

# JSX란?

#TIL/dev/web/babel/jsx

1. javascript extension이란 이름을 불리는 것 같고, React 공식문서에는 jsx없이도 사용할 수 있다고 하고 있으나 매우 강하게 결합되어있는 것 같다(구글링해보면 대부분 React 관련 문서에서 jsx를 설명하고 있다)
2. jsx는 어떤 녀석인가 궁금하여 조금 찾아봤는데 태생이 뭔지 모르겠다. jsx라는 표준이 있어서 Babel과는 독립적인 친구인지 찾아보려 했으나 관련 자료를 찾기는 힘들다. 아무래도 Babel-jsx는 묶어서 생각해도 무방할 것 같고, 이 친구들이 많은 경우 React와 조합되어 쓰인다고 보면 될 것 같다.

## 사용법

### 1. Variable in jsx

```
// Whenever you want to use an expression (something that produces a value) in
// JSX,
// you need to wrap the expression in single curly braces, `{}`.
render() {
  const name = 'Tyler'
  return ( <div> <h1>Hello, {name}</h1> <p>Today is {new
Date().toLocaleDateString()}</p> <p>What is 2 + 2? {2 + 2}</p> </div> )
}
```

### 2. Conditional Rendering

- 일반적으로 상태에 따른 UI 렌더링은 front-end framework의 기능으로써 제공된다: 프레임워크 내장 기능으로써.

```
// Angular
<h1 *ngIf="authenticated; else elseBlock">Welcome back!</h1>
<ng-template #elseBlock><h1>Login to see your dashboard</h1></ng-template>

// Vue
```

```
<h1 v-if="authenticated">Welcome back!</h1>
<h1 v-else>Login to see your dashboard</h1>
```

- 하지만 리엑트는 조금 다르다. 그들은 API를 늘리는 대신, 그러한 조건부 렌더링을 Javascript feature를 이용하여 구현하기로 하였다. 왜냐하면 jsx는 본질적으로 javascript이기 때문이다!

```
//if else
render() {
  if(true) {
    return <h1>well</h1>
  }
  else {
    return <h1>hoho</h1>
  }
}

//Ternary Operator
render() {
  return true ? <h1>well</h1> : <h1>haha</h1>
}

//in jsx, use `{}`
render() {
  return (
    <div>
      {true ? <h1>well</h1> : <h1>haha</h1>}
    </div>
  )
}
```

3. 아무 의미 없는 Root Node를 할당하고 싶다면 `React.Fragment` 혹은 `<>` 를 쓰면 된다.

#### 4. Capitalization

- React는 React Component와 build-in HTML element( like `<span>` )을 구분할까? 그것은 바로 Capitalization 에 있다. 따라서, React Component를 쓸때면 반드시 대문자로 명명하도록 해야 한다. 아니면 React는 build-in HTML element처럼 판단할 것이다.

## 5. 조건 분기

- if else 대신 `&&` or `||`
  - `&&`은 조건이 `true`일때 다음을 반환
- `||`은 조건이 `false`일때 다음을 반환

## 정의

1. `React.Component`는 `JSX.Element`를 리턴하는 함수 혹은 클래스이다.

`React Components Vs. JSX elements`

2. 결국 리액트 컴포넌트는 함수이며, 이를 호출하는 것과 JSX를 선언하는 것은 동일하다. JSX는 Syntactic sugar다. 관련 갑을 논박은 [minding\\_rn/pull/87](#) PR에서 확인가능.