Advanced Javascript

Day 14 - Destructuring

ES6 allows you to write the code in such a way that makes your code more modern and readable.

By using ES6 features, we write less and do more, so the term 'Write less, do more' suits ES6.

Destructuring

```
const { name } = user;

const user = {
  'name': 'Alex',
  'address': '15th Park Avenue',
  'age': 43
}
```

```javascript
const user = {
    'name': 'Alex',
    'address': '15th Park Avenue',
    'age': 43
}
const { name, age } = user;


  console.log(name, age, salary); // Output, Alex 43

  123455
```

```javascript
const user = {
    'name': 'Alex',
    'address': '15th Park Avenue',
    'age': 43
}

const { address: permanentAddress } = user;


  console.log(permanentAddress); // 15th Park Avenue
```

```javascript
const user = {
        'name': 'Alex',
        'address': '15th Park Avenue',
        'age': 43,
        'department':{
            'name': 'Sales',
            'Shift': 'Morning',
            'address': {
                'city': 'Bangalore',
                'street': '7th Residency Rd',
                'zip': 560001
            }
        }

    }


  const { department: { address } } = user;
```

```javascript
const user = {
    'name': 'Alex',
    'address': '15th Park Avenue',
    'age': 43

}


  logDetails(user);



function logDetails({name, age}) {
    console.log(`${name} is ${age} year(s) old!`)

}
```

spread

ES6 introduced a new operator referred to as a spread operator, which consists of three dots (...). It allows an iterable to expand in places where more than zero arguments are expected. It gives us the privilege to obtain the parameters from an array.

```
1.    let colors = ['Red', 'Yellow'];

2.    let newColors = [...colors, 'Violet', 'Orange', 'Green'];

3.    console.log(newColors);
```

```
const user = {
    'name': 'Alex',
    'address': '15th Park Avenue',
    'age': 43
}


const updatedUser = {...user, salary:12345};
```

Spread operator and Strings

Here, we have constructed an array **str** from individual strings.

```
1.  let str = ['A', ...'EIO', 'U'];

2.  console.log(str);
```

Rest

The rest parameter is introduced in ECMAScript 2015 or ES6, which improves the ability to handle parameters. The rest parameter allows us to represent an indefinite number of arguments as an array. By using the rest parameter, a function can be called with any number of arguments.

```javascript
const user = {
    'name': 'Alex',
    'address': '15th Park Avenue',
    'age': 43

  }



const {age, ...rest} = user;

  console.log(age, rest);
```

```javascript
function show(...args) {

  let sum = 0;

  for (let i of args) {

    sum += i;

  }

  console.log("Sum = "+sum);

}


show(10, 20, 30);
```

Rest parameter: collecting

Spread operator: expanding