will throw an error.
D. undefined

Answer: B

8. Which of the following is most suitable for a grid layout?

A. toolbar
B. menu
C. form
D. complex layout, like a newspaper

Answer: D

9. Which model defines structures similar to tables using 2 dimensions?
A. CSS Box model
B. CSS3 Grid Layout model
C. CSS3 Flexbox Box model
D. CSS float

Answer: B

10. You are creating a grid layout. What does 1fr mean in the following code?

```
grid-template-columns: 150px 150px 1fr 1fr;
```

A. The first two columns will be two fraction units of the stated width.
B. The third and fourth columns is 1 fraction unit of the remaining space in the grid.
C. The second column will be double the stated width.
D. The second column will be half of the remaining space in the grid.

Answer: B

11. What layout modules require containers? (select all that apply)
A. CSS Box model
B. CSS3 Grid Layout model
C. CSS3 Flexbox Box model
D. CSS float

Answer: B and C

12. How many columns and rows are defined in the following code?

```
section {
  display: grid;

  grid-template-columns: 100px 1fr 1fr 250px;

  grid-rows: 50px 1fr 1fr;
}

section header {
  grid-column: 1 / 4;

  grid-row: 1;
}

section nav {
  grid-column: 1;

  grid-row: 2 / 3;
}

section article {
  grid-column: 2;

  grid-row: 2;
}
```

```
section aside {
  grid-column: 3;

  grid-row: 2;
}

section footer {
  grid-column: 1 / 4;

  grid-row: 3;
}
```

A. 1 rows and 2 columns
B. 2 rows and 3 columns
C. 3 rows and 2 columns
D. 3 rows and 4 columns

Answer: D

13. What kind of scoping does JavaScript use?

A. Literal
B. Lexical
C. Segmental
D. Sequential

Answer: B

Explanation: Like most modern programming languages, JavaScript uses lexical scoping. This means that functions are executed using the variable scope that was in effect when they were defined, not the variable scope that is in effect when they are invoked.

14. What is closure?

A. Function objects
B. Scope where function's variables are resolved
C. Both Function objects and Scope where function's variables are resolved
D. Function return value

Answer: C

Explanation: A combination of a function object and a scope (a set of variable bindings) in which the function's variables are resolved is called a closure.

15. What will be the output

```
var x = 10;

function counter() {
  let x = 1;
  return function () {
    return x++;
  };
}

scoreTracker = counter();
scoreTracker();
scoreTracker();
scoreTracker();
const latest = scoreTracker();

console.log(latest);
```

A. 10
B. 14
C. 4
D. 1

Answer: C

16. What will be printed in the console on execution of the following JS code:

```
var array = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15];
var myArr = array.filter((v) => v % 3 === 0);

console.log(myArr);
```

A. myArr
B. [3, 6, 9, 12, 15]
C. [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
D. [1, 2, 4, 5, 7, 8, 10, 11, 13, 14]

Answer: B

Explanation: This option is correct because, filter function will iterate through the array and will create an array of values if the condition is true. Hence all the elements divisible by three will be printed.

17. What is the output? Why?

```
console.log([1, 2, 3] == [1, 2, 3]);
```

Answer: false

Explanation: Arrays are a form of object. Objects are refence type data type. Whenever we define an object in call stack we don't store the value, we store the memory location in heap memory.

When we define [1, 2, 3] array, its content is stored in heap. Let's say at heap memory location EF10020. And in call stack we store this address not the content.

When we define the other [1, 2, 3] array, its content is aslo stored in heap. Let's say at heap memory location FF220202. And in call stack we store this address not the content.

So when we check whether they are equal we compare the addresses not the content. Therefore, they are not equal.

18. What is the output?

```
const arr1 = [1, 3, 5];
const arr2 = arr1;

console.log(arr1 === arr2);
```

A. true
B. false
C. error
D. time to time they are equal 😃

Answer: A

Explanation: Remember array object is a reference data type. When we write arr1 = arr2, we are assigning the address. So now arr1 and arr2 both store same address.

19. What is the output?

```
const arr1 = [1, 3, 5];
const arr2 = arr1;

arr1[0] = 100;

console.log(arr2);
```

A. [1, 3, 5]
B. [100, 3, 5]
C. error
D. []

Answer: B

Explanation: Remember array object is a reference data type. When we write arr1 = arr2, we are assigning the address. So now arr1 and arr2 both store same address and they both refer to the same memory location in heap. If we change any of them both of arrays will be affected from this assignment.

20. What is the output?

```
let arr1 = [1, 3, 5];
let arr2 = arr1;

arr1 = [2, 4, 6];

console.log(arr2);
```

A. [1, 3, 5]
B. [2, 4, 5]
C. error
D. []

Answer: A

Explanation: Remember array object is a reference data type. When we write arr1 = arr2, we are assigning the address. So now arr1 and arr2 both store same address and they both refer to the same memory location in heap. However, when we write arr1 = [2, 4, 6], we define a brand new array in heap. So arr1 and arr2 are not connected anymore.

Interview Questions 15m

1. How do callbacks work in JavaScript?

Answer: A callback is a function passed as an argument to another function, and a callback function can run after another function has finished. JavaScript functions execute in the sequence they get called, not in the defined sequence.

2. What Is Object Methods?

Methods are the functions that let the object do something or let something be done to it. There is a small difference between a function and a method – at a function is a standalone unit of statements and a method is

attached to an object and can be referenced by the this keyword.

Methods are useful for everything from displaying the contents of the object to the screen to performing complex mathematical operations on a group of local properties and parameters.

3. How do you check if an object is empty or not in Javascript?

You can check if an object is empty or not by using the Object.keys() method. This method will return an array of all the keys in the object, which you can then check to see if it is empty or not.

4. What is a CSS Grid?

A CSS Grid is a layout system for creating two-dimensional grid-based layouts. It is a powerful tool for creating responsive layouts, and can be used to create a wide variety of different types of layouts.

5. Why would you use CSS Grid instead of Flexbox?

CSS Grid is a newer layout system that offers a more powerful and flexible grid-based layout system than Flexbox. While Flexbox is better suited for one-dimensional layouts, CSS Grid is designed for two-dimensional layouts, making it a better choice for more complex web designs.

Coding Challenge 15m

- JS-CC-05 Array Iterations

☕

Coffee Break 10m

☕

Video of the Week 10m

CSS Grid vs CSS Flexbox: When to use CSS Grid vs Flexbox?

Case study/Project 15m

HC-07-Bootstrap Web Page

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
