

# React guide 1

## 1. 명칭을 마음대로 쓸 때와 그렇지 않은 경우

```
import React, { useState, useRef } from 'react';
import getAverage from '../pipes/Average';
```

- 되는 경우 : React, getAverage
- 안 되는 경우 : useState, useRef

\*즉 {} 안에 쓰여지면 정확한 이름을 써야 한다. 그렇지 않으면 오류가 난다.

## 2. 명칭을 마음대로 쓸 때와 그렇지 않은 경우 II

```
const navigate = value => {
  history.push(`/${value}`);
};
```

위 경우에서 괄호 안에 있거나 괄호가 없는 경우는 이름을 마음대로 써도 된다.  
즉 value라는 이름은 마음대로 쓸 수 있다.

```
const navigate = ({value}) => {
  history.push(`/${value}`);
};
```

위와 같이 {} 안에 있는 경우는 이름이 정해져 있으므로 그 이름을 변경하면 쓸 수 없다.

## 3. 명칭을 마음대로 쓸 때와 그렇지 않은 경우 III

arrow function(=>)을 쓸 때에도 마찬가지로 ({ value }) => 와 같은 형태에서는 value는 정해진 키를 의미하므로 변경할 수 없다. 그러나 (item) => 혹은 item => 와 같은 형태에서는 내 마음대로 변경이 가능하다.

map() 메서드와 같은 것을 쓸 때에도 마찬가지로 array.map(item => 라고 쓰여질 때 item은 마음대로 이름 변경이 가능하다. index가 포함될 때에는 array.map((item, index) => 와 같이 괄호가 포함된다. 이 때 item의 이름과 index의 이름은 마음대로 변경이 가능하다. **중요한 것은 순서다.** 앞에 있는 것이 배열에 포함된 n번째 object가 되는 것이고, 뒤에 오는 것이 n번을 의미하는 index가 된다.

## 4. useState에 대한 정의

react 내에서 사용되는 값은 state에 보관된다.

```
const [data, setData] = useState('');
```

위와 같은 형태로 사용이 되는데, 우리가 쓰는 값은 data위치에, 그리고 그 data를 변경할 수 있는 함수는 그 뒤에 넣어준다. 즉 data에 '안녕하세요'라는 글자를 넣기 위해서는 setData('안녕하세요'); 라고 해주면 된다.

## 5. useEffect에 대한 정의

useEffect는 두 가지의 경우로 쓰인다.

#1번째 경우 (Component가 실행될 때 처음 한 번만 실행할 것들을 넣는 경우)

```
useEffect(() => {  
    alert('환영합니다');  
}, []);
```

#2번째 경우 (특정한 값을 실시간으로 체크해야 하는 경우)

```
useEffect(() => {  
    // 감시하고 싶은 대상에게 변화가 생겼을 때 피드백  
}, [감시하고 싶은 대상]);  
  
useEffect(() => {  
    if (남자친구.location === '클럽') {  
        남자친구를 공격한다 // 피드백  
    } else if (남자친구.location === '회사') {  
        남자친구를 격려한다 // 피드백  
    }  
}, [남자친구]); // 감시할 대상은 []안에 넣어준다.
```

## 6. useRef의 사용

1) jQuery에서의 사용법

```
<input type="text" id="name"/>  
$('#name').value();
```

2) react에서의 사용법

```
const name = useRef();  
<input ref={name}/>
```

## 7. Send Component에서 Get Component에 id 값을 전달하는 방법

1) url로 전달할 때

```
가. <Route path="/get/:id"/> *Route에 params값을 정의  
나. const Send = ({ history }) => { ... *history 추가  
다. history.push('/get/2981'); *get으로 이동하면서 2981이라는 id값을 전달  
다. const Get = ( { match }) => { ... *get에서 match를 이용하여 받음
```

2) 자식 Component에 전달할 때

```
가. <Child id={id}/> *html 코드 내에서 직접적으로 props에 id를 추가해준다.  
나. const Child = ({ id }) => { ... *id값을 매개변수로 받아준다.
```