



1강 - 함수 표현식과 화살표 함수

함수 표현식이란??

화살표 함수 (arrow function)

함수 표현식이란??

- 일반적으로 함수를 선언할 때는 아래와 같이 선언합니다.
- 이 방식을 **함수 선언문** 방식이라고 부릅니다.

```
function foo(param) {  
  console.log(param);  
}
```

- 자바스크립트에서는 함수를 하나의 값으로 보고 아래와 같이 **함수 표현식**을 이용하여 선언할 수도 있습니다.

```
const foo = function(param) {  
  console.log(param);  
};
```

- 호출시에는 변수 foo에 담긴 함수표현식을 변수 이름 foo로 호출합니다.

```
foo('hello'); // hello가 콘솔에 출력됨!
```

- 또한 자바스크립트는 선언한 함수를 변수에 담는 것도 가능합니다.

```
function smile() {  
  console.log('웃으면 행복합니다');  
}  
  
const hahaha = smile;  
  
smile(); // 웃으면 행복합니다 메시지 출력  
hahaha(); // 웃으면 행복합니다 메시지 출력
```

- 위에서 중요한 내용은 smile이라는 함수를 hahaha변수에 저장할 때 smile() 이렇게 뒤에 소괄호를 붙여 호출한게 아니라 소괄호 없이 집어 넣었다는 점입니다.
- `const hahaha = smile();` 이렇게 처리했다면 위와 같이 hello()를 호출하는 방법은 적용되지 않습니다.

화살표 함수 (arrow function)

- 함수는 함수 선언문 방식과 함수 표현식 방식으로 만들 수 있습니다.
- ES6부터는 더 간편한 방법으로 함수를 만드는 **화살표 함수** 문법이 추가되었습니다.

```
// 함수 선언문 방식
function add(n1, n2) {
  return n1 + n2;
}

// 함수 표현식 방식
const add = function(n1, n2) {
  return n1 + n2;
};

// 화살표 함수 방식
const add = (n1, n2) => {
  return n1 + n2;
};
```

- 위 3가지 방식은 모두 동일하게 2개의 정수를 전달하면 그 합을 반환하는 add라는 함수를 정의합니다.
- 화살표 함수는 함수 내부의 코드가 단 한 줄밖에 없다면 함수블록(중괄호)를 생략할 수 있습니다.
- 또한 그 한 줄의 코드가 리턴문(return)이라면 return키워드도 생략이 가능합니다.

```
// 화살표 함수 방식
const add = (n1, n2) => {
  return n1 + n2;
};
```

```
};  
  
// 좀 더 간략하게 개선  
const add = (n1, n2) => n1 + n2;
```

- 만약 매개변수가 없다면 빈 소괄호를 사용합니다.
- 매개변수가 단 한 개라면 소괄호를 생략할 수 있습니다.

```
function hello() {  
  console.log('안녕?');  
}  
  
// 매개변수가 없는 경우의 화살표 함수  
const hello = () => console.log('안녕');  
  
// =====  
  
function sayName(name) {  
  console.log(`${name}님 반가워요~`);  
}  
  
// 매개변수가 한 개인 경우의 화살표 함수  
const sayName = name => console.log(`${name}님 반가워요~`);
```

reference :

함수 표현식

 <https://ko.javascript.info/function-expressions>



JAVASCRIPT.INFO
The Modern JavaScript Tutorial

화살표 함수 기본



<https://ko.javascript.info/arrow-functions-basics>



JAVASCRIPT.INFO
The Modern JavaScript Tutorial