TW-005 TEAM LEAD VERSION (Sprint-3 Week-2)

TEAMWORK

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 (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.

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. Which class indicates red colored border in bootsrap? 

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

Answer: B

2. Which class indicates lowercased text in bootsrap?

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

Answer: C

3. Which class adds zebra-stripes to a table?

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

Answer: C

4. 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

5. What is the output of this code block?

```
let daltones = ['joe', 'Jack', 'Willam', 'Averell']
daltones.shift()
daltones.pop()
console.log(daltones)
```

A. `['joe', Jack']`
B. `[]`
C. `['Jack', 'Willam']`
D. `['joe','Averell']`

Answer: C

6. What is the output of this code block?

```
let daltones = ['joe', 'Jack', 'Willam', 'Averell']

for(let i =0; i< daltones.length; i++){
    if(i == 1){
        continue;
    }
    console.log(daltones[i]);
}
```

A. Jack
B. joe, Jack
C. joe, Jack, Willam
D. joe, Willam, Averell

Answer: D

7. 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

8. Write a for loop for 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]);
}

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

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

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


for (let key in fullStack){
    console.log(key);
}


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

10. 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 - this.year)
    }

myCar.age(2022) //Output: 57
```

Interview Questions 15m

1. What Are The Number Methods in javaScript?

Answer :

The Number object contains only the default methods that are a part of every object's definition.

- toExponential(): Forces a number to display in exponential notation, even if the number is in the range in which JavaScript normally uses standard notation.
- toFixed(): Formats a number with a specific number of digits to the right of the decimal.
- toLocaleString(): Returns a string value version of the current number in a format that may vary according to a browser's local settings.
- toPrecision(): Defines how many total digits (including digits to the left and right of the decimal) to display of a number.
- toString(): Returns the string representation of the number's value.
- valueOf(): Returns the number's value.

2. What Is Javascript Date Object?

Answer :

The Date object is a datatype built into the JavaScript language. Date objects are created with the new Date( ) as shown below.

Once a Date object is created, a number of methods allow you to operate on it. Most methods simply allow you to get and set the year, month, day, hour, minute, second, and millisecond fields of the object, using either local time or UTC (universal, or GMT) time.

The ECMAScript standard requires the Date object to be able to represent any date and time, to millisecond precision, within 100 million days before or after 1/1/1970. This is a range of plus or minus 273,785 years, so JavaScript can represent date and time till the year 275755.

Syntax: You can use any of the following syntaxes to create a Date object using Date() constructor.

new Date( ) new Date(milliseconds) new Date(datestring) new Date(year,month,date[,hour,minute,second,millisecond])

3. What are forms and how to create forms in HTML?

Answer:

Here is a list of the methods used with Date and their description.

Date(): Returns today's date and time getDate(): Returns the day of the month for the specified date according to local time. getDay(): Returns the day of the week for the specified date according to local time. getFullYear(): Returns the year of the specified date according to local time. getHours(): Returns the hour in the specified date according to local time. getMilliseconds(): Returns the milliseconds in the specified date according to local time. getMinutes(): Returns the minutes in the specified date according to local time. getMonth(): Returns the month in the specified date according to local time. getSeconds():Returns the seconds in the specified date according to local time. getTime(): Returns the numeric value of the specified date as the number of milliseconds since January 1, 1970, 00:00:00 UTC. getTimezoneOffset(): Returns the time-zone offset in minutes for the current locale. getUTCDate(): Returns the day (date) of the month in the specified date according to universal time. getUTCDay(): Returns the day of the week in the specified date according to universal time. getUTCFullYear(): Returns the year in the specified date according to universal time. getUTCHours(): Returns the hours in the specified date according to universal time. getUTCMilliseconds(): Returns the milliseconds in the specified date according to universal time. getUTCMinutes(): Returns the minutes in the specified date according to universal time. getUTCMonth(): Returns the month in the specified date according to universal time. getUTCSeconds(): Returns the seconds in the specified date according to universal time. getYear(): Deprecated - Returns the year in the specified date according to local time. Use getFullYear instead. setDate(): Sets the day of the month for a specified date according to local time. setFullYear(): Sets the full year for a specified date according to local time. setHours(): Sets the hours for a specified date according to local time. setMilliseconds(): Sets the milliseconds for a specified date according to local time. setMinutes(): Sets the minutes for a specified date according to local time. setMonth(): Sets the month for a specified date according to local time. setSeconds(): Sets the seconds for a specified date according to local time. setTime(): Sets the Date object to the time represented by a number of milliseconds since January 1, 1970, 00:00:00 UTC. setUTCDate(): Sets the day of the month for a specified date according to universal time. setUTCFullYear(): Sets the full year for a specified date according to universal time. setUTCHours(): Sets the hour for a specified date according to universal time. setUTCMilliseconds(): Sets the milliseconds for a specified date according to universal time. setUTCMinutes(): Sets the minutes for a specified date according to universal time. setUTCMonth(): Sets the month for a specified date according to universal time. setUTCSeconds(): Sets the seconds for a specified date according to universal time. setYear(): Deprecated - Sets the year for a specified date according to local time. Use setFullYear instead. toDateString(): Returns the "date" portion of the Date as a human-readable string. toGMTString(): Deprecated - Converts a date to a string, using the Internet GMT conventions. Use toUTCString instead. toLocaleDateString(): Returns the "date" portion of the Date as a string, using the current locale's conventions. toLocaleFormat(): Converts a date to a string, using a format string. toLocaleString(): Converts a date to a string, using the current locale's conventions. toLocaleTimeString(): Returns the "time" portion of the Date as a string, using the current locale's conventions.

toSource(): Returns a string representing the source for an equivalent Date object; you can use this value to create a new object. toString(): Returns a string representing the specified Date object. toTimeString(): Returns the "time" portion of the Date as a human-readable string. toUTCString(): Converts a date to a string, using the universal time convention. valueOf(): Returns the primitive value of a Date object.

4. 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.*

5. 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).

-

Coding Challenge 20m

- Coding Challenge: Roman Numerals

☕

Coffee Break 10m

☕

Video of the Week 5m

- Object Destructuring in Javascript

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

- Bootstrap Web Page

Closing 5m

-Next week's plan

-QA Session