

[React 공부 2일차]

▼ React 기본 문법 3가지

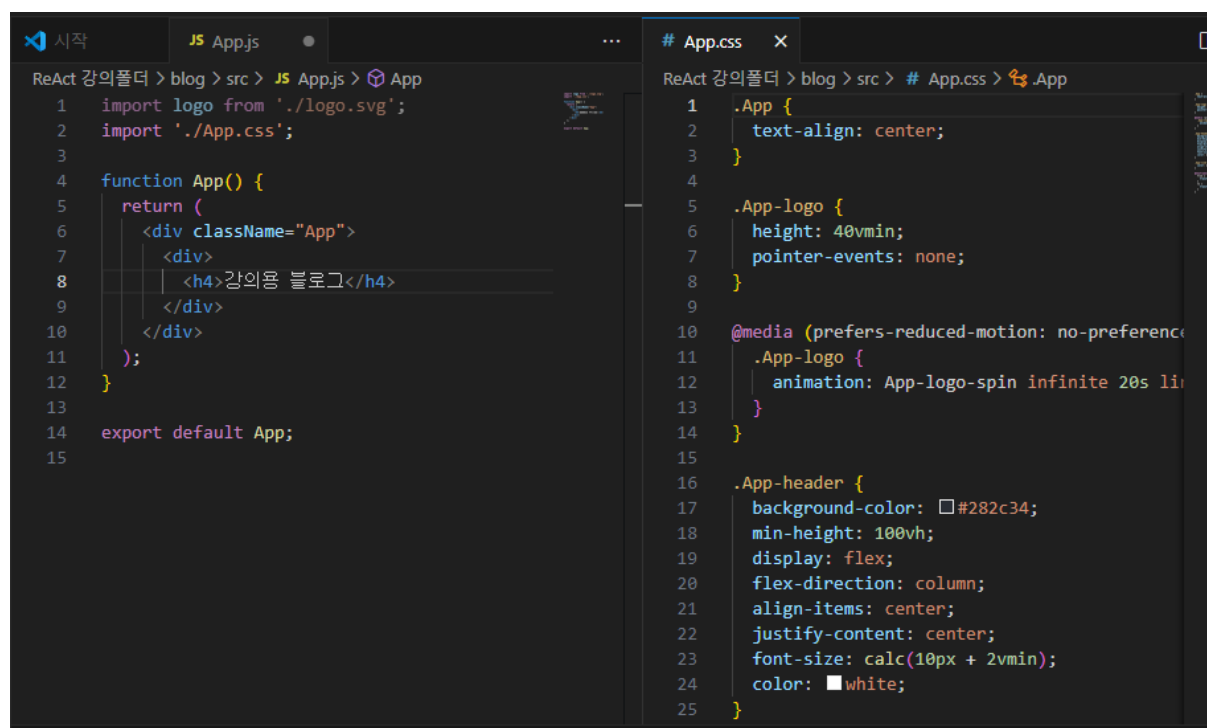
React 문법은 기본 html 문법과 js 문법을 이용하지만 조금은 다른 형태로 이용합니다.

```
function App() {  
  return (  
    <div className="App">  
      <header className="App-header">  
        <img src={logo} className="App-logo" alt="logo" />  
        <p>  
          리액트 공부 2일차  
        </p>  
        <a  
          className="App-link"  
          href="https://reactjs.org"  
          target="_blank"  
          rel="noopener noreferrer"  
        >  
          Learn React  
        </a>  
      </header>  
    </div>  
  );  
}
```

저번에 마지막으로 실행시키고 확인후에 종료했던 프로젝트를 불러옵니다.

```
ReAct 강의폴더 > blog > src > JS App.js > App
1  import logo from './logo.svg';
2  import './App.css';
3
4  function App() {
5    return (
6      <div className="App">
7      </div>
8    );
9  }
10
11  export default App;
12
```

안에 있던 기본내용들은 모두 삭제해주고 자신의 홈페이지를 만들기위해 문법을 공부해봤습니다.



```
ReAct 강의폴더 > blog > src > JS App.js > App
1  import logo from './logo.svg';
2  import './App.css';
3
4  function App() {
5    return (
6      <div className="App">
7        <div>
8          <h4>강의용 블로그</h4>
9        </div>
10     </div>
11   );
12 }
13
14 export default App;
15
```

```
ReAct 강의폴더 > blog > src > # App.css > .App
1  .App {
2    text-align: center;
3  }
4
5  .App-logo {
6    height: 40vmin;
7    pointer-events: none;
8  }
9
10 @media (prefers-reduced-motion: no-preference) {
11   .App-logo {
12     animation: App-logo-spin infinite 20s linear;
13   }
14 }
15
16 .App-header {
17   background-color: #282c34;
18   min-height: 100vh;
19   display: flex;
20   flex-direction: column;
21   align-items: center;
22   justify-content: center;
23   font-size: calc(10px + 2vmin);
24   color: white;
25 }
```

새로운 div를 생성하고 상단 배너에 제목을 정해주고 h4 로 지정해서 만들어줍니다.

그리고 뒤에 배경색과 글자색을 정해주기위해 css 를 켜줍니다.

이 과정은 기존에 사용하던 방식과 크게 다른점은 없으며 import 값으로 css를 불러오고있어 정상적으로 작동되고있는것을 확인가능했습니다.

```
function App() {
  return (
    <div className="App">
      <div className="black-nav">
        <h4>강의용 블로그</h4>
      </div>
    </div>
  );
}
```

다른점이 있다면 div 에 class를 주기위해 <div class="이름"> 으로 하면 js 문법이라 인식을 못해서 jsx 에서는 className 이란 태그를 사용해주었습니다.

```
let post = '강남 우동 맛집';
```

이제 데이터를 만들어주고 이 데이터를 활용하여 코딩을 해주기에 앞서

```
document.querySelector('h4').innerHTML = post;
```

기존 html 에서는 let 으로 지정해준 데이터를 사용하기위해선 직접 html 태그를 이용하여 넣어주었어야하는 반면에 jsx 문법에서는

```
<h4>{ post }</h4>
```

{ 중괄호 } 를 이용하여 데이터를 바인딩해줍니다. jsx 문법에선 {중괄호} 가 데이터 바인딩을 담당하고있었습니다.

```
<h4 id={ post }>강의용 블로그</h4>
```

{ 중괄호 } 는 어디서든 사용이가능하며 이런식으로 id 값에 넣어주어도 제대로 값이 들어가는것을 확인할 수 있었습니다.

```

<div id="root">
  <div class="App">
    <div class="black-nav"> flex
      <h4 id="강남 우동 맛집">강의용 블로그</h4>
    </div> = $0
    <h4>강남 우동 맛집</h4>
  </div>
</div>
<!--

```

f12 개발자도구를 사용하여 들어가보면 h4 id 값에 post 값이 제대로 들어간것이 확인되는 모습입니다.

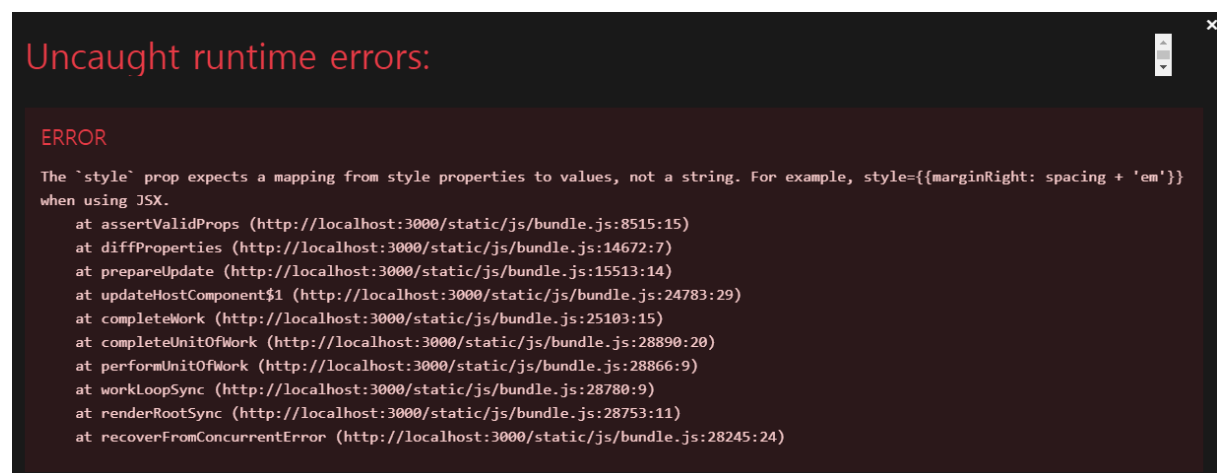
jsx 에서도 css 를 사용하지않고 직접적으로 style 값을 지정해줘서 사용이 가능합니다.
배우기전 기존에 사용했던 문법을

```

<div className="black-nav">
  <h4 style="color : red">강의용 블로그</h4>
</div>

```

이런 식으로 넣어준다면 오류가 나는것을 확인할 수 있습니다.



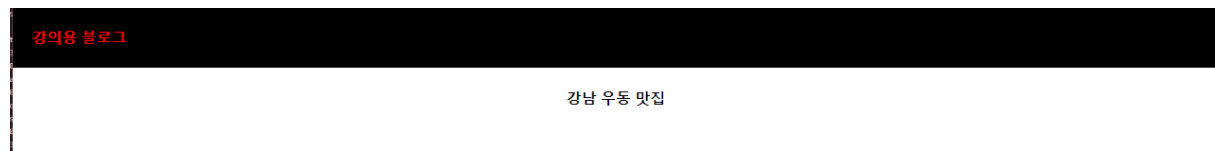
jsx 문법에서 style 값을 직접 주는법은

```

<div className="black-nav">
  <h4 style={ {color : 'red', fontSize : '16px'} }>강의용 블로그</h4>
</div>

```

{중괄호} 안에 {중괄호코드값} 을 넣어서 직접 지정해주면 정상적으로 작동하는것을 확인할 수 있습니다.



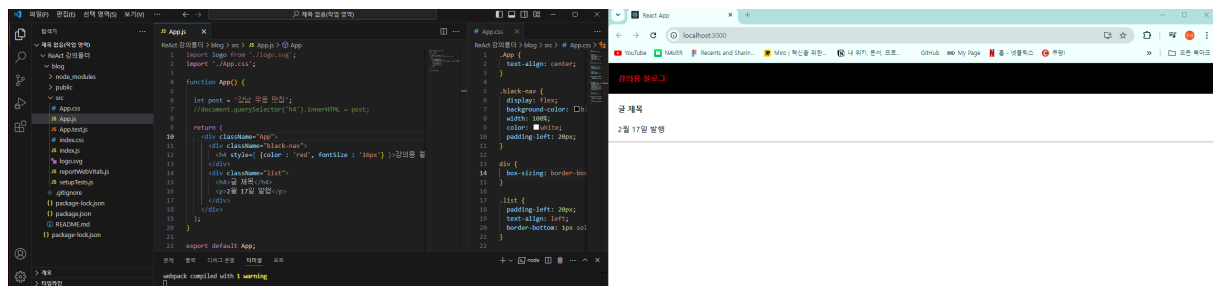
요약하자면

jsx 문법에선 class 지정은 className 을 사용합니다.

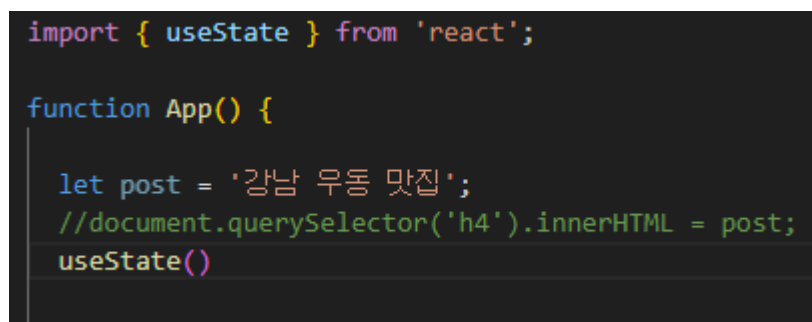
데이터 바인딩을 위해선 {중괄호} 를 활용합니다.

style 값을 직접 지정해줄때에도 { { } } 문법을 사용하여 지정해줍니다.

▼ react state문 사용



글 목록을 만들어주기위해 새로운 div와 css를 적용시켜준 모습입니다.



이번에 사용해볼 문법은 state 문입니다.

import 로 react 에서 useState 문을 연결시켜주고 아래에 입력해주면 연결이 잘 되어있는 것을 확인할 수 있습니다.

```
//document.querySelector('h4').innerHTML = post;
let [titleName, b] = useState('남자 코트 추천');
// a = state 에 보관했던 자료
// b = state 변경을 도와주는 함수
// [1,2,3] array 자료
let num = [1,2,3];
let a = num[0];
let c = num[1];
let [a, c] = [1, 2];
// Destructuring 문법
```

이런식으로 let 으로 변수를 만들어주고 useState 로 값을 지정해줍니다.

여기서 let [a, b] = useState('aa'); 으로 해준것은 Destructuring 문법으로 a 는 state 에 보관했던자료 b 에는 state 의 변경을 도와주는 함수를 넣는방식으로 사용합니다.

일반 변수로 html을 만들게되면 안에 값이 변경되었을때 재렌더링이 이루어지지않아서 값이 바뀌지않는 경우가있는데

useState문을 사용하면 데이터값이 변경되었을때 재렌더링에 들어가게되어 html값이 바로 변경이 됩니다.

요약하자면 쉽게 변경되거나 자주 변경되는 값은 state 로 지정해주고

오랫동안 변경되지않는값은 하드코딩 혹은 변수로 지정해놓으면 됩니다.

```

function App() {

  let post = '역삼 우동 맛집';
  //document.querySelector('h4').innerHTML = post;
  let [titleName, b] = useState('여자 코트 추천');
  let [logo, setLogo] = useState('ReactBlog');

  return (
    <div className="App">
      <div className="black-nav">
        <h4>{ logo }</h4>
      </div>
      <div className="list">
        <h4>{ titleName }</h4>
        <h4>{ post }</h4>
        <p>2월 17일 발행</p>
      </div>
    </div>
  );
}

```

글리스트를 만들어주는데 아래 list 목록을 3개로 만들것입니다.

```

import logo from './logo.svg';
import './App.css';
import { useState } from 'react';

function App() {

  let post = '역삼 우동 맛집';
  const listItem = [];
  //document.querySelector('h4').innerHTML = post;
  let [titleName, b] = useState(['남성 코트 추천', '여성 코트 추천', '강남 우동 맛집']);
  let [logo, setLogo] = useState('ReactBlog');

  for (let i = 0; i < 3; i++) {
    listItem.push(
      <div className="list">
        <h4>{ titleName[i] }</h4>
        <p>2월 17일 발행</p>
      </div>
    )
  }

  return (
    <div className="App">
      <div className="black-nav">
        <h4>{ logo }</h4>
      </div>
      <h4>{ listItem[0] }</h4>
      <h4>{ listItem[1] }</h4>
      <h4>{ listItem[2] }</h4>
    </div>
  );
}

export default App;

```

하드코딩과 배열변수를 사용하여 값을 지정해줍니다.

ReactBlog
남성 코트 추천
2월 17일 발행
여성 코트 추천
2월 17일 발행
강남 우동 맛집
2월 17일 발행

해서 출력이 정상으로 되었다면 state 문법 사용은 여기까지입니다.

연락처

✉ alarm153@Naver.com 📞 +82 10-4750-0732 🏠 서울시 관악구 신림동