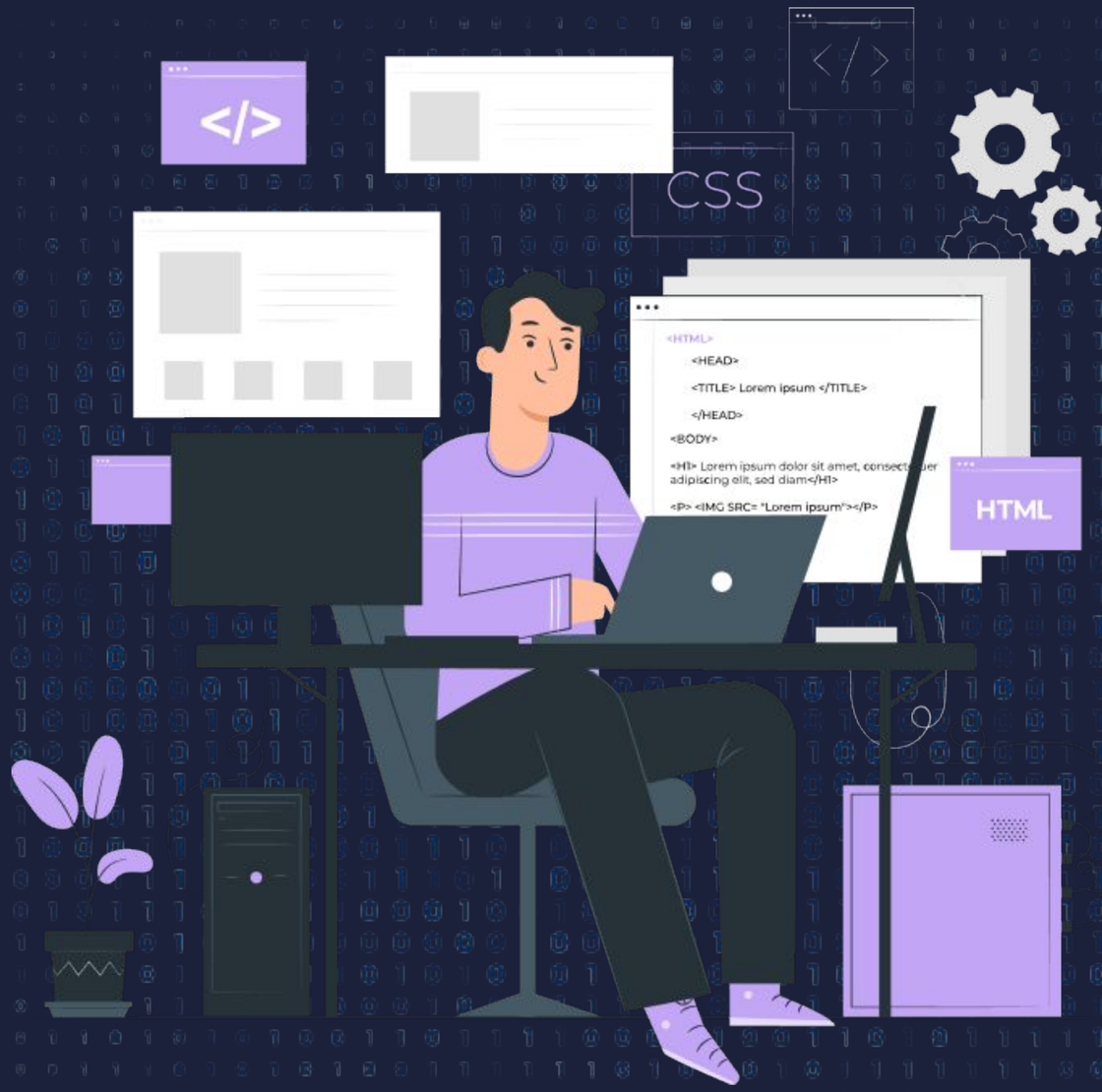
PW SKILLS

Array methods



Lecture CheckList

- What are Arrays Methods
- Explain different types of Array methods with examples

What are Arrays Methods

Array methods are built-in functions in JavaScript that allow you to perform various operations on arrays. They can be used to manipulate arrays in various ways, including adding and removing elements, sorting, transforming, and filtering the elements of an array.

Explain different types of Array methods with examples

Some of the different types of basic Array methods include the following -

1. pop(),
2. push(),
3. hift(),
4. unshift(),
5. concat(),
6. join(),
7. slice(),
8. splice(),
9. reverse(),
10. indexOf(),
11. toString(),
12. flat(),
13. isArray().

pop()

```
let array1 = [1, 2, 3];
let array2 = [4, 5, 6];
let newArray = array1.concat(array2);
console.log(newArray);
// Output: [1, 2, 3, 4, 5, 6]

let array1 = [1, 2, 3];
let array2 = [4, 5, 6];
let array3 = [7, 8, 9];
let newArray = array1.concat(array2, array3);
console.log(newArray);
// Output: [1, 2, 3, 4, 5, 6, 7, 8, 9]
```

unshift()

```
let fruits = ['banana', 'cherry'];
let length = fruits.unshift('apple');
console.log(fruits); // Output: ['apple', 'banana', 'cherry']
console.log(length); // Output: 3

let fruits = ['banana', 'cherry'];
let length = fruits.unshift('apple', 'orange');
console.log(fruits); // Output: ['apple', 'orange',
'banana', 'cherry']
console.log(length); // Output: 4
```

slice()

```
let fruits = ['apple', 'banana', 'cherry'];
let first = fruits.shift();
console.log(fruits); // Output: ['banana', 'cherry']
console.log(first); // Output: 'apple'
```

push()

```
let fruits = ['apple', 'banana', 'cherry'];
let last = fruits.pop();
console.log(fruits); // Output: [ 'apple', 'banana' ]
console.log(last); // Output: 'cherry'

let fruit = ['apple', 'banana'];
let length = fruit.push('cherry', 'orange');
console.log(fruit); // Output: ['apple', 'banana', 'cherry',
'orange']
console.log(length); // Output: 4
```

Concat():

```
let fruits = ['apple', 'banana'];
let length = fruits.push('cherry');
console.log(fruits); // Output: ['apple', 'banana', 'cherry']
console.log(length); // Output: 3
```

slice()

```
let fruits = ['apple', 'banana', 'cherry', 'orange'];
let citrus = fruits.slice(1, 3);
console.log(fruits); // Output: ['apple', 'banana', 'cherry',
'orange']
console.log(citrus); // Output: ['banana', 'cherry']

let numbers = [1, 2, 3, 4, 5];
let part = numbers.slice(2);
console.log(numbers); // Output: [1, 2, 3, 4, 5]
console.log(part); // Output: [3, 4, 5]
```

join()

```
let fruits = ['apple', 'banana', 'cherry'];
let result = fruits.join();
console.log(result); // 'apple,banana,cherry'

let result = fruits.join('-');
console.log(result); // 'apple-banana-cherry'
```

splice()

```
let numbers = [1, 2, 3, 4, 5];
let removed = numbers.splice(2, 2, 6, 7);
console.log(numbers); // Output: [1, 2, 6, 7, 5]
console.log(removed); // Output: [3, 4]
```

reverse()

```
let fruits = ['apple', 'banana', 'kiwi', 'mango'];
fruits.reverse();
console.log(fruits);
// Output: ["mango", "kiwi", "banana", "apple"]
```

toString()

```
const numbers = [1, 2, 3, 4, 5, 6, 7, 8];

const numberOfString = numbers.toString();
console.log("Type of - ", typeof numberOfString);
console.log("Number of String", numberOfString);
//---- output ----
// Type of - string
// Number of String "1, 2, 3, 4, 5, 6, 7, 8"
```

indexOf()

```
let fruits = ['apple', 'banana', 'kiwi', 'mango'];
let index = fruits.indexOf('kiwi');
console.log(index);
// Output: 2

let fruits = ['apple', 'banana', 'kiwi', 'mango', 'kiwi'];
let index = fruits.indexOf('kiwi', 3);
console.log(index);
// Output: 4
```

flat()

```
const nestArray = [1, 2, 3, [4, 5]];
const flatArray = nestArray.flat(); // default
const flatArrayOne = nestArray.flat(1); // default
console.log(flatArray); // [1,2 3, 4, 5]
console.log(flatArrayOne) // [1,2 3, 4, 5]
const nestDepth = [0, 1, 2, [3, 4, [5, 6, [7, 8]]]];
const flatOneDefault = nestDepth.flat(1);
console.log(flatOneDefault); // [ 0, 1, 2, 3, 4, [ 5, 6, [ 7, 8 ]
] ]
const flatTwo = nestDepth.flat(2);
console.log(flatTwo); // [ 0, 1, 2, 3, 4, 5, 6, [ 7, 8 ] ]
const flatThree = nestDepth.flat(3);
console.log(flatThree); // [  0, 1, 2, 3, 4,5, 6, 7, 8 ]
const infinity = nestDepth.flat(Infinity);
console.log(infinity); // [  0, 1, 2, 3, 4,5, 6, 7, 8 ]
```

isArray()

```
const lanArr = ["javascript", "python", "java", "golang"];
console.log(Array.isArray(lanArr)); // true

const cityName = "Bangalore";
console.log(Array.isArray(cityName)); // false
```


SKILLS
THANK YOU