#웹프로그래밍 응용

함수합성

소프트웨어학과 2022764034 이이슬

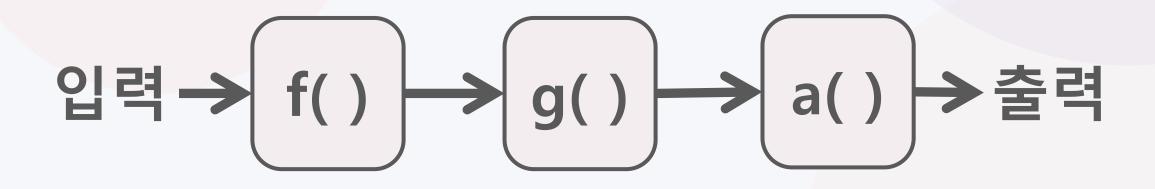
함수합성

function composition (함수합성)

프로그램



함수합성





함수합성 장점



작은 함수들이 독립적으로 동작하므로 코드 구조가 명확해지고 이해가 쉬움

작은 함수들은 다른 프로 젝트에서도 재사용 가능 작은 함수들을 독립적으로 테스트하고 수정할 수 있 어 전체 코드 관리가 용이



실습 1

```
const minus = num => num-1;
const plus = num => num+2;
const multiply = num => num*3;
```

console.log(minus(multiply(plus(5))));



pipe, compose

pipe, compose: 간단한 함수들을 조합하여 복잡한 함수를 만드는 강력한 도구

pipe() 함수: 여러 함수를 결합하며, 왼쪽에서 오른쪽으로 진행됨

compose() 함수 : pipe()와 비슷하지만, 함수 실행순서가 반대임 오른쪽에서 왼쪽으로 진행됨



pipe() 실습

```
const pipe = (f,g,a) => (data) => a(g(f(data)));
const minus = num => num-1;
const plus = num => num+2;
const multiply = num => num*3;

const result = pipe(plus,multiply,minus)(5);
console.log(result);
```

- 왼쪽에서 오른쪽으로 실행됨

compose() 실습

```
const compose = (f,g,a) => (data) =>f(g(a(data)));
const minus = num => num-1;
const plus = num => num+2;
const multiply = num => num*3;

const result = compose(minus,multiply,plus)(5);
console.log(result);
```

- 오른쪽에서 왼쪽으로 실행됨

실습 텍스트

실습1

```
const minus = num => num-1;
const plus = num => num+2;
const multiply = num => num*3;
console.log(minus(multiply(plus(5))));
```

실습2 : pipe()

```
const pipe = (f,g,a) => (data) => a(g(f(data)));
const minus = num => num-1;
const plus = num => num+2;
const multiply = num => num*3;

const result = pipe(plus,multiply,minus)(5);
console.log(result);
```

실습2: compose()

```
const compose = (f,g,a) => (data) =>f(g(a(data)));
const minus = num => num-1;
const plus = num => num+2;
const multiply = num => num*3;

const result = compose(minus,multiply,plus)(5);
console.log(result);
```



참고자료

https://www.youtube.com/watch?v=OgBk44TRjLs

https://velog.io/@a0im/js-%ED%95%A8%EC%88%98%ED%95%A9%EC%84%B1Function-

Composition

https://velog.io/@nittre/JavaScript-%ED%95%A8%EC%88%98-

%ED%95%A9%EC%84%B1Function-Composition

https://mathbang.net/475#gsc.tab=0

https://nantes.tistory.com/355

https://developefeel.tistory.com/150

감사합니다 :D