

Ask Company



리액트와 함께 장고 시작하기 / 리액트

# 순수 함수와 커링 기법

여러분의 파이썬/장고 페이스메이커가 되겠습니다.

# 리엑트는 함수형 프로그래밍을 적극 활용

컴포넌트의 많은 루틴을 **순수 함수**로서 작성하기를 요구

상태값/속성값이 같으면, 항상 같은 값을 반환해야 합니다.

다른 Side effects를 발생시키지 않아야 합니다. (HTTP 요청, 데이터 저장, 쿠키 조작 등)

컴포넌트의 상태값은 **불변 객체 (Immutable Object)**로 관리해야만 합니다.

수정할 때에는 기존 값을 변경하는 것이 아니라, 같은 이름의 새로운 객체를 생성합니다.

이를 통해, UI개발의 복잡도를 낮추고, 버그 발생 확률도 줄입니다.

jsx 문법으로 작성된 Header 리엑트 컴포넌트

```
const Header = (props) => (  
  <div>  
    <h1>{props.title}</h1>  
  </div>  
)
```

# 순수 함수

하나 이상의 인자를 받고, 인자를 변경하지 않고, 참조하여 새로운 값을 반환

Side Effects가 없도록 구성

```
let tom = {  
  name: "Tom",  
  canRun: false  
};
```

```
function not_pure_fn() {  
  tom.canRun = true;  
}
```

```
function pure_fn1(person) {  
  return {  
    ...person,  
    canRun: true  
  };  
}
```

```
const pure_fn2 = (person) => ({  
  ...person,  
  canRun: true  
});
```

# 순수 함수를 활용한 데이터 변환

reduce, filter, map, join 등

javascript

```
const numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];  
const number = numbers.reduce((acc, n) => acc + n, 0);  
console.log(number);
```

```
const even_numbers = numbers.filter(i => i % 2 == 0);  
console.log(even_numbers);
```

비교) python

```
from functools import reduce
```

```
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]  
number = reduce(lambda acc, n: acc + n, numbers, 0)  
print(number)
```

```
even_numbers = filter(lambda i: i % 2 == 0, numbers)  
print(tuple(even_numbers))
```

# 커링 (Currying)

리액트 전반적으로 많이 사용됩니다.  
(고차 컴포넌트, Hook 등)

파이썬에서는 장식자(Decorators)에  
사용되는 기법

## 일부의 인자를 고정한 새로운 함수를 반환하는 함수를 만드는 기법

### javascript #1

```
function userLogs(username) {  
  function wrap(message) {  
    console.log(`${username} - ${message}`);  
  }  
  return wrap;  
}  
  
const log = userLogs('Chinseok Lee');  
log('Hello World');
```

### javascript #2

```
const userLogs = username => message => {  
  console.log(`${username} - ${message}`);  
};  
  
const log = userLogs('Chinseok Lee');  
log('Hello World');
```

### 비교) python

```
def userLogs(username):  
    def wrap(message):  
        print(f"{username} - {message}")  
    return wrap  
  
log = userLogs('Chinseok Lee')  
log('Hello World')
```

Life is short.  
You need Python and Django.

I will be your pacemaker.

