Essential JavaScript Array Methods and ES6 Features

Array Methods

1. Map

The `map` method transforms each element in an array and returns a new array with the results.

```
const numbers = [1, 2, 3, 4, 5];
const squaredNumbers = numbers.map((num) => num ** 2);
// Result: [1, 4, 9, 16, 25]
```

2. Filter

The `filter` method creates a new array with elements that satisfy a specified condition.

```
const numbers = [1, 2, 3, 4, 5];
const oddNumbers = numbers.filter((num) => num % 2 !== 0);
// Result: [1, 3, 5]
```

3. Every

The `every` method checks if all elements in an array pass a specified test.

```
const positiveNumbers = [1, 2, 3, 4, 5];
const allPositive = positiveNumbers.every((num) => num > 0);
// Result: true
```

4. Reduce

The `reduce` method aggregates array values into a single result.

```
const numbers = [1, 2, 3, 4, 5];
const sum = numbers.reduce((acc, curr) => acc + curr, 0);
// Result: 15
```

ES6 Features

1. Default Parameters

ES6 allows setting default values for function parameters.

```
function greet(name = 'Guest') {
  console.log(`Hello, ${name}!`);
}
// greet(); // Hello, Guest!
// greet('John'); // Hello, John!
```

2. Spread Operator

2.1 Spread (Array Literals)

Expands array elements for easy concatenation or cloning.

```
const arr1 = [1, 2, 3];
const arr2 = [...arr1, 4, 5, 6];
// Result: [1, 2, 3, 4, 5, 6]
```

2.2 Spread (Object Literals)

Merges object properties efficiently.

```
const obj1 = { x: 1, y: 2 };
const obj2 = { ...obj1, z: 3 };
// Result: { x: 1, y: 2, z: 3 }
```

3. Rest Parameter

Allows a function to accept an indefinite number of arguments as an array.

```
function sum(...numbers) {
  return numbers.reduce((acc, num) => acc + num, 0);
}
// sum(1, 2, 3, 4, 5); // Result: 15
```

4. Destructuring

4.1 Destructuring (Arrays)

Simplifies extracting values from arrays.

```
const [first, second, ...rest] = [1, 2, 3, 4, 5];
// first: 1, second: 2, rest: [3, 4, 5]
```

4.2 Destructuring (Objects)

Facilitates extracting values from object properties.

```
const person = { name: 'John', age: 30, country: 'USA' };
const { name, age } = person;
// name: John, age: 30
```

mastering these array methods and ES6 features will greatly enhance your JavaScript development skills, promoting cleaner and more efficient code.

Check_Out_Detailed_Blog:-https://medium.com/@srivastavayushmaan1347/mastering-javascript-array-methods-and-es6-features-dd4f24eed19f