



실무 중심!
FE 입문자를 위한 React

강사 소개



프론트엔드 개발자 유동균

소개

前 라인 파이낸셜 프론트엔드 개발자

활동

- 삼성SSAFY, 프로그래머스, 엘리스, 인프런 등 프론트엔드 개발 강의 다수 진행
- 웹 성능 최적화 프로젝트 및 강의 제작
- 마이크로소프트웨어 393호(JS 주제), 392호(블록체인 주제) 기고

강사님의 한마디

“리액트를 리액트답게 사용하기”

리액트가 시장을 지배해가는 과정에서 리액트와 관련된 강의나 학생들의 수요는 점점 많아지고 있습니다. 하지만 리액트 교육을 제대로 된 곳에서 듣지 않거나 스스로 학습 해서 활용하는 경우 리액트의 장점과 특성, 그리고 활용 이유에 대해서 이해하지 못한 상태로 개발을 하는 경우가 많습니다. 리액트에 대해서 이해를 하지 못한 상태로 개발을 하게 되면 리액트의 장점을 살리기 힘들고 겉보기에 구현만 된 사이트를 만드는 경우가 많습니다. 본 강의에서는 JSX의 기본문법부터 리액트의 특징과 서비스적인 구조를 함께 다루면서 리액트의 강점과 활용 이유에 대해서 이해하며 실제 서비스를 제작해 가는 과정에 대해서 경험해볼 수 있습니다.

Chapter 01 React 소개	1
1-1. React란?	1
1) React의 뜻	1
2) React의 장점	1
3) React의 트렌드화가 가능했던 이유	1
1-2. React의 특징	2
1) 컴포넌트 기반 설계	2
2) 가상 돔	2
3) 그 외	2
Chapter 02 JSX	3
2-1. JSX란?	3
1) JSX란?	3
2) JSX의 특징	3
2-2. JSX에서 Javascript 사용	4
1) Javascript 사용 방법	4
2-3. JSX에서의 조건문	5
1) JSX 안에서의 조건문을 사용하는 법	5
2-4. JSX에서의 반복문	6
1) JSX 안에서의 반복문을 사용하는 법	6
2-5. JSX 스타일링	7
1) JSX에 CSS를 적용하여 스타일링하는 법	7
2-6. JSX로 구구단 출력하기	8
1) 구구단을 출력하는 법	8

Chapter 03 컴포넌트	9
3-1. 컴포넌트란?	9
1) 컴포넌트의 뜻	9
2) 컴포넌트 제작	9
3) JSX와의 차이점	10
4) 컴포넌트 생성 시 주의사항	10
3-2. Props	11
1) Props의 개념	11
2) Props의 활용 팁	11
3-3. State	12
1) State의 개념	12
2) State가 만들어지는 과정	12
3-4. 클래스형 컴포넌트 vs 함수형 컴포넌트	13
1) 클래스형 컴포넌트	13
2) 함수형 컴포넌트	13
3) Hooks의 등장 이유	13
3-5. 스터디파이 강의 컴포넌트 만들기	14
1) 컴포넌트 제작 연습	14
Chapter 04 LifeCycle과 Hooks	15
4-1. Hooks 종류	15
1) Hooks 종류	15
2) Hooks 의 작동 실습	15
4-2. React 렌더링 과정	16
1) React의 렌더링 과정	16
2) React의 라이프 사이클	16

4-3. 아코디언 컴포넌트 만들기	17
1) 아코디언 컴포넌트	17
 Chapter 05 이벤트 핸들링	 18
5-1. 이벤트 연결하기	18
1) JSX에서 함수 연결	18
2) HTML과 차이점	18
5-2. 이벤트 종류	18
1) 활용 가능한 이벤트의 종류	19
2) 이벤트 실습	19
5-3. Form	19
1) Form 요소 컨트롤 패턴	20
2) Form 코드 입력 요소 실습	20
5-4. 설문조사 만들기	20
1) 설문조사 페이지 제작	21
 Chapter 06 React 환경 설정	 22
6-1. Node.js, Vs code 설치	22
1) React 개발에 필요한 환경을 설치하는 법	22
2) Node와 VS Code 에디터 설치	22
3) 직접 설치하는 실습	22
6-2. create-react-app 프로젝트 생성	23
1) create-react-app이라는 툴을 이용하여 React 환경을 구축	23
2) CRA를 사용	23
2) React 환경 설치 실습	23
6-3. create-react-app 구조 소개	24
1) CRA로 만들어진 프로젝트 내부 확인 실습	24

6-4. eslint, prettier 설정	25
1) eslint	25
2) package.json에 있는 eslint 설정을 따로 설정 파일로 분리하는 실습	25
3) prettier	25
4) vs code의 플러그인 기능을 사용하여 prettier를 설치 실습	25

실습 Chapter	25
------------	----

Chapter 01 React 소개

Chapter 01 React 소개

1-1. React란?

<이번 시간에 배워요!>

- React의 뜻
- React의 장점
- React의 트렌드화가 가능했던 이유

1) React의 뜻

사용자 인터페이스를 만들기 위한 Javascript 라이브러리 (SPA 라이브러리)

= 웹 개발을 쉽게할 수 있도록 도와주는 Javascript 라이브러리

프레임워크: 개발을 위한 기본 틀

라이브러리: 개발에 필요한 도구 또는 그 집합 전체

2) React의 장점

1. 트렌드

- 대중적인 이용
- 수요가 많음
- 방대한 생태계

2. 편리성

- 화면을 여러 단위로 쪼개고 재사용하는 식으로 코드 관리가 용이
- 여러 유용한 라이브러리들

3) React의 트렌드화가 가능했던 이유

-전통적인 웹 서비스의 문제 해결

-SPA 방식 즉 브라우저에서 화면을 그리는 방식을 도입

1-2. React의 특징

<이번 시간에 배워요!>

- 컴포넌트 기반 설계
- 웹 페이지를 구성하는 요소
- 그 외

1) 컴포넌트 기반 설계

컴포넌트 = 웹 페이지를 구성하는 요소 하나

- 의미 단위로 구성
- 재사용성 및 유지보수성 증가

2) 가상 돔

실제 DOM의 복사본으로 SPA에서의 동적인 변화를 효율적으로 관리하기 위해 사용
돔을 직접 변경하며 생기는 비효율을 해결

3) 그 외

- CSR 브라우저에서 화면 렌더링
- 웹 개발 뿐 아니라 앱 개발을 위한 라이브러리도 존재

Chapter 02 JSX

Chapter 02 JSX

2-1. JSX란?

〈이번 시간에 배워요!〉

- JSX란?
- JSX의 특징

1) JSX란?

Javascript 확장한 문법으로 Javascript XML이라는 의미 즉 HTML+ Javascript

실습코드 링크 : <https://codesandbox.io/s/2-1-29k09>

2) JSX의 특징

- JSX에서 사용되는 태그의 속성 이름이 다름
- 태그를 명시적으로 닫아줘야 함
- 하나의 태그로 감싸져 있어야 함

Chapter 02 JSX

2-2. JSX에서 Javascript 사용

<이번 시간에 배워요!>

- Javascript 사용 방법

1) Javascript 사용 방법

실습코드 링크 : <https://codesandbox.io/s/2-2-d23bt>

Chapter 02 JSX

2-3. JSX에서의 조건문

〈이번 시간에 배워요!〉

- JSX 안에서의 조건문을 사용하는 법

1) JSX 안에서의 조건문을 사용하는 법

실습코드 링크 : <https://codesandbox.io/s/2-3-qeb7z>

Chapter 02 JSX

2-4. JSX에서의 반복문

〈이번 시간에 배워요!〉

- JSX 안에서의 반복문을 사용하는 법

1) JSX 안에서의 반복문을 사용하는 법

실습코드 링크 : <https://codesandbox.io/s/2-4-uv16i>

Chapter 02 JSX

2-5. JSX 스타일링

〈이번 시간에 배워요!〉

- JSX에 CSS를 적용하여 스타일링하는 법

1) JSX에 CSS를 적용하여 스타일링하는 법

실습코드 링크 : <https://codesandbox.io/s/2-5-tkgtq>

실습을 통해 함께 학습하는 강의입니다.

2-6. JSX로 구구단 출력하기

〈이번 시간에 배워요!〉

- 구구단을 출력하는 법

1) 구구단을 출력하는 법

- 전 강의의 스타일링 법을 응용하여 구구단 출력

실습코드 링크 : <https://codesandbox.io/s/2-6-silseub-bjv3c>

실습을 통해 함께 학습하는 강의입니다.

Chapter 03 컴포넌트

Chapter 03 컴포넌트

3-1. 컴포넌트란?

<이번 시간에 배워요!>

- 컴포넌트의 뜻
- 컴포넌트 제작 - JSX와
의 차이점 - 컴포넌트 생
성 시 주의사항

1) 컴포넌트의 뜻

- React는 컴포넌트라는 단위로 화면이 구성
- 스스로 상태를 관리하는 캡슐화된 코드 조각



- (예시) 화면을 각 요소로 쪼갠 것
- 하나의 JSX를 반환하는 함수

2) 컴포넌트 제작

실습코드 링크 : <https://codesandbox.io/s/3-1-x5rc1>

Chapter 03 컴포넌트

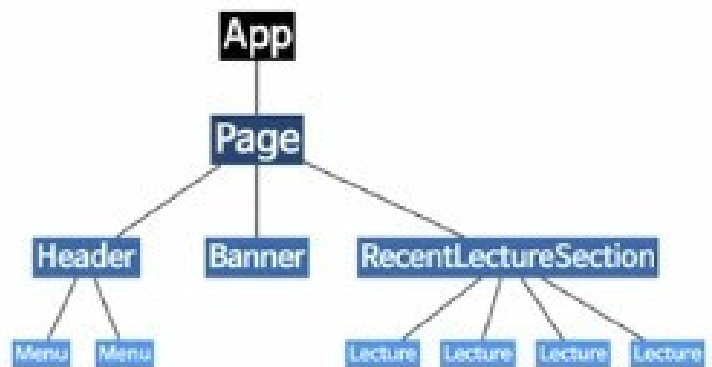
3) JSX와의 차이점

- 컴포넌트는 기본적으로 함수이기 때문에 자신만의 고유한 로직 존재
- 스스로의 상태 반영 가능

4) 컴포넌트 생성 시 주의사항

- 컴포넌트 이름은 무조건 PascalCase = 단어의 시작 문자는 대문자로 할 것
- 의미단위로 쪼개어 파일을 분리할 것

(예시)



- 최상위 컴포넌트 이름은 일반적으로 App

3-2. Props

〈이번 시간에 배워요!〉

- Props의 개념
- Props의 활용 팁

1) Props의 개념

props는 properties의 줄임말

부모 컴포넌트에서 자식 컴포넌트로 내려주는 데이터

2) Props의 활용 팁

- 구조분해할당 구문 활용
- 특정 Props에 기본 값을 지정 가능
- Props는 읽기 전용

실습코드 링크 : <https://codesandbox.io/s/3-2-uiqd9>

실습을 통해 함께 학습하는 강의입니다.

3-3. State

〈이번 시간에 배워요!〉

- State의 개념
- State가 만들어지는 과정

1) State의 개념

- 컴포넌트 내부에서 사용되는 일종의 변수
- 컴포넌트 스스로 상태를 관리하게 만드는 존재

2) State가 만들어지는 과정

실습코드 링크 : <https://codesandbox.io/s/3-3-mcfei>

실습을 통해 함께 학습하는 강의입니다.

3-4. 클래스형 컴포넌트 vs 함수형 컴포넌트

<이번 시간에 배워요!>

- 클래스형 컴포넌트
- 함수형 컴포넌트
- Hooks의 등장 이유

1) 클래스형 컴포넌트

- 클래식 문법으로 구현한 옛날 컴포넌트
- 아직도 사용되고 있고 라이브러리와 서비스에서 활용되고 있기에 학습

2) 함수형 컴포넌트

- 구조가 클래스보다 단순한 함수를 사용하기로 결정하며 대세로 사용되는 중
- 코드 재활용성에 유리한 새로운 컴포넌트 제작 방식

3) Hooks의 등장 이유

- 클래스형 컴포넌트의 단점을 보완하기 위해 함수형 컴포넌트에 지급된 개념

실습코드 링크 : <https://codesandbox.io/s/3-4-g2y3h>

실습을 통해 함께 학습하는 강의입니다.

3-5. 스터디파이 강의 컴포넌트 만들기

〈이번 시간에 배워요!〉

- 컴포넌트 제작 연습

1) 컴포넌트 제작 연습

1. 연습시 요구사항

- 스터디 파이에 있는 강의 요소로 직접 컴포넌트 구성
- 이미지 주소, 태그, 제목, 가격, 유형을 Props로 받아서 처리
- 가격은 3자리마다 콤마를 표기

실습코드 링크 : <https://codesandbox.io/s/3-5-njerw>

실습을 통해 함께 학습하는 강의입니다.

Chapter 04 LifeCycle과 Hooks

Chapter 04 LifeCycle과 Hooks

4-1. Hooks 종류

〈이번 시간에 배워요!〉

- Hooks 종류
- Hooks 의 작동 실습

1) Hooks 종류

기본 내장 Hooks는 10가지이며 사용되는 것은 3가지 정도 뿐

- useState
- useEffect
- useCallback

2) Hooks 의 작동 실습

실습코드 링크 : <https://codesandbox.io/s/4-1-x7uye>

실습을 통해 함께 학습하는 강의입니다.

4-2. React 렌더링 과정

<이번 시간에 배워요!>

- React의 렌더링 과정
- React의 라이프 사이클

1) React의 렌더링 과정

1. Rerendering

- 컴포넌트의 상태 즉 State 또는 Props가 변경되면서 해당 컴포넌트를 다시 실행하여 화면을 다시 그리는 것을 의미

2) React의 라이프 사이클

클래스형 컴포넌트와 함수형 컴포넌트로 구분 가능

1. 클래스형 컴포넌트의 라이프 사이클

- 컴포넌트가 실행되는 것 = 컴포넌트의 마운트
 - 마운트가 되었을 때 constructor 실행 → getDerivedStateFromProps 메소드 실행 → render 메소드 실행 → componentDidMount라는 메소드 실행
 - 메소드에서 컴포넌트를 다시 rendering 할 것인지, 이번 상태 변화에 따른 rerendering은 생략할 것인지를 결정
 - 함수가 false를 반환하면 렌더링을 진행하지 않고 true를 반환하면 렌더링을 진행
 - rendering이 완료되면 마무리로 componentDidUpdate를 실행
- 컴포넌트가 지워지는 것 = 언마운트

2. 함수형 컴포넌트의 라이프 사이클

- 마운트 → 바로 자기 자신을 실행 → 반환된 JSX 값을 DOM에 반영
- useEffect라는 Hook을 실행
- 클래스형 컴포넌트의 componentDidMount, componentDidUpdate, componentWillUnmount 메소드 = useEffect hook으로 통일
- 데이터 할 때는 다시 함수를 실행 → 그 JSX를 DOM에 반영
- useEffect hook을 실행
- 언마운트 시엔 useEffect 함수만 마지막으로 실행

실습코드 링크 : <https://codesandbox.io/s/4-2-8ek4h>

4-3. 아코디언 컴포넌트 만들기

〈이번 시간에 배워요!〉

- 아코디언 컴포넌트 제작 연습

1) 아코디언 컴포넌트

- 아코디언은 펼치기 기능 - 아코디언을 React의 컴포넌트로 구현 제작

실습코드 링크 : <https://codesandbox.io/p/sandbox/4-3-ly15l>

Chapter 05 이벤트 핸들링

Chapter 05 이벤트 핸들링

5-1. 이벤트 연결하기

<이번 시간에 배워요!>

- JSX에서 함수 연결
- HTML과 차이점

1) JSX에서 함수 연결

React 컴포넌트의 상태에 변화를 주는 것 = 이벤트

React에서는 모두 소문자인 onclick이 아니라 camelCase 규칙에 따라 가운데 C가 대문자인 onClick 속성에 이벤트 함수 추가 → 버튼이 클릭되었을 때 handleClick 함수 실행

2) HTML과 차이점

HTML에서는 소문자 onclick에 실행할 코드 자체를 넣어줬지만,

React에서는 camelCase의 onClick에 실행할 함수 추가

실습코드 링크 : <https://codesandbox.io/s/5-1-2poeh>

실습을 통해 함께 학습하는 강의입니다.

5-2. 이벤트 종류

<이번 시간에 배워요!>

- 활용 가능한 이벤트의 종류
- 이벤트 실습

1) 활용 가능한 이벤트의 종류

1. 마우스와 관련된 이벤트

- 마우스의 버튼이 눌렸을 때 발생 onMouseDown
- 누른 것을 다시 떼었을 때 발생 onMouseUp onClick
- 클릭의 의미가 눌렀다 떼 시 발생 onClick
- 의미특정 요소에 마우스 커서가 들어갔을 때 발생 onMouseEnter
- 다시 밖으로 나갔을 때 발생 onMouseLeave
- 마우스 커서를 움직이면 발생 onMouseMove

2. 키보드와 관련된 이벤트

- 키를 눌렀을 때 발생 onKeyDown, onKeyPress
- 누른 키를 떼었을 때 발생 onKeyUp

3. Focus 이벤트와 Form 관련 이벤트

- 어떤 요소에 포커스가 되면 onFocus 발생
- 포커스가 사라지면 onBlur 발생
- 텍스트 입력 박스와 같은 Form 요소의 값이 변경 되었을 때 onChange 발생

2) 이벤트 실습

실습코드 링크 : <https://codesandbox.io/s/5-2-z6ubs>

실습을 통해 함께 학습하는 강의입니다.

5-3. Form

〈이번 시간에 배워요!〉

- Form 요소 컨트롤 패턴
- Form 코드 입력 요소 실습

1) Form 요소 컨트롤 패턴

- Controlled Component 의 개념 = React에 의해 입력 요소의 값이 제어되는 컴포넌트
- TextInput이라는 컴포넌트가 input 요소를 렌더링하는 것
= value 속성에 text라는 state가 있다는 것
- onChange 이벤트 text 값을 입력 값으로 셋팅하는 setText 함수를 호출
사용자가 값을 입력 → text에 바로 입력한 값이 반영 → 상태 업데이트
- Controlled Component는 state를 직접 input 요소의 value에 연결시켜 주기 때문에 그 둘은 완전히 동일한 값

2) Form 코드 입력 요소 실습

실습코드 링크 : <https://codesandbox.io/s/5-3-crdmt>

5-4. 설문조사 만들기

〈이번 시간에 배워요!〉

- 설문조사 페이지 제작

1) 설문조사 페이지 제작

제작시 요구사항 만족

- 이름과 사는 곳을 입력받도록 구현
- TextInput 컴포넌트와 Select 컴포넌트 구현

실습코드 링크 : <https://codesandbox.io/s/5-4-2jxeb>

실습을 통해 함께 학습하는 강의입니다.

Chapter 06 React 환경 설정

Chapter 06 React 환경 설정

6-1. Node.js, Vs code 설치

<이번 시간에 배워요!>

- React 개발에 필요한 환경을 설치하는 법
- Node와 VS Code 에디터 설치
- 직접 설치하는 실습

1) React 개발에 필요한 환경을 설치하는 법

CodeSandbox의 환경을 직접 설치하여 실행

2) Node와 VS Code 에디터 설치

- 브라우저 밖에서도 javascript를 실행 가능
- 브라우저 밖에서 서버를 구축하는 등의 javascript를 실행할 수 있게 해주는 런타임 환경 제공 = 브라우저 밖에서도 Javascript를 실행할 수 있게 제공
- react 개발환경을 쉽게 구축 → Node.js가 필요
- 특징 오픈 소스 Javascript 엔진 크롬의 V8 엔진을 기반으로 동작
- 싱글 쓰레드 non-blocking I/O 이벤트 기반의 비동기 방식으로 동작
 - = 코드 실행 → 시간을 오래 잡아먹는 코드를 만났을 경우
 - 해당 코드의 실행이 끝날 때까지 기다리는 것 X
 - 나중에 시간이 지나서 해당 코드 실행이 완료되면 그 때 결과를 받아 따로 처리 O
- NPM(Node Package Manager) 이란 Javascript 라이브러리들을 쉽게 설치해서 사용할 수 있도록 도와주는 도구

3) 직접 설치하는 실습

링크 : <https://nodejs.org/ko/download/>

6-2. create-react-app 프로젝트 생성

<이번 시간에 배워요!>

- create-react-app이라는 툴을 이용하여 React 환경을 구축
- CRA를 사용
- React 환경 설치 실습

1) create-react-app이라는 툴을 이용하여 React 환경을 구축

- Create React App (줄여서 CRA)
- = React 기반의 프로젝트 개발 환경을 구성해주는 툴
- = React 개발을 더욱 편하게 할 수 있게 도와주는 굉장히 다양한 툴들과 환경들을 함께 설치 및 설정

2) CRA를 사용

- npx create-react-app memo-project 입력 => 설치 실행

- npx

= node를 설치할 때 npm과 함께 설치된 툴

= npm registry에 올라가 있는 최신 버전의 모듈 바로 실행 → memo-project라는 이름의 react 프로젝트를 생성 → 실행한 위치에 프로젝트 이름으로 된 폴더 생성 (react 개발 환경)

2) React 환경 설치 실습

6-3. create-react-app 구조 소개

〈이번 시간에 배워요!〉

- CRA로 만들어진 프로젝트 내부 확인 실습

1) CRA로 만들어진 프로젝트 내부 확인 실습

실습을 통해 함께 학습하는 강의입니다.

6-4. eslint, prettier 설정

<이번 시간에 배워요!>

- eslint
- package.json에 있는 eslint 설정을 따로 설정 파일로 분리하는 실습
- prettier
- vs code의 플러그인 기능을 사용하여 prettier를 설치 실습

1) eslint

- 문법 및 코드 스타일을 검사해주는 도구
- lint가 코드 모니터링 진행, 문제가 될 법한 코드를 째게 되면 에러메시지 송부
- 문법적 오류 사전 발견 가능, 통일된 코드 구성 가능
- CRA는 그중 eslint 사용
- 설정은 package.json에서 확인 가능

2) package.json에 있는 eslint 설정을 따로 설정 파일로 분리하는 실습

3) prettier

- 코드를 정해진 규칙에 맞게 자동으로 수정
- CRA에 포함되어 있지 않아 직접 설치 필요

4) vs code의 플러그인 기능을 사용하여 prettier를 설치 실습

실습 Chapter

실습 Chapter

챕터7부터 챕터18은 실습을 통해 함께 학습하는 강의입니다.

Chapter 07

메모장 만들기 코드: <https://github.com/hackurity01/memo-project>

Chapter 08 ~ Chapter 14

- survey-pie 코드: <https://github.com/hackurity01/survey-pie>
- survey-pie 디자인 가이드: <https://www.figma.com/file/ZZTXHmDabIXFIQRhlznkiT/design>

Chapter 15 ~ Chapter 18

- survey-pie-admin 코드: <https://github.com/hackurity01/survey-pie-admin>