

05_props

리액트가 가진 특징

- (1) 재사용성 : 코드를 반복해서 쓰지 않기
 - (2) 관심사 분리 : 하나의 컴포넌트에 너무 많은 걸 담지 않기
- **react \Rightarrow HTML + js (CSS 옵션) \Rightarrow Component**

props

함수에 매개변수를 넣는 것처럼 컴포넌트에 전달하는 값(Properties)

```
//<App />
const num = 10;

<Component num ={num}>

    <p>{props.num}</p>
```

- 성공 토스트 2개, 경고 2개, 에러 2개 (값을 각각 다르게 해서 props로 넣어보기)

(1) app.js

```
import './App.css';
import Toast from './component/Toast';

function App() {
  const messageArray = [
    // 속성이 2개라면, 값을 바로 넣는것보다 다음과 같이 오브젝트를 만들어서 넣어준다.
    {
      title: 'success',
      text: 'Right On!'
    },
    {
      title: 'warning',
      text: 'Right On!'
    },
    {
      title: 'error',
      text: 'Right On!'
    }
  ]

  return (
    <div className='wrapper'>
      <Toast title={messageArray[0].title} text={messageArray[0].text} />
      <Toast title={messageArray[1].title} text={messageArray[1].text} />
      <Toast title={messageArray[2].title} text={messageArray[2].text} />
    </div>
  )
}
```

```

    <Toast title={messageArray[3].title} text={messageArray[3].text} />
    <Toast title={messageArray[4].title} text={messageArray[4].text} />
    <Toast title={messageArray[5].title} text={messageArray[5].text} />
  </div>
);
}

export default App;

```

(2) toast.js

```

import React from 'react';
import './Toast.css';

const Toast = (props) => {
  return (
    <div className={`toast toast-${props.title}`}>
      <main className='toast__message'>
        <header className='toast__message-title'>{props.title}</header>
        <p className='toast__message-text'>{props.text}</p>
      </main>
      <button className='toast__button'>Dismiss</button>
    </div>
  );
};

export default Toast;

```

==> 이부분을 더 간단하게 만드는 방법 : map을 사용하여 반복하여 나올 수 있도록 만들기

```

<Toast title={messageArray[0].title} text={messageArray[0].text} />
<Toast title={messageArray[1].title} text={messageArray[1].text} />
<Toast title={messageArray[2].title} text={messageArray[2].text} />
<Toast title={messageArray[3].title} text={messageArray[3].text} />
<Toast title={messageArray[4].title} text={messageArray[4].text} />
<Toast title={messageArray[5].title} text={messageArray[5].text} />

```

[1] 은 [2] 와 같은 코드

```

return (
  <div className='wrapper'>
    {messageArray.map((messgae) => {
      <Toast title={messgae.title} text={messageArray.text} />
    })}

  </div>
);

```

```

return (
  <div className='wrapper'>
    {messageArray.map((messgae, index) => {
      <Toast title={messgaeArray[index].title} text={messageArray.text} />
    })}

  </div>
);

```

[3] 더 간결한 코드

```

return (
  <div className="wrapper">
    {messageArray.map((message) => (
      <Toast title={messgaeArray[index].title} text={messageArray.text} />
    ))}
  </div>
);

```

중괄호와 return 을 생략하여 3줄로 간략하게 코드 작성 가능

- props는 컴포넌트의 자바스크립트 기능을 도입한 개념
- 함수의 매개변수를 넣는 것처럼 컴포넌트에 props를 넣어줄 수 있다.