

Javascript ES6 Features



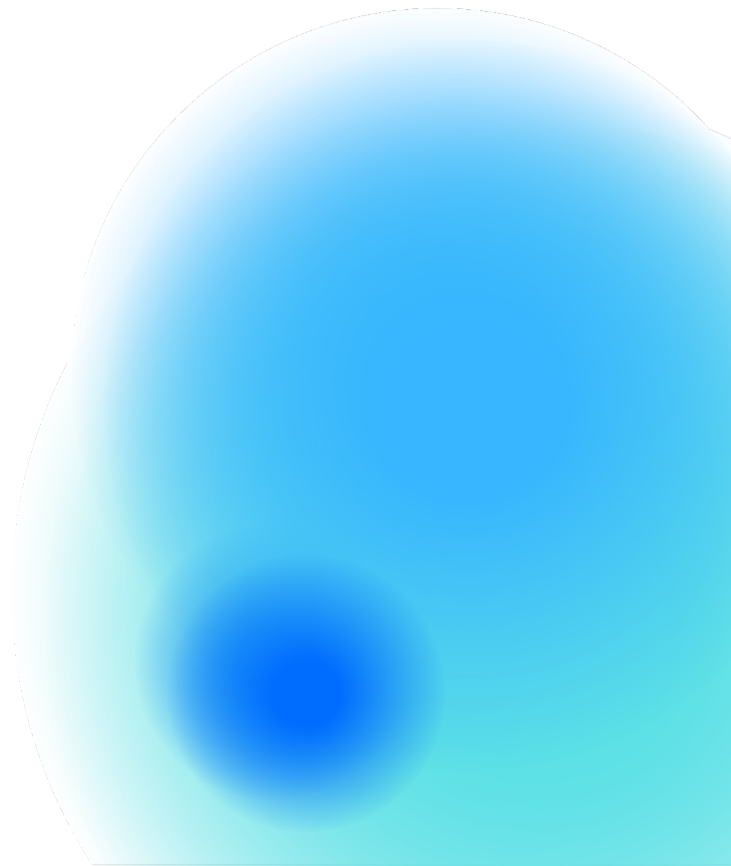
Javascript ES6 Features

Arrow Function

```
const createUser = (user)=>{  
  //logic to create user  
  console.log('User Created');  
}  
createUser(userData);
```

IIFE (Immediately Invoked Function Expression)

```
(function(){  
  console.log('This function ran as soon as it was created');  
})();
```



Javascript ES6 Features

Destructuring

Destructuring is a feature in JavaScript that makes it easy to extract values from arrays or properties from objects and assign them to variables.

```
const numbers = [1, 2, 3];
const [a, b, c] = numbers;
console.log(a); // 1
console.log(b); // 2
```

Rest Operator

```
const numbers = [1, 2, 3];
const [a, ...rest] = numbers;
console.log(a); // 1
console.log(b); // [2, 3]
```

Javascript ES6 Features

Object Destructuring

Extract values from an object using matching property names.

```
const person = { name: "Alice", age: 25, city: "Delhi"};
const { name, age } = person;
console.log(name); // Alice
console.log(age); // 25
```

Rename Key

```
const { name: fullName } = person;
console.log(fullName); // Alice
```

Destructuring in Function Parameters

```
function greet({ name, city }) {
  console.log(`Hello ${name} from ${city}`);
}
greet({ name: "Bob", city: "Mumbai" }); // Hello Bob from Mumbai
```

Javascript ES6 Features

Spread Operator (...)

The spread operator (...) allows us to expand elements of an array, object, or iterable into individual elements.

Copying an array

```
const nums = [1, 2, 3];
const copy = [...nums];
```

Merging arrays

```
const a = [1, 2];
const b = [3, 4];
const merged = [...a, ...b];
```

Adding Elements

```
const numbers = [2, 3];
const result = [1, ...numbers, 4];
```

Javascript ES6 Features

Spread in Objects (ES2018+)

The spread operator (...) allows us to expand elements of an array, object, or iterable into individual elements.

Copying objects

```
const user = { name: "Alice", age: 25 };
const clone = { ...user };
```

Merging objects

```
const obj1 = { a: 1 };
const obj2 = { b: 2 };
const merged = { ...obj1, ...obj2 };
```

Spread with Function Calls

```
function sum(x, y, z) { return x + y + z;}
const values = [1, 2, 3];
console.log(sum(...values));
```


Javascript ES6 Features

Hoisting

Hoisting is a behavior in JavaScript where variable and function declarations are moved to the top of their scope (script or function) before code execution. But only declarations are hoisted, not initializations.

Variable Hoisting : variables declared with var are hoisted with value undefined.

```
console.log(a); // undefined
var a = 10;
```

let and const are also hoisted but in Temporal Dead Zone (TDZ) so if one access them before declaration will throw error

Function Hoisting: Function Declarations are fully hoisted

```
sayHello(); // Hello!
function sayHello() {
  console.log("Hello!");
}
```

THANK YOU

PHONE NUMBER

(+91) 778 899 2897

WEBSITE

www.indixpert.com

