Daily Task:

(Interview question)

1. Break:

The break statement "jumps out" of a loop

Continue:

The continue statement "jumps over" one iteration in the loop.

2. Function types:

A set of statements that performs a task or calculates a value.

Types:

→ Named functions
→ Anonymous functions
↔ Immediately invoked function

(i) Named function:

It's useful if we need to call a function many times to pass different values to it or run it several times.

ex:

```
function oddOrEven(number) {
    if (number % 2 == 0) {
        return "number is even"
    } else {
        return "number is odd"
    }
```

(ii) Anonymous function:

The anonymous functions don't have names. They need to be tied to something: variable or an event to run.

```
let oddorEven = function (N) {
    if (N % 2 == 0) {
        return "Number is even"
    } else {
        return "Number is odd"
    }
}
```

(iii) Immediately invoked functions:

It expressions as soon as the brower encounters it. The basic of this functions is that it runs immediately where its located in the code & produces a direct output.

```
let msg = (function () {
    let name = "Hai";
    return name;
})();
```

3, Replace:

~~If search argument is a string,~~

The string method String.replace (regExp Search, replaceWith) Search & replace the occurrences of the regular expression regExp Search with replace string.

```
let a = "Hello world";
let b = a.replace("world", "everyone")
console.log(b);
```

replaceAll();

The method String.replaceAll (search, replaceWith) replaceall appearances of search string with replacewith.

4. Ternary operator: (conditional)

It's the only JS operator that takes three operands: a condition followed by a question mark (?), then an expression to execute if the condition is truthy followed by colon (:), & finally the expression to execute if the condition is falsy. This operator is frequently used as an alternative if ... else statement.

syntax:

condition ? exprIfTrue : exprIfFalse

5. To link css files:

Three ways to insert CSS

↳ External CSS
↳ Internal CSS
↳ Inline CSS

External CSS:

With an external style sheet, you can change the look of an entire website by changing just one file.

<link rel="stylesheet" href="style.css">

Internal CSS:

An internal style sheet may be used if one single HTML page has a unique style.

Inline CSS:

An inline style may be used to apply a unique style for a single element

6. splice:

It will be changes the original array.

There are '3' value pass the array.

(Target, deletecount, add)

var A = [1,2,3,4,5]

var B = A.splice(2)

console.log(B)

O/P → [3,4,5] → array size A = [1,2]

Slice:

Not affected in original array.

(start, end)

var A = [1,2,3,4,5]

var B = A.slice(1,4)

console.log(B)

O/P → [2,3,4] → array size A = [1,2,3,4,5]

7. Local Storage:

This read only interface property provides access to the document's local storage object, the stored data is stored across browser

session storage methods:

→ setItem() method:

localstorage.setItem(key, value)

→ getItem() method:

localstorage.getItem(key)

→ removeItem:

localstorage.removeItem(key)

↳ clear() method:

localstorage.clear()

Session storage:

data in sessionstorage is cleared when the page session ends.

Methods:

↳ save data to session storage

sessionStorage.setItem("key", "value")

↳ Read data from session storage

let lastname = sessionstorage.getItem("

↳ Remove data from sessionstorage

sessionstorage.removeItem("key")

↳ RemoveAll

sessionstorage.clear()

8. Spread operator:

(...) Allows an iterable as an array expression or string to be expanded in places where zero or more arguments are expected, or an object expression to be expanded in places where zero or more key-value pairs are expected.

ex: function sum(x, y, z) {

return x+y+z

}

const numbers = [1, 2, 3]

```
const A = [1, 2, 3];
const B = [4, 5, 6];
const numberOfA&B = [...A, ...B];
    console.log (numberofA&B)
```

O/P ⟹ 1, 2, 3, 4, 5, 6

visibility:

The visibility property specifies whether or not an element

visibility: visible → The particular content or section will be view.

visibility: hidden → The content or section will be hidden.

display:

The display property specifies the display behavior (the type of rendering box) of an element.

display: none → The element is completely removed.

display: inline → displays an element as an inline element (like . Any height & width properties will have no effect.

display: inline-block → displays an element as an inline-level block container. The element itself is formatted as an inline element, but you can apply height & width values.

display: block → displays an element as a block element (like <p>). It starts on a new line, & takes up the whole width.

10) Invoke function:

The code inside a function is executed when the function is invoked.

The code inside a function is not executed when the function is desired.

or:

```
function myfunction (a,b) {
    return a*b;
}
console.log(myfunction (10,12))
```

O/P → 120

string methods:

→ at()

To display the index position of the string

ex:

```
let mystring = 'Table of contents';
let b = mystring.at(4);
console.log(b);
```

o/p → 'e'

```
let mystring = 'Table of contents';
let mystring2 = 'will be change';
```

→ concat()

To merge the two string variable

ex:

```
let b = mystring.concat(mystring2)
console.log(b)
```

(" ", mystring2)

o/p:

Table of contentswill be change

Table of contents will be change

→ endswith()

string endwith the characters of a specified string. returning true or false.

```
let t = mystring.endswith('contents', mystring.length)
```

o/p → true.

```
let = mystring.endwith('contents')
```

o/p → true

↳ includes() To check the charactor in the string To return bool value

let b = mystring.Includes('of') ↳ includes also position

o/p → true

↳ indexof()

To display the length of the string Position

let b= "thai ovoide - thai selvam"

let c = b.indexof('ide')

o/p → 11

↳ lastindexof()

To display the last length of the same index value of the string position.

let b= b.lastindex of ("Hai")

o/p → 12

↳ charCodeAt()

To display unique code character number will be displayed.

let b= mystring.charCodeAt("T")

o/p → 32

↳ fromcharcode()

To display the charactor of the string

let b= String.fromCharCode("114");

console.log(b);

o/p → "r"

↳ match ( )

To display orital & duplicate string element will be in array.

let b = "Hai everyone - Hai selvam"

let c = b.match (/Hai/g)

O/P ⇒ ['Hai', 'Hai']

↳ repeat ()

To repeat the string variable.

let a = "Hello world";

let b = a.repeat (3)

O/P → Hello worldHello worldHello world

↳ Replace ():

let a = "Hello world"

let b = a.replace (/everyone/world/g, "Everyone")

O/P → Hello Everyone.

↳ search ()

To display index position of the element

let a = "Hello world"

let b = a.search ("world")

O/P → 5

↳ slice ()

→ Index position

start, end & (n-1)

To cut & display the needed element

let a = "Hello world"

let b = a.slice (3, 7)

O/P → lo w

↳ split()

To split the value in qvation to Kyphen, underscore.

let a = "Hollo world"

let b = a.split(" ") → ("-") ("/")

o/p: "Hollowo" "world"

↳ startwith()

To chalk the string startwith Particular Character "true or false" → Return

let a = "Hollo world"

let b = a.startwith("Hollo")

e/p → true

↳ substring:

It's similar to slice. To cut the needed value

let a = "Hollo world" (start, end)

let b = a.substring (2, 4) (n-1)

o/p → ll

↳ substr()

To start the index position, and the extend value in number.

let a = "Hollo world"

let b = a.substr(2, 5) (start, start after the Index value)

o/p → llo wo

↳ tolowercase()

All the characters in small letter

let a = "Hollo world"

let b = a.tolowercase()

o/p → hollo world

↳ toUpperCase()

All the charactors in capital letter

let A = "Hello world"

let B = A.toUpperCase(A)

O/P → HOLLO WORLD

↳ trim

To remove the starting & ending unwanted space

let a = " Hollo world "

let b = a.trim()

O/P → Hollo world