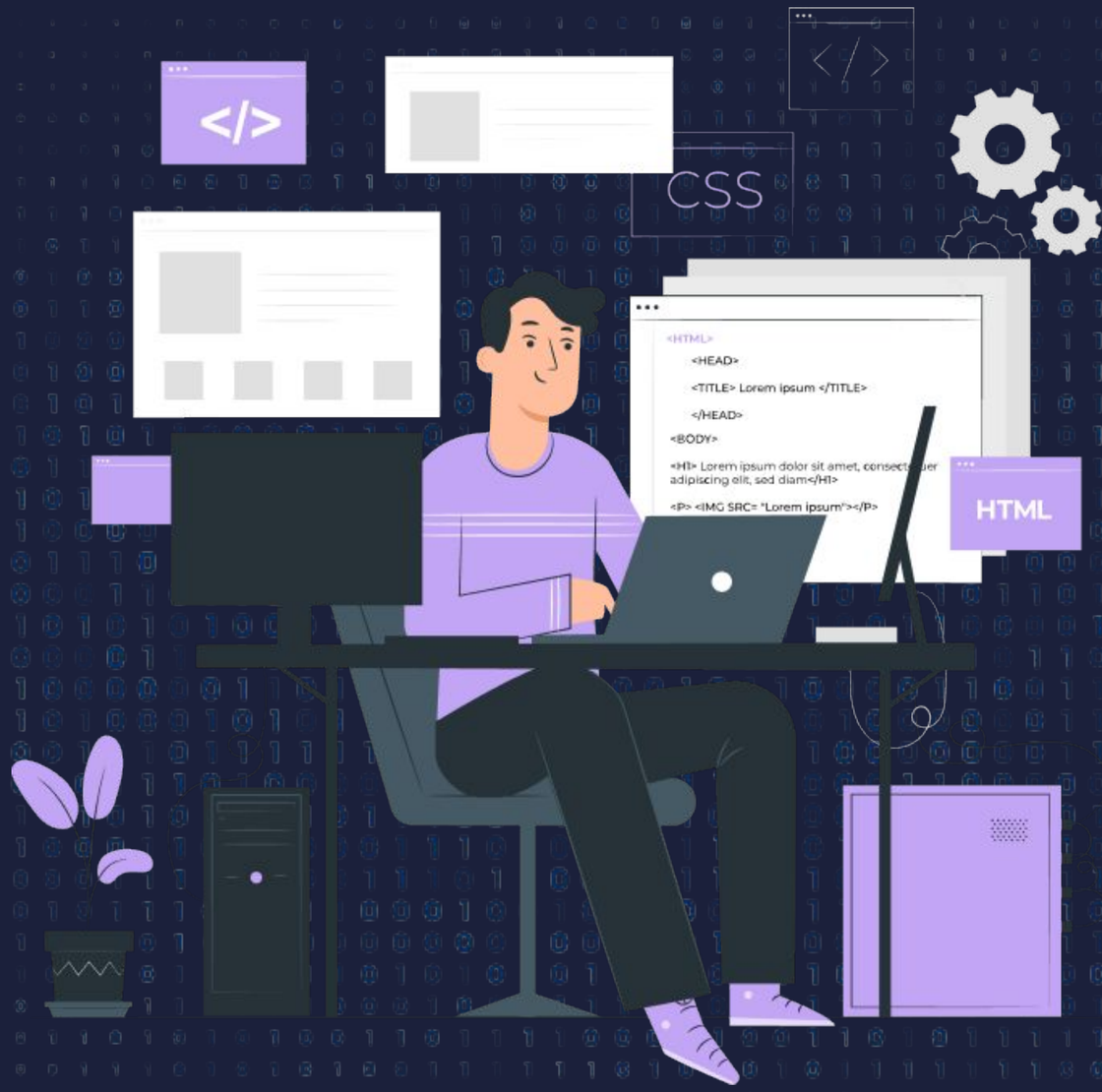
PW SKILLS

Spread and Rest Operators



Topics:

- Introduction to Spread and Rest Operators
- Spread operator with example
- Rest operator with example

Introduction to Spread and Rest Operators

The Rest and Spread operators in JavaScript are two powerful features that allow you to work with arrays and objects in a more flexible and concise way. The Spread operator allows you to spread elements of an array or iterable object into separate arguments, while the Rest operator gathers elements into an array. These operators provide a convenient way to manipulate and manage arrays and objects and are widely used in many programming scenarios. Let's look into these operators in this lecture.

Spread operator with example

The Spread operator, in general, allows iterable (arrays /objects /strings) to be expanded into single arguments/elements.

Some of the applications where spread operators are used -

1. Creating a new Array
2. Adding new values in Array
3. Concatenating two Array
4. Spread an array of arguments to be passed as individual params
5. Using with Strings
6. Spread Operator with Objects

Creating a copy of an existing Array -

```
// 1.Creating a New Array
let techStack = [
  "HTML",
  "CSS",
  "JS",
  "React",
  "Node",
  "Express",
  "MongoDB",
  "Git",
];
let newArrayCreated = [...techStack];
console.log(newArrayCreated);
// OUTPUT: ["HTML", "CSS", "JS", "React", "Node",
"Express", "MongoDB", "Git"]
```

Adding new values in Array and Object

```
// 2.Adding New Values in array and object
let arr = ["HTML", "CSS", "JS", "React", "Node"];
let newArr = [...arr, "Git"];
console.log(newArr);
// OUTPUT: [ 'HTML', 'CSS', 'JS', 'React', 'Node',
'Git' ]

let obj = { name: "PW Skills", course: "Full Stack Web
Developer" };
let newObj = { ...obj, ratings: 5 };
console.log(newObj);

// OUTPUT: { name: 'PW Skills', course: 'Full Stack
Web Developer', ratings: 5 }
```

Concatenating two Array

```
// 3.Concatenating two arrays
let arr1 = ["HTML", "CSS", "JS"];
let arr2 = ["React", "Node", "Express"];
let concatenatedArray = [...arr1, ...arr2];
console.log(concatenatedArray);
// OUTPUT: [ 'HTML', 'CSS', 'JS', 'React', 'Node',
'Express' ]
```

Using with Strings

```
// 5. Using with Strings
let name = "PW Skills";
let arrayofCharacters = [...name];
console.log(arrayofCharacters);
// OUTPUT : ['P', 'W', ' ', 'S', 'k', 'i', 'l', 'l',
's']
```

Spread Operator with Objects

```
// 6. Spread Operator with Objects
let obj1 = { name: "PW Skills", course: "Full stack
web development" };
let obj2 = { rating: 5, reviews: 2000 };
let newObjCreated = { ...obj1, ...obj2 };
console.log(newObjCreated);
/*
OUTPUT:{
    name: 'PW Skills',
    course: "Full stack web development",
    rating: 5,
    reviews: 2000 }*/
```

```
let obj1 = {
  name: "PW Skills",
  course: "Full stack web development",
  numberOfStudentsEnrolled: 1000,
};
let obj2 = { rating: 5, reviews: 2000,
numberOfStudentsEnrolled: 2000 };
let newObjCreated = { ...obj1, ...obj2 };
console.log(newObjCreated);
/*{name: 'PW Skills',
      course: 'Full stack web development,
      numberOfStudentsEnrolled: 2000,
      rating: 5,
      reviews: 2000}*/
```

Rest operator with example

The rest operator is used to collect all elements into an array. The representation is the same as a Spread operator but used differently. The rest operator gathers elements into an array

Some of the applications where Rest operators are used include -

1. Collecting all remaining parameters in a function.
2. Destructuring

Collecting all remaining parameters in a function.

```
// 1. Rest  - Collecting all remaining parameters in a
function.
function sumOfAllNumbers(...numbers) {
  return numbers.reduce((acc, curr) => acc + curr);
}
console.log(sumOfAllNumbers(1, 2, 3, 4));
// OUTPUT: 10
```

Destructuring

```
//2 Rest - Destructuring
// destructuring an array
let arr = ["HTML", "CSS", "JS", "React", "Node",
"Express", "Git"];
let [element1, element2, ...remainingElements] = arr;
console.log(element1); // OUTPUT: HTML
console.log(element2); // OUTPUT: CSS
console.log(remainingElements);
// OUTPUT: [ 'JS', 'React', 'Node', 'Express', 'Git' ]
// destructuring object
let obj = {
  name: "PW Skills",
  course: "Full Stack Web Development",
  rating: 5,};
let { name, ...remainingProperties } = obj;
console.log(name); // OUTPUT: PW Skills
console.log(remainingProperties);
// OUTPUT: { course: 'Full Stack Web Development',
rating: 5 }
```


SKILLS
THANK YOU