

5. JSX를 사용하는 리액트 (1)

Prof. Seunghyun Park (sp@hansung.ac.kr)

Division of Computer Engineering

학습 목표: 5장. JSX를 사용하는 리액트

- JSX
- Babel
- JSX 활용 예제: Recipe
- 웹팩
- 프로젝트 구성하기

- JavaScript의 확장 문법 (JavaScript XML)

- HTML 코드를 작성하는 방법과 유사
- 직관적이고 가독성이 좋음

- 객체로 다루며, 표현식 포함 가능

- 변수 할당, 매개변수 전달, 함수 반환 등 일반 JavaScript 표현식으로 활용 가능

표현식 포함 가능

```
const element = <tag>{expression}</tag>;
```

object

객체로 활용

리액트 엘리먼트

변환을 위한 변환기 필요

JSX

```
React.createElement(IngredientsList, { id:"1", ... }, children);
```

```
<IngredientsList id="1" ... >
```

```
<Ingredient />
```

```
<Ingredient />
```

```
</IngredientsList>
```

다른 컴포넌트를 포함

JSX 활용 예제 (1)

```
<!-- Root element -->
<div id="root"></div>

/* ch05-01-jsx.html */
function formatName(user) {
  return `${user.firstName} ${user.lastName}`
}

const user = {
  firstName: 'Harper',
  lastName: 'Perez'
};
```

```
const element = (
  <h1>
    Hello, {formatName(user)}!
  </h1>
);
```

JSX

```
ReactDOM.render(
  element,
  document.getElementById('root')
);
```

```
<h1>
  {formatName(user)}
  === `${user.firstName} ${user.lastName}`
  === "Harper Perez"
</h1>
```

<h1> 엘리먼트 내부에서 JS 표현식 평가

Hello, Harper Perez!

#text 148.86 × 64

```
<!DOCTYPE html>
<html>
  <head>...</head>
  <body>
    <!-- Root element -->
    <div id="root">
      <h1>
        "Hello, " == $0
        "Harper Perez"
        "!"
      </h1>
    </div>
    <!-- React Library, React DOM -->
```

JSX 활용 예제 (2)

```
/* ch05-02-jsx.html from ch04-03-01-3-func.html */
```

```
function IngredientsList(props) {  
  return React.createElement(  
    "ul",  
    { className: "ingredients" },  
    props.items.map( (ingrd, i) =>  
      React.createElement("li", {key: i}, ingrd) )  
  );  
}
```

```
/* ch05-03-jsx.html from ch04-03-01-3-func.html */
```

```
function IngredientsList(props) {  
  return (  
    <ul class="ingredients">  
      { props.items.map(  
        (ingrd, i) => <li key={i}>{ingrd}</li> )}  
    </ul>  
  );  
}
```

- 무염 버터 1 컵
- 크런치 땅콩 버터 1 컵
- 흑설탕 1 컵
- 백설탕 1 컵
- 달걀 2 개
- 일반 밀가루 2.5 컵
- 베이킹 소다 1 티스푼
- 소금 0.5 티스푼

ul.ingredients 285.33 × 170.67

```
<!-- Root element -->  
▼ <div id="root">  
  ▼ <ul class="ingredients">  
    ▶ <li>...</li>  
    ▶ <li>...</li>  
    ▶ <li>...</li>  
    ▶ <li>...</li>  
    ▶ <li>...</li>  
    ▶ <li>...</li>  
    ▶ <li>...</li>  
    ▶ <li>...</li>
```

babel

- 코드 변환의 필요성 → babel 활용

- JavaScript 최신 문법의 활용과 호환성
- JSX 코드를 리액트 코드로 변환

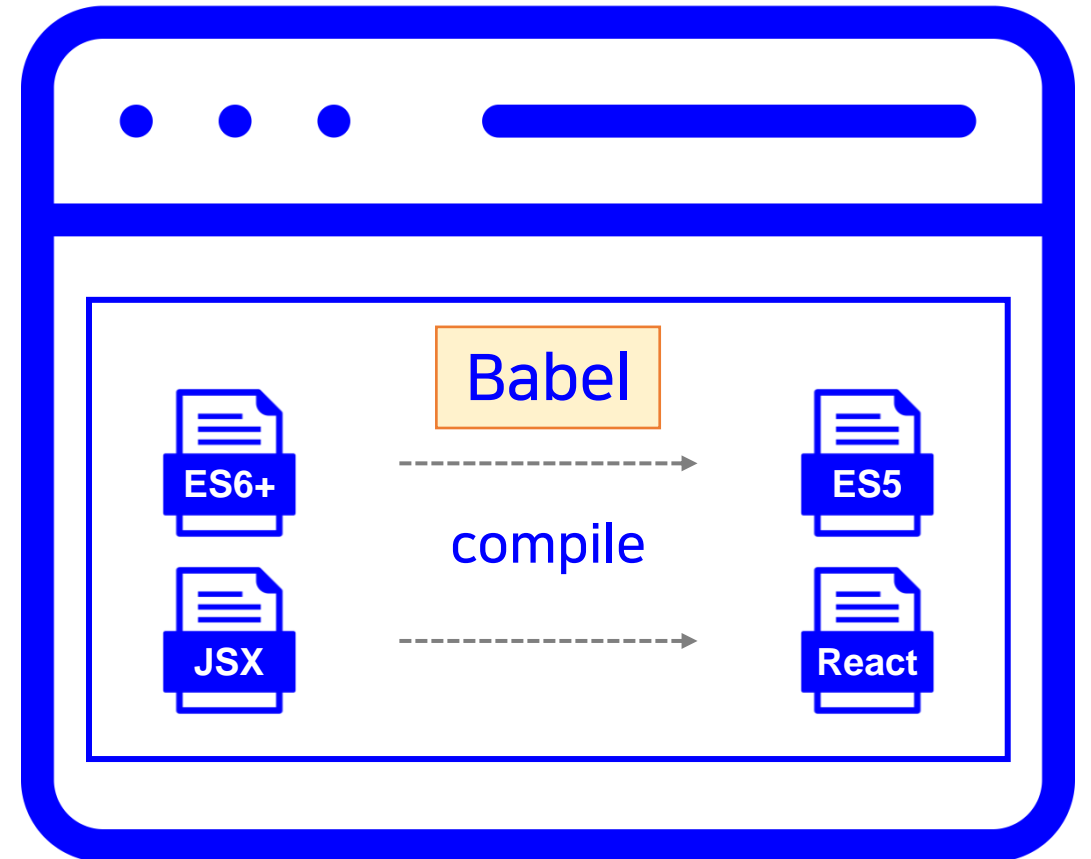
- babel 사용법

- 스크립트 타입 선언

```
<script type="text/babel">
```

- CDN 링크를 HTML에 포함

```
<script src="https://unpkg.com/babel-standalone@6.15.0/babel.min.js"></script>
```



```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Page setup</title>
</head>
<body>

<!-- Root element -->
<div id="root"></div>

<!-- React Library & React DOM-->
<script src="https://unpkg.com/react@16/umd/react.development.js"></script>
<script src="https://unpkg.com/react-dom@16/umd/react-dom.development.js"></script>
<script src="https://unpkg.com/babel-standalone@6.15.0/babel.min.js"></script>

<script type="text/babel">

/* ch05-04-babel.html from ch04-01-01-page-setup.html */
// JavaScript 파일의 링크를 script 태그에 포함
// JSX 코드

</script>

</body>
</html>
```

```
<script type="text/babel">

const element = (
  <h1>
    Hello, {formatName(user)}!
  </h1>
);
```

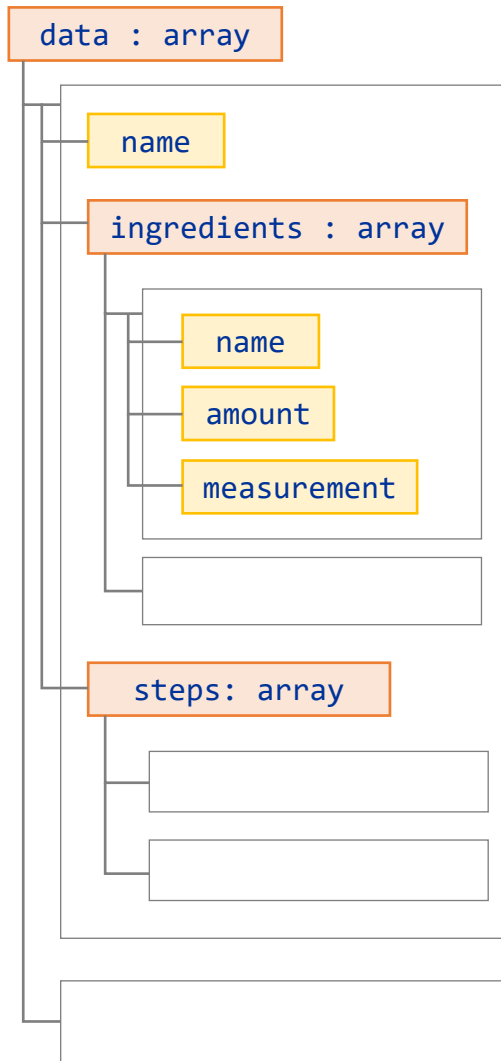
Babel

```
<script type="module">

const element = React.createElement(
  "h1",
  null,
  "Hello, ",
  formatName(user),
  "!"
);
```

JSX 활용 예제: Recipes

```
const data = [
  {
    "name": "Baked Salmon",
    "ingredients": [
      { "name": "연어", "amount": 500, ... },
      ...,
    ],
    "steps": [
      "오븐을 180도로 예열한다.",
      ...,
    ]
  },
  {
    "name": "Fish Tacos",
    "ingredients": [
      { "name": "흰살생선", "amount": 500, ... },
      ...,
    ],
    "steps": [
      "생선을 그릴에 익힌다.",
      ...,
    ]
  }
]
```



section 434.4 × 386.54

Baked Salmon

- 연어
- 잣
- 버터 상추
- 옐로 스쿼시
- 올리브 오일
- 마늘

recipe

- 오븐을 180도로 예열한다.
- 유리 베이킹 그릇에 올리브 오일을 두른다.
- 연어, 마늘, 잣을 그릇에 담는다.
- 오븐에서 15분간 익힌다.
- 옐로 스쿼시를 추가하고 다시 30분간 오븐에서 익힌다.
- 오븐에서 그릇을 꺼내서 15분간 식힌다음에 들어서 내놓는다.

Fish Tacos

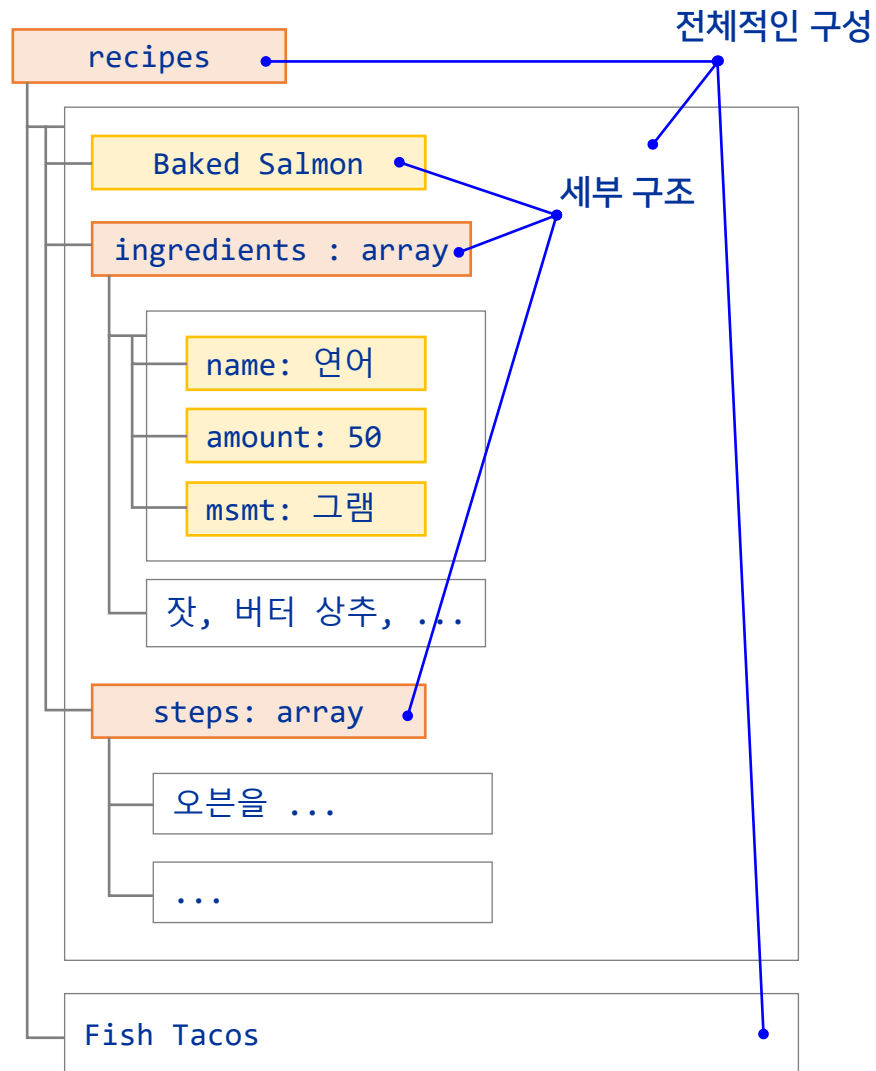
- 흰살생선
- 치즈
- 아이스버그 상추
- 토마토
- 토띠야

recipe

- 생선을 그릴에 익힌다.
- 토띠야 3장 위에 생선을 얹는다.
- 토띠야에 얹은 생선 위에 상추, 토마토, 치즈

```
<!DOCTYPE html>
<html>
  <head>...</head>
  <body>
    <!-- Root element -->
    <div id="root">
      <section>
        <header>
          <h1>recipes</h1>
        </header>
        <div class="recipes">
          <section == $0
            <h1>Baked Salmon</h1>
            <ul>
              <li>...</li>
              <li>...</li>
              <li>...</li>
              <li>...</li>
              <li>...</li>
              <li>...</li>
            </ul>
            <h3>recipe</h3>
            <ul>...</ul>
          </section>
          <section>
            <h1>Fish Tacos</h1>
            <ul>...</ul>
            <h3>recipe</h3>
            <ul>...</ul>
          </section>
        </div>
      </section>
    </div>
  </body>
</html>
<!-- React Library & React DOM-->
```


JSX 활용 예제: Recipes



```
<!-- Root element -->
<div id="root"></div>

<!-- React Library, React DOM & babel-->
<script src=react.js, react-dom.js, babel.js></script>

<script type="text/babel">

/* ch05-05-1-recipe.html */

// data
const data = [ ... ];

// component: describing a recipe
const Recipe = function( ... ){ ... };

// component: describing a menu
const Menu = function(...){ ... };

// Menu를 현재의 DOM 안에 렌더링
ReactDOM.render(
  <Menu recipes={data} title="recipes" />,
  document.getElementById('root')
)

</script>
```

세부 구조

전체적인 구성

JSX 활용 예제: Recipes

```
<!-- Root element -->
<div id="root"></div>

<script type="text/babel">
/* ch05-05-1-recipe.html */

// data
const data = [ ... ];

// component: Menu
const Menu = function;

// component: Recipe
const Recipe = function;

// rendering
ReactDOM.render();
```

```
// component: describing a menu
const Menu = function(props){
  return (
    <section>
      <header>
        <h1>{props.title}</h1>
      </header>
      <div className="recipes">
        { props.recipes.map( (recipe, i) =>
          (<Recipe key={i}
            name={recipe.name}
            ingredients={recipe.ingredients}
            steps={recipe.steps} />) ) }
      </div>
    </section>
  );
}
```

```
// component: describing a recipe
const Recipe = function({name, ingredients, steps} ){
  return (
    <section>
      <h1>{name}</h1>
      <ul>
        {ingredients.map( (ingrd, i) =>
          (<li key={i}>{ingrd.name}</li>) )}
      </ul>
      <section>
        <h3>recipe</h3>
        <ul>
          {steps.map( (step, i) =>
            (<p key={i}>{step}</p>) )}
        </ul>
      </section>
    </section>
  );
}
```

```
// Menu를 현재의 DOM 안에 렌더링
ReactDOM.render(
  <Menu recipes={data} title="recipes" />,
  document.getElementById('root')
)
```

```
<div id="root">
  <Menu />
</div>
```

```
<section>
  <header>
    <h1></h1>
  </header>
  <div>
    <Recipe />
  </div>
</section>
```

```
<section>
  <h1></h1>
  <ul> <li><li>...<li> </ul>
  <section>
    <h3></h3>
    <ul> <p><p>...<p> </ul>
  </section>
</section>
```