

JS-중급 9

≡ 태그	
📅 날짜	@2023년 6월 2일

나머지 매개변수, 전개 구문(Rest parameters, Spread syntax)

인수 전달

```
function showName(name){
// name 매개변수의 갯수는 제한이 없다.
  console.log(name);
}

showName('Mike'); // 'Mike'
showName('Mike', 'Tom'); // ?

showName(); // undefined

// 정확히 매개변수의 갯수를 맞출 필요 없다.
// 하나도 넣지 않아도 에러는 발생하지 않지만 undefined
```

함수의 인수를 얻는 방법

1. arguments
2. 나머지 매개 변수

arguments :

- 함수로 넘어온 모든 인수에 접근할 수 있다.
- 함수 내에서 이용 가능한 지역 변수
- length / index
- Array 형태의 객체
- 배열의 내장 메서드 없음 (forEach, map)

```
function showName(name){
  console.log(arguments.length);
  console.log(arguments[0]);
  console.log(arguments[1]);
}
```

```

showName('Mike', 'Tom');
//2
//'Mike'
//'Tom'

-----

ES6환경이라면 나머지 매개변수(Rest parameters)를 권장
매개변수(Rest parameters) : ...
정해지지 않은 갯수의 인수를 배열로 나타나게 한다.

function showName(...name){
  console.log(names);
}

showName(); // []
showName('Mike'); // ['Mike']
showName('Mike', 'Tom'); // ['Mike', 'Tom']

// 나머지 매개 변수
// 전달 받은 모든 수를 더해야함
function add(...numbers) {
  let result = 0;
  numbers.forEach((num) => (result += num));
  console.log(result);
}
add(1,2,3);
add(1,2,3,4,5,6,7,8,9,10);
6
55

-----

// 나머지 매개 변수
// user 객체를 만들어 주는 생성자 함수를 만듦

function User(name, age, ...skills){
  this.name = name;
  this.age = age;
  this.skills = skills;
}

const user1 = new User('Mike', 30, 'html', 'css');
const user2 = new User('Tom', 20, 'JS', 'React');
const user3 = new User('Jane', 10, 'English');

console.log(user1);
console.log(user2);
console.log(user3);

// [object Object]
{

```

```

    "name": "Mike",
    "age": 30,
    "skills": [
      "html",
      "css"
    ]
  }
// [object Object]
{
  "name": "Tom",
  "age": 20,
  "skills": [
    "JS",
    "React"
  ]
}
// [object Object]
{
  "name": "Jane",
  "age": 10,
  "skills": [
    "English"
  ]
}

```

나머지 매개변수는 항상 마지막에 위치해야한다.

전개 구문(Spread syntax) : 배열

```

let arr1 = [1,2,3];
let arr2 = [4,5,6];

let result = [...arr1, ...arr2];

console.log(result); // [1,2,3,4,5,6]

```

```

let result = [0, ...arr1, ...arr2, 7,8,9];

```

전개구문 : 객체

```

let user = {name : 'Mike'}
let mike = {...user, age: 30}

console.log(mike) // {name : 'Mike', age : 30}

```

전개구문 : 복제

```

let arr = [1,2,3];
let arr2 = [...arr]; // [1,2,3]

let user = {name : 'Mike', age : 30};
let user2 = {...user}

user2.name = "Tom";

console.log(user.name); // "Mike"
console.log(user2.name); // "Tom"

```

```

// 전개 구문
// arr1을 [4,5,6,1,2,3]으로

```

```

let arr1 = [1,2,3];
let arr2 = [4,5,6];

arr2.forEach((num)=>{
  arr1.unshift(num);
});

```

```

console.log(arr1);

```

```

// [object Array] (6)
[6,5,4,1,2,3]

```

```

// 전개 구문
// arr1을 [4,5,6,1,2,3]으로

```

```

let arr1 = [1,2,3];
let arr2 = [4,5,6];

arr2.reverse().forEach((num)=>{
  arr1.unshift(num);
});

```

```

console.log(arr1);
// [object Array] (6)
[4,5,6,1,2,3]

```

```

// 전개 구문
// arr1을 [4,5,6,1,2,3]으로

```

```

let arr1 = [1,2,3];
let arr2 = [4,5,6];

```

```
arr1 = [...arr2, ...arr1]

console.log(arr1);
// [object Array] (6)
[4,5,6,1,2,3]
```

객체

```
let user = {name : "Mike"};
let info = {age : 30};
let fe = ["JS", "React"];
let lang = ["Korean", "English"];

user = Object.assign({}, user, info, {
  skills : [],
});

fe.forEach((item) => {
  user.skills.push(item);
});
lang.forEach((item) => {
  user.skills.push(item);
});

console.log(user);

// [object Object]
{
  "name": "Mike",
  "age": 30,
  "skills": [
    "JS",
    "React",
    "Korean",
    "English"
  ]
}

-----

let user = {name : "Mike"};
let info = {age : 30};
let fe = ["JS", "React"];
let lang = ["Korean", "English"];

user = {
  ...user,
```

```
    ...info,  
    skills : [...fe, ...lang]  
  }  
  console.log(user);  
  
  // [object Object]  
  {  
    "name": "Mike",  
    "age": 30,  
    "skills": [  
      "JS",  
      "React",  
      "Korean",  
      "English"  
    ]  
  }  
}
```