

학교에서 실습 테스트해보자....!! 코드 실행 안되면 Code Runner 설치해보고 안되면... 큰일이고..

(슬라이드 1) 안녕하십니까 오늘 함수합성에 대해 발표하게 된 소프트웨어학과 22학번 이이슬입니다.

(슬라이드 2) function composition 함수합성은 말그대로 다수의 함수들을 합성해서 사용한다는 것 인데요.

(슬라이드 3) 저희가 함수형 프로그래밍을 보면 하나의 프로그램이 함수들의 집합이라고 볼 수 있죠. 그리고 또 하나의 함수가 또 다른 함수들의 집합으로 구성이 될 수도 있는 거고요.

그 함수 집합에 속해 있는 하나의 함수를 좀 더 자세히 보자면 입력 값을 전달하고 정해진 기능을 수행 후 출력 값을 반환하죠. 그리고 여기에서 반환된 값이 다음 함수의 입력 값으로 전달돼서 사용되기도 하고 거기에서 또 반환되는 출력 값을 또 그 다음 함수의 입력 값으로 전달하고 이런 식으로 계속 이어서 실행을 할 수 있는데요

(슬라이드 4) 이런 식의 그림이 그려지는 거죠. 입력 값을 받아와서 f라는 함수에 전달하고, f에서 반환된 값을 또 g라는 함수에 전달하고 또 마찬가지로 방법으로 a함수로 전달시켜서 최종 출력 값을 만들어 내는 것입니다. 이런 방법을 사용해서 우리가 처리할 입력 값 데이터를 함수들을 통해서 흐를 수 있게 만들어 주는 겁니다.

(슬라이드 5) 이러한 함수 합성 장점의 첫번째로는 높은 가독성, 작은 함수들이 독립적으로 동작하므로 코드 구조가 명확해지고 이해하기 쉽습니다. 두번째로는 강화된 재사용성, 작은 함수들은 다른 프로젝트에서도 재사용 될 수 있습니다. 마지막으로는 향상된 유지보수성, 작은 함수들을 독립적으로 테스트하고 수정할 수 있어 전체 코드 관리가 용이 해집니다.

(슬라이드 6) 이제 간단한 실습을 하나 해보겠습니다. 위와 같은 식을 구현하기 위해 입력 값에 플러스 2를 하는 plus함수, 입력 값에 마이너스 1을 하는 minus함수, 입력 값에 3을 곱하는 multiply함수, 이러한 함수 3개를 구현하고 플러스 함수의 결과 값을 멀티플라이 입력 값으로 전달, 그리고 나온 결과 값을 마이너스 함수의 입력 값으로 전달해 출력 값을 구합니다. 이를 구현한 코드는 ppt 9페이지에 텍스트로 넣었으니 복사 붙여넣기 해주시면 될 것 같습니다!!

(코드복붙) 이 코드는 화살표 함수로 3개의 함수를 정의하였고, 순차적으로 값을 출력하도록 console.log 명령어에 넣어주고 실행하면 결과값인 20이 나오게 됩니다.

(슬라이드 7) 다음으로 pipe, compose함수인데요. 이는 간단한 함수들을 조합하여 복잡한 함수를 만드는 강력한 도구입니다. pipe()함수는 여러 함수를 결합하며, 왼쪽에서 오른쪽으로 진행되며, compose()함수는 pipe와 비슷하지만 함수 실행순서가 반대로 오른쪽에서 왼쪽으로 진행됩니다.

(슬라이드 8) 이제 이 함수들을 구현해 볼 것인데요. 마찬가지로 9페이지에서 복사 붙여넣기 해주시면 됩니다! 먼저 pipe는 왼쪽에서 오른쪽으로 실행되기 때문에

```
const pipe = (f, g, a) => (data) => a(g(f(data)));
```

f, g, a함수가 주어진다면 f, g, a 순으로 실행되도록 이와 같이 작성해 주고,

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

```
console.log(result);
```

이와 같이 먼저 실행시켜야 할 함수를 왼쪽에서 오른쪽 순으로 실행시켜 줍니다.

다음으로 compose는 pipe와 반대로 오른쪽에서 왼쪽으로 실행되기 때문에 f, g, a가 주어질 때 a, g, f 순으로 실행되도록 이와 같이 작성해 줍니다.

```
const compose = (f, g, a) =>(data) => f(g(a(data)));
```

그리고 이와 같이 먼저 실행시켜야 할 함수를 오른쪽에서 왼쪽 순으로 실행시켜줍니다.

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

(슬라이드 9) 이상으로 발표를 마치겠습니다. 감사합니다.