### 참조: <https://react.codepot.kr/docs/week01/doc5>

## JSX에서 표현식 사용하기

## JSX는 중괄호 안에 유효한 모든 javscript 표현식을 넣을 수 있습니다.

**표현식(expressions)**

값을 산출해내는 코드의 조각

| // 숫자 표현식  10;  10 + 20;  // 문자 표현식  "hello";  "hello" + "world";  // 논리 표현식  30 > 20;  10 < 20;  // 할당 연산자를 통한 표현식  i = 10;  total = 0;  fruits[1] = "avocado";  // 함수 호출 표현식  sayHello();  ()={ } |
| --- |

### 문장 (statements)

문장은 특정한 행동을 하겠다는 설명 (JSX에서는 사용불가)

| // 변수 선언  var sum;  var average;  // if, else 문  if (expression)  statement 1  else  statement 2  // for 문  for(let i = 0; i < 10; i++){  console.log(i)  } |
| --- |

### JSX에서 변수 넣기[#](https://react.codepot.kr/docs/week01/doc5#jsx%EC%97%90%EC%84%9C-%EB%B3%80%EC%88%98-%EB%84%A3%EA%B8%B0)

먼저 변수를 JSX 내부에 넣어보도록 하겠습니다.

| const name = "code pot";  const App = () => {  return <div>{name}</div>;  }; |
| --- |

위 코드처럼 선언한 변수를 중괄호로 감싸 JSX에서 사용할 수 있습니다.

### JSX에서 함수 사용하기

| const user = {  firstName: 'toy',  lastName: 'crane'  };  const formatName(user) {  return user.firstName + ' ' + user.lastName;  }  const App = () => {  return <div>{formatName(user)}</div>  } |
| --- |

### 

| function formatName(user) {  return user.firstName + ' ' + user.lastName; }const user = {  firstName: 'hong',  lastName: 'gildong' };const element = (  <h1>  Hello, {formatName(user)}!  </h1> );ReactDOM.render(  element,  document.getElementById('root') ); |
| --- |

### 

### JSX에서 if문처럼 사용하기

위에서 이야기 했던 것처럼 JSX 내 에서는 **표현식**만 사용할 수 있습니다.   
그래서 if문과 같은 효과를 내려면 **삼항 연산자**를 사용해야 합니다.

| const isLogin = true;  const App = () => {  // isLogin이 True이면 첫번째 요소인 <div>로그인 완료</div>의 표현식을 실행  // isLogin이 False이면 두번째 요소인 <div>로그인 해주세요</div> 표현식을 실행  return {isLogin ? <div>로그인 완료</div> : <div>로그인 해주세요</div>  } |
| --- |

### 

### JSX에서 for문처럼 사용하기

if문과 마찬가지로 for문도 문장이기 때문에 사용할 수 없습니다.   
그래서 주로 map을 사용합니다.

| const fruits = ["사과", "감", "배", "딸기"];  const App = () => {  // fruits를 한바퀴 씩 돌면서 fruits가 끝날 때까지  // map 안에 있는 함수((fruit) => <div>{fruit}</div>)를 실행함  // 함수 안에 있는 fruit은 fruits 리스트의 각 요소입니다.  return fruits.map((fruit) => <div>{fruit}</div>);  }; |
| --- |

map() 메서드는 배열 내의 모든 요소 각각에 대하여 주어진 함수를 호출한 결과를 모아 새로운 배열을 반환합니다.

| const array1 = [1, 4, 9, 16];  // pass a function to map  const map1 = array1.map(x => x \* 2);  console.log(map1);  // expected output: Array [2, 8, 18, 32] |
| --- |