

# 03. 함수

## 3.1 함수 기본 문법

자바스크립트에서 함수(function)는 특정 작업을 수행하는 코드 블록입니다.

함수는 재사용 가능한 코드를 작성할 수 있게 해준다.

함수는 입력값을 받아 처리하고 결과를 반환한다.

### 1. 함수 선언

- 함수는 `function` 키워드로 선언합니다.

```
function add(x, y) {  
    return x + y;  
}  
console.log(add(5, 10)); // 15
```

### 2. 함수 표현식

- 함수 표현식은 함수를 변수에 할당하는 방식

```
const add = function(x, y) {  
    return x + y;  
}  
console.log(add(5, 10)); // 15
```

### 3. 화살표 함수

- ES6에서 도입된 간결한 함수 표현.

```
function add(a, b) {  
    return a + b;  
}  
console.log(add(10, 20));  
  
const add2 = function (a, b) {  
    return a + b;  
}
```

```

}
console.log(add2(10, 20));

const add3 = (a, b) => {
  return a + b;
}
console.log(add3(10, 20));

const add4 = (a, b) => a + b;
console.log(add4(10, 20));

```

#### 4. 기본 매개변수 값

```

function greet(name = "Guest") {
  return `Hello, ${name}!`;
}

console.log(greet("Alice")); // "Hello, Alice!"
console.log(greet()); // "Hello, Guest!"

```

#### 5. 가변 인자 함수

```

function sum(...numbers) {
  let total = 0;
  for(let i of numbers) {
    total = total + i;
  }
  return total;
}

console.log(sum(1, 2, 3, 4)); // 10

```

#### 6. 스코프(Scope)

- **함수 스코프:** `var` 로 선언한 변수는 함수 내에서만 유효합니다.

```
function functionScopeExample() {
  if (true) {
    var x = 10; // 'var'는 함수 스코프
  }
  console.log(x); // 10: x는 functionScopeExample 함수 내에서 유효하므로 출력됨
}

functionScopeExample();
```

- **블록 스코프:** `let`, `const` 로 선언한 변수는 블록 내에서만 유효합니다.

```
function blockScopeExample() {
  if (true) {
    let y = 20; // 'let'은 블록 스코프
    const z = 30; // 'const'는 블록 스코프
  }
  console.log(y); // ReferenceError: y is not defined
  console.log(z); // ReferenceError: z is not defined
}

blockScopeExample();
```

- **글로벌 스코프:** 함수 외부에서 선언된 변수는 어디에서나 접근 가능합니다.

```
let globalVar = 'global';
function testScope() {
  let localVar = 'local';
  console.log(globalVar); // global
  console.log(localVar); // local
}
testScope();
// console.log(localVar); // 오류 발생
```

