

# JS기본-3

☰ 태그	
📅 날짜	@2023년 5월 26일

함수(function)

console, alert, confirm 은 내장 함수이다

```
함수      함수명      매개변수
function sayHello(name){
  console.log('Hello, ${name}');
}
```

매개변수가 여러개 라면 , 로 구분한다.

```
// 함수 작성
// 매개변수가 없을 경우 괄호 안을 비우면 된다.
function showError() {
  alert('에러발생 다시시도');
}

showError();

// 이 에러 메시지를 한 서비스에서 100개 넘게 사용한다면
// 매번 길게 입력할 필요없이 사용할 수 있다.
// 또한 유지보수가 쉽다. alert 만 고쳐주면 된다.

-----

// 함수 작성
// 매개변수가 있는 함수
```

```
function sayHello(name){
  const msg = `hello, ${name}`
  console.log(msg);
}
```

```
sayHello('mike');
sayHello('mino');
sayHello('hongsi');
// hello, mike
// hello, mino
// hello, hongsi
```

-----

만약 로그인하지 않아서 사용자의 이름을 모르면 어떻게 할까

```
function sayHello(name){
  let msg = `hello`;
  if (name) {
    msg = `hello, ${name}`;
  }
  console.log(msg);
}
```

```
sayHello();
```

```
// hello
```

```
-----
function sayHello(name){
  console.log(name); // undefined
  let msg = `hello`;
  if (name) {
    msg = `hello, ${name}`;
  }
  console.log(msg); // hello
}
```

```
sayHello();
// 매개변수가 비어있어서 if문이 실행되지 않아
// hello만 실행되는 것이다.
```

```
-----
function sayHello(name){
  let msg = `hello`;
  if (name) {
    msg += ', ' + name;
  }
  console.log(msg);
}
```

```
sayHello(); // hello
sayHello('mike'); // hello, mike
```

```
// msg는 함수 밖에서 사용할 수 없다.
// 함수 내부에서 선언한 변수를 지역변수라고 하는데
// 함수 내부에서만 사용할 수 있다.
```

```
// 전역변수와 지역 변수

let msg = "welcome"; // 전역 변수

function sayHello(name){
  let msg = "hello" // 지역변수
  console.log(msg + ' ' + name);
}

sayHello('Mike');
console.log(msg)

// 함수 내부에서 let으로 전역 변수와 동일한 이름으로 선언할 수 있고
// 서로 간섭을 받지 않는다.

"welcome"
"hello Mike"
"welcome"
```

```
// 전역변수와 지역 변수

let name = 'mike';

function sayHello(name){
  console.log(name)
}

sayHello();
sayHello('jane');

undefined
"jane"

// 매개변수로 받은 값은 복사된 후 함수의 지역변수가 된다.
// 전체 서비스에서 공통으로 바라봐야 되는 바운스를
// 제외하고는 지역 변수를 쓰는 습관을 들이는게 좋다.
```

```
//OR

function sayHello(name){
  let newName = name || 'friend';
  let msg = `Hello, ${newName}`
  console.log(msg)
}

sayHello();
sayHello('jan');

"Hello, friend"
"Hello, jan"

// 매개변수가 없으면 언디파인드가 들어가기 때문에
// let newName = name || 'friend'; 여기서
// false기 때문에 friend가 나오게된다.
```

```
// return으로 값 반환
// 두개의 값을 입력 받아 값을 반환

function add(num1, num2){
  return num1 + num2;
}

const result = add(2,3);
console.log(result);

5
```

```
function showError(){
  alert('에러가 발생했습니다. ');
}

const result = showError();
console.log(result)

// alert만 띄어진다. 그리고 undefined 반환
// 리턴문이 없는 함수도 undefined를 반환한다.
// 혹은 리턴만 있어도 undefined를 반환한다.
```

```
// 리턴문이 있으면 그 즉시 리턴 오른쪽값을 반환한다.  
// 그래서 함수를 종료하는 목적으로도 사용한다.
```

tip

함수(function)

- 한번에 한작업에 집중
- 읽기 쉽고 어떤 동작인지 알 수 있게 네이밍
  - showError - 에러를 보여줌
  - getName - 이름을 얻어옴
  - createUserData - 유저데이터 생성
  - checkLogin - 로그인 여부 체크

읽기 쉽고 어떤 동작을 하는건지 이름만 보고 알 수 있게 이름을 짓는 것이 좋다.