

React 기초 II

01 Props와 State



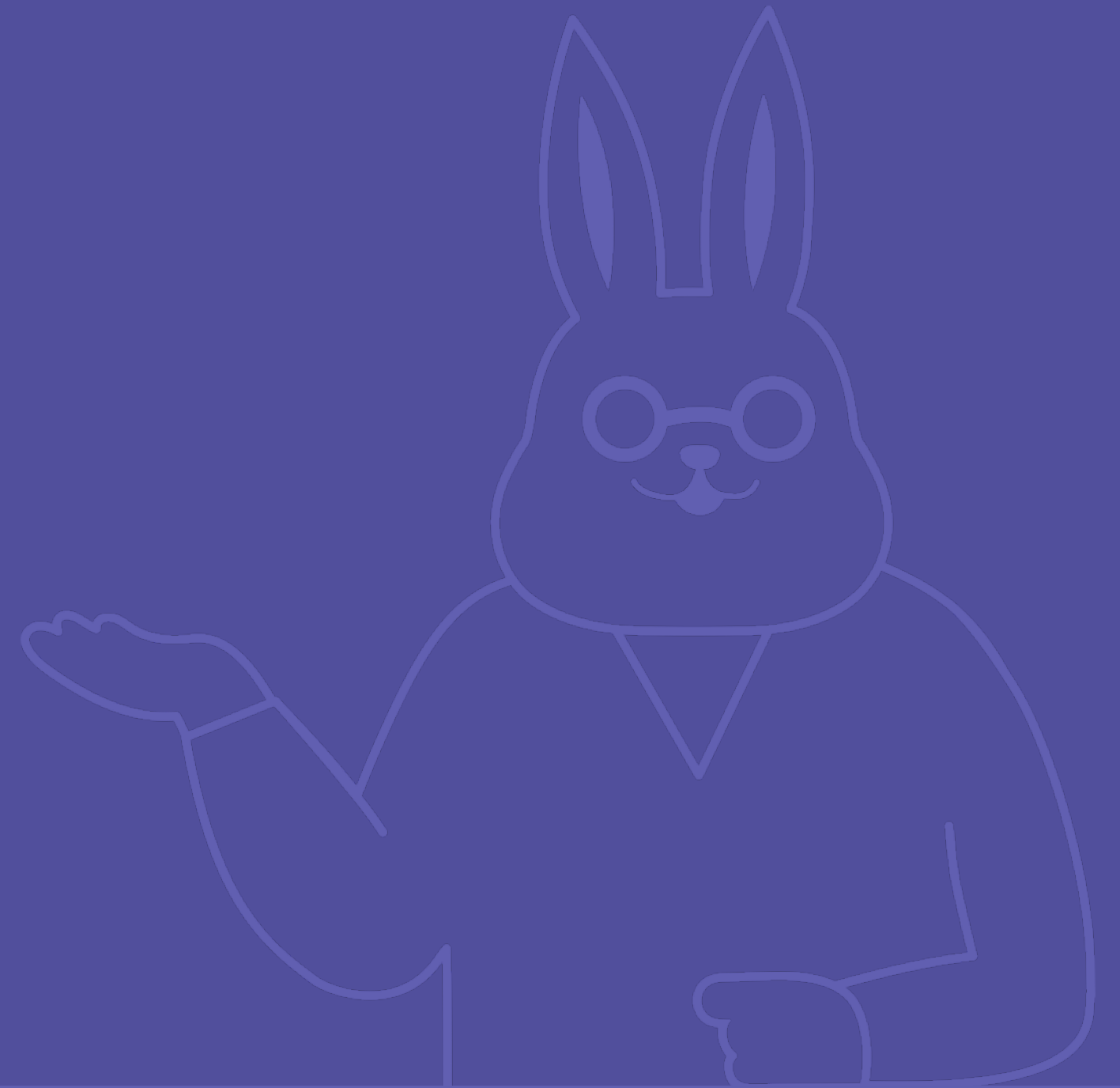
목차

01. Props

02. State

01

Props(Properties)



✓ Props란

컴포넌트 생성

```
const Welcome = (props) => {  
  return <h1>Hello, {props.name}</h1>;  
}
```



컴포넌트 사용

```
const App = () => {  
  return <div>  
    <Welcome name="수영" />  
    <Welcome name="민수" />  
    <Welcome name="영희" />  
  </div>;  
}
```

기본적으로 Component에 원하는 값을 넘겨줄 때 사용하며
넘겨줄 수 있는 값은 변수, 함수, 객체, 배열 등 JavaScript의 요소라면 제한이 없습니다.
주로 Component의 '재사용'을 위하여 사용합니다.

✓ Props는 읽기 전용입니다.

Don't

```
const Welcome = (props) => {  
  props.name = props.name + "님";  
  return <h1>Hello, {props.name}</h1>;  
}
```

Props의 값을 임의로 변경해서 사용하지 마세요.

Do

```
const Welcome = (props) => {  
  const username = props.name + "님";  
  return <h1>Hello, {username}</h1>;  
}
```

만약 Props 값을 변경해서 사용하고 싶다면,
새로운 변수를 생성하세요.

✓ DOM Element의 Attributes

- 기본적인 DOM Element(div, span 등)들의 Attribute는 camel case로 작성합니다.
예: tabIndex, className 등
- 그러나 'data-' 또는 'aria-' 로 시작하는 Attribute는 예외입니다.
예: data-type, aria-label 등
- HTML의 Attribute와 다른 이름을 가지는 Attribute가 있습니다.
class → className, for → htmlFor 등
- HTML의 Attribute와 다른 동작 방식을 가진 Attribute가 있습니다.
checked(defaultChecked), value(defaultValue), style 등
- React에서만 쓰이는 새로운 Attribute가 있습니다.
key, dangerouslySetInnerHTML 등

✓ HTML과 다른 방식의 React Attribute(`checked`, `value`)

코드

```
(<input type="checkbox" checked={false} />)
```

HTML에서 `checked` 또는 `value`는 해당 값이 ‘초기값’으로 쓰이지만 React 내에서는 **현재 값**을 의미합니다.

이 말은 즉, 위 예시처럼 `checked` 값이 **false**로 고정돼있는 경우에 사용자가 `checkbox`를 클릭하여도 값의 변화가 일어나지 않습니다.

만약 ‘초기값’의 의미로 `checked` 또는 `value`를 사용하고 싶다면 `defaultChecked`, `defaultValue` Attribute를 설정하세요.

✓ React에서만 쓰이는 새로운 Attribute(key)

코드

```
const Names = () => {  
  const names = [  
    {key: '1', value: '민수'},  
    {key: '2', value: '영희'},  
    {key: '3', value: '길동'},  
  ]  
  return (  
    <div>  
      {names.map((item) => (  
        <li key={item.key}>{item.value}</li>  
      ))}  
    </div>  
  )  
}
```

Key는 React가 어떤 항목을 변경, 추가 또는 삭제할지 식별하는 것을 돕습니다.

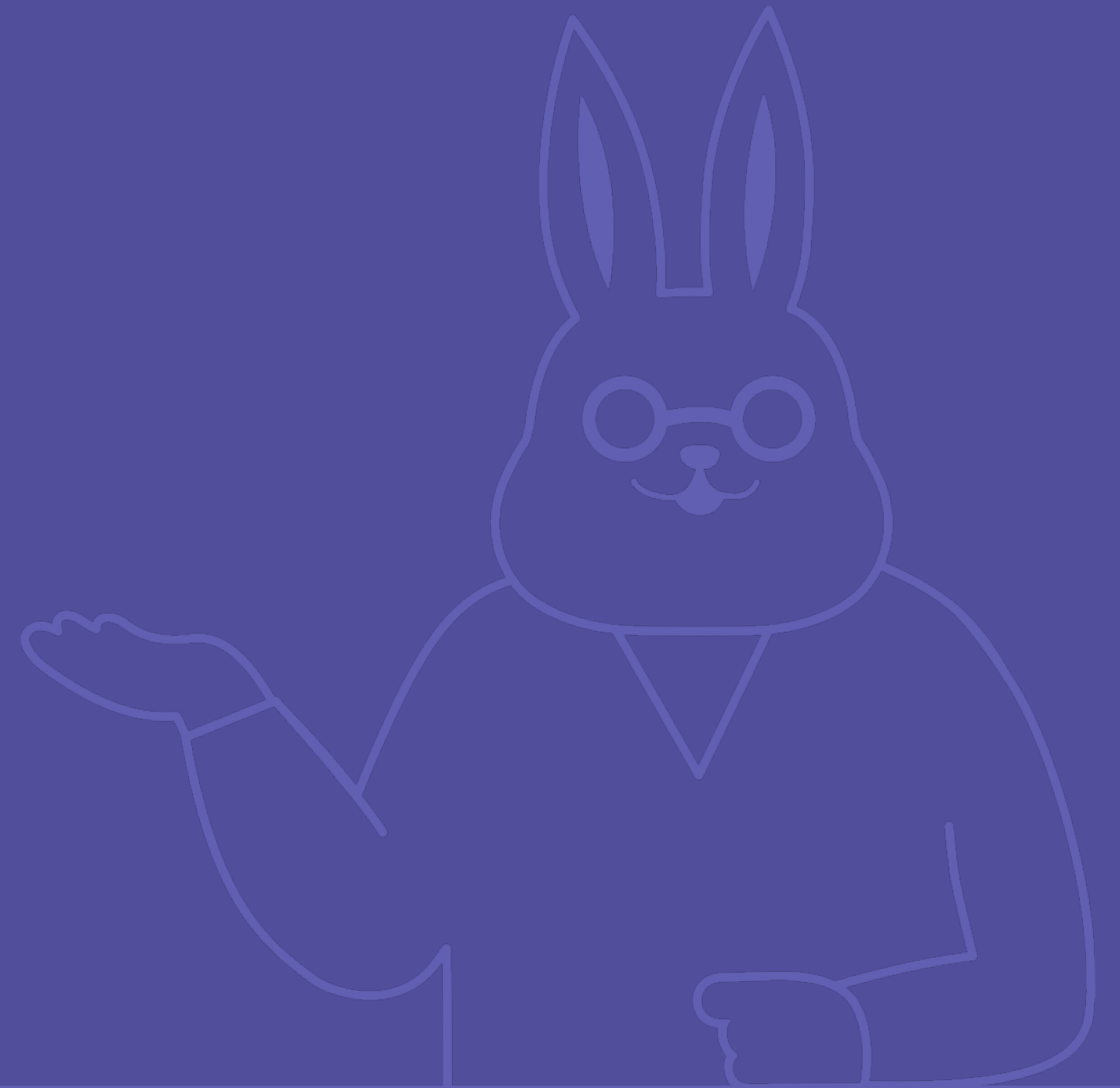
Key는 엘리먼트에 안정적인 고유성을 부여하기 위해 배열 내부의 Element에 지정해야 합니다.

Key는 배열 안에서 형제 사이에서 고유해야 하고 전체 범위에서 고유할 필요는 없습니다.

두 개의 다른 배열을 만들 때 동일한 key를 사용할 수 있습니다.

02

State



✓ State란

예시

```
import { useState } from 'react';

function Example() {
  const [count, setCount] = useState(0);
  return (
    <div>
      <p>버튼을 {count}번 눌렀습니다.</p>
      <button onClick={() => setCount(count + 1)}>
        클릭
      </button>
    </div>
  );
}
```

State는 Component 내에서 유동적으로 변할 수 있는 값을 저장합니다.

개발자가 의도한 동작에 의해 변할 수도 있고 사용자의 입력에 따라 새로운 값으로 변경될 수도 있습니다.

State 값이 변경되고 재렌더링이 필요한 경우에 React가 자동으로 계산하여 변경된 부분을 렌더링합니다.

✓ State 값을 직접 변경하지 마세요.

예시

```
import { useState } from 'react';

function Example() {
  let [count, setCount] = useState(0);
  return (
    <div>
      <p>버튼을 {count}번 눌렀습니다.</p>
      <button onClick={() => {count = count + 1;}}>
        클릭
      </button>
    </div>
  );
}
```

State 값을 직접 변경하게 되면 React가 Component를 다시 렌더링할 타이밍을 알아차리지 못합니다.

반드시 setState 함수를 이용해 값을 변경하세요.

setState 함수를 호출할 때 React에게 “다시 렌더링 해주세요~” 라는 명령이 내려집니다.

✓ State를 변경하는 두 가지 방법

예시

```
const [count, setCount] = useState(0);  
setCount(count + 1);
```

setState 내에 변경할 값을 넣기

예시

```
const [count, setCount] = useState(0);  
setCount((current) => {  
  return current + 1  
})
```

setState에 함수를 넣기

setState 함수에는 변경할 값을 직접 넣는 방법이 있고, 함수를 넣는 방법이 있습니다.

함수를 넣는 경우 함수가 **반환(return)하는 값으로 State가 변경**됩니다.

위 예시처럼 **현재 값을 기반으로 State를 변경하고자 하는 경우** 함수를 넣는 방법을 권장합니다.

✓ Object를 갖는 State를 만들 때 주의사항

예시

```
const [user, setUser] =  
useState({name: "민수", grade: 1})  
setUser((current) => {  
  current.grade = 2;  
  return current;  
})
```

예시와 같이 Object를 값으로 갖는 State도 만들 수 있습니다.

그러나 예시의 경우 React가 State의 변경을 감지하지 못합니다!

user object 안의 grade가 변경되었지만 user 자체는 변경되지 않았기 때문입니다.

✔ Object를 갖는 State를 만들 때 주의사항

```
{name: "민수",  
  grade: 1}
```



```
{name: "민수",  
  grade: 2}
```

user의 grade가 변경되었지만 user의 내용을 담은 object 자체가 변경된 것은 아닙니다.

```
{name: "민수",  
  grade: 1}
```



```
{name: "민수",  
  grade: 1}
```



```
{name: "민수",  
  grade: 2}
```

기존 user의 내용을 새로운 object에 담고 grade를 변경합니다.

✔ Object를 갖는 State를 만들 때 주의사항

이전 페이지의 설명을 코드로 작성해보면...

예시

```
const [user, setUser] =  
useState({name: "민수", grade: 1})  
setUser((current) => {  
  current.grade = 2;  
  return current;  
})
```

예시

```
const [user, setUser] =  
useState({name: '민수', grade: 1 })  
setUser((current) => {  
  const newUser = { ...current }  
  newUser.grade = 2  
  return newUser  
})
```

크레딧

/* elice */

코스 매니저

강윤수

콘텐츠 제작자

마로

강사

마로

감수자

장석준

디자이너

강혜정

연락처

TEL

070-4633-2015

WEB

<https://elice.io>

E-MAIL

contact@elice.io

