컴포넌트?

특정 코드 뭉치를 다른 부분에 이식하거나 재사용하기 위해 사용하는 코드 블록 단위

파일 단위로 작성한 후 필요한 위치에서 임포트해 사용할 수 있다.

component 사용원리?

첫 번째 줄에서 import한 {component}를 상속받아 R003 클래스(새로만든js파일)에서 사용

return된 html코드를 render()함수를 사용해 화면에 표시

App.js 에서 import해서 사용가능

생명주기?

component의 생성, 변경, 소멸 과정

생성 : constructor() getDerivedStateFormProps() render() componentDidMount()

render()?

render()는 return되는 html형식의 코드를 화면에 그려주는 함수

화면 내용이 변경돼야 할 시점에 자동으로 호출

constructor(props)

생명주기 함수 중 가장 먼저 실행되며, 처음 한번만 호출

component 내부에서 사용되는 변수(state)를 선언하고

부모 객체에서 전달받은 변수(props)를 초기화할 때 사용한다.

super()함수는 가장 위에 호출해야 한다

getDerivedStateFromProps(props, state)

constructor() 다음으로 실행된다.

component가 새로운 props를 받게 됐을 때 state를 변경해준다.

App.js에서 전달한 prop\_value라는 변수를 props.prop\_value로 접근해 값을 가져올 수 있다.

componentDidMount()

작성한 함수들 중 가장 마지막으로 실행된다.

render()가 return 되는 html 형식의 코드를 화면에 그려진 후 실행된다.

화면이 그려진 후에 실행돼야하는 이벤트 처리, 초기화 등 가장 많이 활용되는 함수다.

setState()

변수의 선언과 초기화를 동시에 실행한다.

shouldComponentUpdate(props,state)

setState()로 인해서 state의 변경이 발생했기 떄문에 ‘변경’단계의 생명주기 함수 shouldComponentUpdate()가 실행된다.

Boolean 유형의 데이터를 반환하는데 return이 true일 경우 render()를 한 번 더 호출한다.

NOTE

함수의 반환값에 따라 render() 재실행 가능한다는 점을 이용

Props나 state변수가 변경될 때 화면을 다시 그리며 제어할 수 있다.

* 직접 해보니 componentDidMount까지 왔다가 constructor 건너뛰고 getDerived 나오고 render 나옴 새로운 getDerived 나오고부터 this.state가 변경됨

ES는 표준화된 스크립트 언어

ES 뒤에 붙은 숫자는 버전

파이썬에서의 f-string == js에서의 백틱+${변수}

f’hello {name}’ == `hello ${name}`

아래는 파이썬과 동일함

startswith

endswith

파이썬의object in list == includes

Object not in list