

12. 함수

▼ 함수란?

일련의 과정을 문으로 구현하고 코드 블록으로 감싸서 하나의 실행단위로 정의한 것.

▼ new Function

string으로 전달된 코드를 실행할 수 있다.

eval()과 달리 함수 스코프를 넘어선 전역 범위에 접근 할 수 없어 eval보다 안전하다.

그러나 역시 XSS공격에 취약하다는 단점이 있다.

▼ 함수 선언문과 함수 표현식

함수 선언문

```
function foo() {  
    return 'something';  
}
```

함수 표현식

```
const foo = function () {  
    return 'something';  
}
```

무엇을 사용해야 하는가?

→ 각각의 장단점이 있으나 중요한 것은 통일성. 표현식을 사용한다면 모두 표현식으로 쓰는 것이 중요

▼ 즉시 실행 함수를 통한 모듈화

함수는 지역 scope를 가진다.

그러므로 전체를 즉시 실행함수로 감싸서 모듈화를 진행할 수 있다.

그러나 요즘은 `<script src='path' type='module'> </script>` 를 사용해서 모듈화 진행
mjs를 사용할 경우 각각의 파일이 독립된 scope를 가진다.

▼ 순수함수란?

어떤 외부 상태에 의존하지도, 변경하지도 않는 부수 효과가 없는 함수.

```
function add (x, y) {  
  return x + y;  
}
```

비순수 함수란?

```
const foo = 1;  
  
function notPureFunc(x, y) {  
  foo++;  
  return x + y;  
}
```

외부 변수를 바꾸게 되거나, 의존성을 지님. → Side Effect라고 부름.

▼ 참고, React의 순수함수와 Side Effect

- React에서 컴포넌트는 함수로 이루어진다.
- 그리고 이 함수의 순수성이란 JSX를 return 해야 한다.
- Strict mode에서 컴포넌트가 2번 실행되는 이유 역시 같은 값이 return되는지 확인하기 위함이다.

- Side Effect란?
 - JSX를 생성하는 동작이외의 모든 행위
 - 이들은 보통 `useEffect`에서 실행된다.
 - `data fetching`, `dom 조작` 등이 있다.