

이벤트 처리

2023.06.16.

- 01. 이벤트 state props와 render 함수
- 02. 컴포넌트 이벤트 만들기
- 03. 함수형 컴포넌트에서 이벤트 처리

01. 이벤트 state props와 render 함수

2023.06.16.

1. 이벤트 설치

1.1 이벤트명 작성규칙

- react에서 사용되는 이벤트는 카멜 표기법으로 표시하도록 약속되어 있다.

ex> html에서 마우스 클릭 이벤트명은 onclick으로 표시하나 react에서는 onClick으로 표시한다.

구문

```
<태그명 이벤트명 = {function (인수, ...) {
  실행 내용;
  ...
}} />
```

예시

```
<a href="/" onClick={function() {
  alert("hi");
}}>{this.state.subject.title}</a>
```

1.2 호출객체 제어

- JavaScript에서 이벤트 함수의 this는 이벤트를 발생시키는 객체로 정의될 수 있으나 React의 이벤트 함수 this는 undefined이다.
- React는 이벤트가 호출될 경우 이벤트를 일으키는 개체와 이벤트 행위를 포함하는 event 객체 (SyntheticBaseEvent)가 첫 번째 매개변수로 추가될 수 있다. 따라서 필요시 이 객체를 활용하여 이벤트 호출 객체를 제어할 수 있다.
- 매개변수로 전달된 SyntheticBaseEvent 객체의 target 속성에서 이벤트 수신 객체를 얻을 수 있다.

구문

```
<태그명 이벤트명 = {function (이벤트객체인수, ...) {
  이벤트 객체를 제어하는 구문;
  ...
}} />
```

예시

```
<a href="/"onClick={function(e) {
  alert("hi");
  console.log(e.target);
  debugger;
}}>{this.state.subject.title}</a>
```

1.3 preventDefault() : 객체의 기본 동작을 막아주는 메서드

Ex> a의 href 위치 이동 방지, form의 action 위치로 이동 및 값 전송 방지

예시

```
<a href="/"onClick={function(e) {
  console.log(e);
  e.preventDefault();
}}>{this.state.subject.title}</a>
```

2. 이벤트에서 state 변경하기

2.1 bind() 사용

- React의 이벤트 함수 this는 undefined이다. 컴포넌트의 state는 this.state로 접근하니 사용할 수 없다.
- 해결방법 : 이벤트 함수 뒤에 bind(this)를 붙인다.

예시

```
<a href="/"onClick={function(e) {
  console.log(e);
  e.preventDefault();
  this.state.mode = "welcome";
}.bind(this)}>{this.state.subject.title}</a>
```

2.2 setState()

- 이벤트 함수 내에서 state의 값을 위 예시 처럼 변경하여도 react는 알아차릴 수 없다. 함수 내부에서 state를 변경하고자 할 경우에는 setState() 함수를 사용해야 한다.

구문

```
<태그명 이벤트명 = {function (이벤트객체인수, ...) {
  ...
  this.setState({스테이트명: 값});
}.bind(this)} />
```

예시

```
<a href="/"onClick={function(e) {
  e.preventDefault();
  this.setState({mode:"welcome"});
}.bind(this)}>{this.state.subject.title}</a>
```

02. 컴포넌트 이벤트 만들기

2023.06.19.

1. 컴포넌트 이벤트 만들기

1.1 정의 : 컴포넌트 사용(호출)시 props로 함수를 정의
구문

```
<컴포넌트 이벤트프롭스명={function(){
  실행 내용
}.bind(this)} />
```

예시

```
<Subject
  title={this.state.subject.title}
  sub={this.state.subject.sub}
  onChangePage={function(){
    this.setState({mode:"welcome"});
  }.bind(this)}
/>
```

2. 이벤트 객체(SyntheticBaseEvent)의 dataset 속성에 변수와 값을 설정하는 방법

2.1 설정(props로 설정)

구문

```
<태그 data-변수={값}>
```

예시

```
<a data-id={3}></a>
```

Ex> <a data-id={3}>

2.2 이벤트 객체를 통한 호출

구문

```
<태그 이벤트props = {function(e){
  ...
  e.target.dataset.변수
  ...
}}
>
```

예시

```
<a
  href="/"
  data-id={contents.id}
  onClick={function(e) {
    e.preventDefault();
    this.props.onChangePage(e.target.dataset.id);
  }.bind(this)}
>{contents.title}</a>
```

3. .bind()의 용법

- 이벤트 함수의 끝에 `.bind()`를 추가하면 매개변수를 이벤트 함수 내로 전달할 수 있다.
- 단, `bind`의 첫번째 매개변수는 `this`로 하고, 두 번째 이후는 사용자가 정의한다.
- 그리고 두 번째 이후의 매개변수는 이벤트 함수의 첫 번째 매개변수 부터 적용된다.
- 만약, 이벤트 함수에 기존 전달되는 매개변수가 있다면 `bind()`로 추가된 매개변수 이후로 기술해야 한다.

구문

```
<태그 이벤트프롭스={function(first, second, e)
  {
    ...
  }.bind(this, first, second)}
>
```

예시

```
<a
  href="/"
  data-id = {contents.id}
  onClick={function(id, e) {
    e.preventDefault();
    this.props.onChangePage(id);
  }.bind(this, contents.id)}
>{contents.title}</a>
```

※ 메뉴 클릭에 따른 서식 설정하기(class 추가/제거, javascript)

```

import { Component } from "react";

class TOC extends Component {
  setActive(id) {
    let as = document.getElementsByClassName("content");
    for(let i=0; i<as.length; i++){
      as[i].classList.remove("active");
    }
    as[id-1].classList.add("active");
  }

  render() {
    console.log("TOC render");

    let data = this.props.data;
    let lists = [];

    for(let contents of data) {
      lists.push(
        <li key={contents.id}>
          <a
            href="/"
            data-id = {contents.id}
            onClick={function(id, e) {
              e.preventDefault();
              this.setActive(id);
              this.props.onChangePage(id);
            }.bind(this, contents.id)}
          >{contents.title}</a>
        </li>
      );
    }

    return(
      <nav>
        <ul>
          {lists}
        </ul>
      </nav>
    );
  }
}

export default TOC;

```

03. 함수형 컴포넌트에서 이벤트 처리

2023.06.19.

1. 컴포넌트 이벤트 만들기

1.1 정의 : 컴포넌트 사용(호출)시 props로 함수를 정의
구문

```
<컴포넌트 이벤트프롭스명={function(){
  실행 내용
}} />
```

예시

```
<Subject
  title={subject.title}
  sub={subject.sub}
  onChangePage={function(){
    alert("hihi");
  }}
/>
```

1.2 호출 : 컴포넌트 정의 파일에서 매개변수로 전달 받은 다음 사용할 장소 지정
구문

```
function 컴포넌트명({프롭스명, ..., 이벤트프롭스명}) {
  return (
    <태그 이벤트명={function() {
      이벤트프롭스명();
    }}
  );
}
```

예시

```
function Subject({title, sub, onChangePage = function(f){return f}}) {
  return(
    <header>
      <h1>
        <a
          href="/"
          onClick={function(e){
            onChangePage();
          }}
        >{title}</a>
      </h1>
      {sub}
    </header>
  );
}
```

1.2.1 함수의 매개변수의 초기화

- 함수는 매개변수 값이 전달되지 않아 발생될 오류를 대비하여 초기화가 가능하다.

구문

```
function 컴포넌트명({
  프롭스명 = 0,
  프롭스명 = [],
  프롭스명 = {},
  이벤트프롭스명 = function(){}
}) {
  return (
    ...
  );
}
```

- 이벤트프롭스의 초기화는 화살표 함수로도 가능함

```
function 컴포넌트명({프롭스명 = 0, ... , 이벤트프롭스명 = f => f}) {
  return (
    ...
  );
}
```

예시

```
function Subject({title, sub, onChangePage = f =>f}) {
  return(
    <header>
      <h1>
        <a
          href="/"
          onClick={function(e){
            onChangePage();
          }}
        >{title}</a>
      </h1>
      {sub}
    </header>
  );
}
```