

질문 답변 및 Takeaway

11/15 질문 답변 (수정)

Q1. 2번 실습문제에서 getGreeting이라는 함수에 user를 보내줘도 보내주지 않아도 결과 값은 같은데 전달해주어야 하는 이유가 있나요?

▼ 문제 코드

```
const user = {
  lastName: '코딩하는',
  firstName: '엘리스',
};

function getGreeting(user) {
  return <h1>Hello, {formatName(user)}!</h1>;
}

//getGreeting()의 결과값을 element에 저장합니다.
const element = getGreeting(user);
```

A1. 반드시 전달해줘야하는 것은 아닙니다. 결과값은 같으니까요. 그런데 이 경우엔 “넣어주는 게 좋다” 측에서 설득을 해볼게요.

수학 함수를 표현하는 방식을 봐봅시다. 아래 두 함수는 완전히 동일한 함수입니다. 표현 방식이 다를 뿐이죠.

$$y = x^2 \dots (1)$$

$$f(x) = x^2 \dots (2)$$

하지만 (2) 번 함수를 통해 추가로 알 수 있는 정보가 있죠. 바로 x값에의해 y값이 바뀔 수 있다는 부가 정보를 줍니다.

x값에 의해 y값이 변하는걸 (y값이 x값에 의존한다는걸)를 알아야하냐?!! 몰라도 되냐??!!에 따라서 전달을 해도 되고 안해도 되는데,

여기서는 개인적으로 넣어주는게 좀 더 좋을 수 있다고 생각을 하는게,

저희가 프로젝트가 커지면 함수가 어마무시하게 많아질건데 우선 어떤 함수가 어떤 기능을 하고 어떤 인풋을 받고 어떤 결과값을 리턴하는지 다 기억하기는 어려워요.

그래서 함수 위에 아래 코드와 같은 방식으로 이게 무슨 함수인지 알려주는 함수 설명서를 작성하기도 하는데요,

그러면 함수를 사용할 때 아래 스크린샷처럼 함수 설명이 보여지게 됩니다.

시나리오) 원하던 결과값이 리턴되지 않은 상황이 오고,

getGreeting(user)를 보게되면,

보내준 user 자체에 문제가 생겨서 그럴까?

아니면 보내준 user에는 이상이 없지만 getGreeting 내 실행문이 잘못되서 그런 것일까?

라는 식으로 문제를 해결할 때 두가지 접근이 가능해요. 하나가 잘못되었다는 걸 발견해서 해결이 되면 다른 건 굳이 확인을 안해도 될 수 있는거죠.

하지만 getGreeting()라는 함수를 보게되면,

어떤 실행문에서 문제가 생겼는지 모든 코드를 읽어봐야하는 상황이 오게 되는거죠.

프로그래밍에서 “divide and conquer 큰 문제를 작은 문제로 나누어 해결한다”라는 말이 있습니다.

user를 넘겨주게 되면 그것이 바로 divide and conquer?!

```
// 예제

/**
 * 웰컴메시지를 렌더링하는 컴포넌트
 * @param {string} msg 웰컴 메시지 내용
 * @returns html 웰커밍 태그
 */
const Welcome = ({msg}) => {
  return <h1>{msg}</h1>;
}
```

```
1 import Welcome from "../components/Welcome";
2
3 function App() {
4   return (
5     <div className="App">
6       <Welcome/>
7     </div>
8   );
9 }
10
11 export default App;
```

[🔍] Welcome

(alias) const Welcome: ({ msg }: string) => JSX.Element ×
import Welcome

웰컴메시지를 렌더링하는 컴포넌트

@param msg — 웰컴 메시지 내용

@returns — html 웰커밍 태그

Q2. "/" 절대적 경로가 언제 쓰이는지에 관해서...

제가 잘못 설명드렸던 개념... 다시 재정리해서 알려드립니다.

저희는 외부 파일을 임포트 할 때 쓰는 import 구문에서는 "/" 를 쓰지 않습니다.

대신 언제 쓰느냐?

1. 터미널에서 `cd /` 를 실행해서 루트 리렉토리로 이동할 때,
2. 아니면...아래처럼 redirect해주는 api를 만들 때

예를 들어 현재 위치가 "<http://localhost:3000/users/profile/edit>" 라고 할 때,

`res.redirect("/home")`가 실행되면 "<http://localhost:3000/home>" 로 이동

`res.redirect("../home")`가 실행된다면 "<http://localhost:3000/users/profile/edit/home>" 으로 이동하게 됩니다.

```
// node.js에서 api 만들 때 사용한 코드를 가져와봤습니다...
// 이해를 위한 코드가 아니라 저런식으로 res.redirect(<<절대경로>>); 사용함을 보여주려고 가져왔습니다
```

```
export const getEdit = async (req, res) => {
  const { id } = req.params;
  const {
    user: { _id },
  } = req.session;
  const image = await Image.findById(id);
  if (!image) {
```

```
    return res.status(404).render("404", { pageTitle: "Image not found." });
  }
  if (String(image.owner) !== String(_id)) {
    req.flash("error", "You are not an i owner!");
    return res.status(403).redirect("/");
  }
  return res.render("edit", { pageTitle: `Edit: ${video.title}`, video });
};
```

Q3. reconfig.js 내 base url 설정 방법

11/15 노션 페이지 끝 확인