

# 섹션 3. 함수

---

윤상현 작성

## 함수 소개와 기본 문법

---

함수 → 특정 기능을 수행하는 코드의 단위

---

```
// 함수 작성 기본 코드
function logText() {
  // 함수 본문
  console.log("hi");
}

// 함수 호출
logText();
```

Output:

```
hi
```

---

## 함수 파라미터와 반환 값

---

파라미터 → 함수 안에 넘길 어떤 값들

---

```
function logText(message) {
  console.log(message);
}

logText("hi");
```

Output:

```
hi
```

---

반환 값 → 함수가 실행이 완료된 후에 호출 위치에 남기는 값

---

```
function logText(message) {  
    return message;  
}  
  
let a = logText("hi");  
  
a
```

Output:

```
"hi"
```

```
function logText() {  
    return "a";  
}  
  
logText();
```

Output:

```
a
```

---

## 함수 선언문과 함수 표현식

---

```
// 함수 선언문  
function logText() {  
    console.log("hi");  
}  
  
// 함수 표현식  
let logText = function() {  
    console.log("hi");  
}
```

---

함수 선언문을 일반적으로 지향하는 사용 방법이라고 생각하면 된다.

---

```
// 파라미터와 반환값을 가진 함수 선언문  
function logText(message) {  
    return message;  
}  
  
let logText = function(message) {  
    return message;  
}
```

---

## 자바스크립트의 특이한 성질 - 일급 함수

---

일급 함수(First Class Function)

→ 함수도 변수처럼 자유롭게 할당하고 활용할 수 있다. 변수가 선언되고 사용될 수 있는 모든 곳에서 함수가 동일하게 사용될 수 있다.

---

Code:

```
function logText(message) {  
  message();  
}  
  
logText(function() {  
  console.log("hi");  
});
```

Output:

```
hi
```

---

## 화살표 함수 기본

---

Code:

```
// 함수 표현식  
var logText = function() {  
  console.log("hi");  
}  
  
// 화살표로 작성된 함수 표현식  
// ES6+에서 지원  
var logTextArrowFunction = () => {  
  console.log("hi")  
}  
  
logTextArrowFunction();
```

```
hi
```

---

## 화살표 함수 축약 문법 2가지

---

Code:

```
let logText = () => {  
  return 'a';  
}  
  
logText();  
  
let logMessage = () => 'a';  
  
logMessage();  
  
// 파라미터가 하나인 경우네는 소괄호 생략 가능  
let logNumber = num => {  
  return num;  
}  
  
logNumber(2);  
  
let logNumber= num => num;  
  
logNumber(1);
```

Output:

```
a  
a  
2  
1
```

---