

6_JS(ES6)_PART_03

- 함수
- 함수 선언식과 함수 표현식
- 화살표 함수

■ 함수

: 함수는 입력과 출력이 존재하는 로직의 단위

: 자신의 외부 코드가 호출할 수 있는 하위 프로그램 (재사용)

1. 함수 선언식과 함수 표현식

함수 선언식	함수 표현식(권장)
<pre>function 함수명(){ 구현 로직 }</pre>	<pre>const 함수명 = function() { 구현로직 }</pre>
호이스팅 영향받음	호이스팅 영향 받지 않음
선언 위에서 호출 가능	클로저, 콜백 사용 가능

함수 표현식을 권장하는 이유

: 함수를 만들기도 전에 호출이 된다면 예상하지도 못한 결과물이 만들어 질 수도 있기 때문에 함수 표현식을 권장한다.

2. 화살표 함수

- ES6 버전에서 추가된 함수의 형태
- 사용을 권장함, 간결하기 때문

```
const 함수명 = function () {  
  구현 로직  
}
```

```
const 함수명 = () => {\n  구현 로직  
}
```

(1) 함수선언식 Function Declaration

```
function calcAge1(birthYear) {  
  return 2023 - birthYear;  
}  
console.log(calcAge1(1993));
```

호이스팅으로 인해 먼저 호출을 해줘도 올바른 값이 나온다.

(2) 함수 표현식 Function Expression

```
const calcAge2 = function (birthYear) {  
  return 2023 - birthYear;  
};  
  
console.log(calcAge2(1992));
```

함수 표현식으로 만들어진 코드는 반드시 initialization 초기 선언이 되어지고 나서 호출을 해야 한다.

(3) 화살표 함수 Arrow Function

```
const calcAge3 = (birthYear) => {  
  return 2024 - birthYear;  
};  
  
console.log(calcAge3(1992));
```

■ Summary

- 함수는 입력과 출력이 존재하는 로직의 단위이며 호출이 가능
- 함수 표현식을 권장하고, 그 중 화살표 함수의 사용을 권장