

## 소프의 처음 만나 리액트



## 제 5장 컴포넌트와 PROPS



## Q & A

- 언제라도 질문하세요  
모르면 외우면 되지만 코딩 안되면 꼭 질문
  1. 대면수업에서 질문
  2. 한성e-class 질의응답게시판
- 강사의 1번 선생님은 여러분의 질문





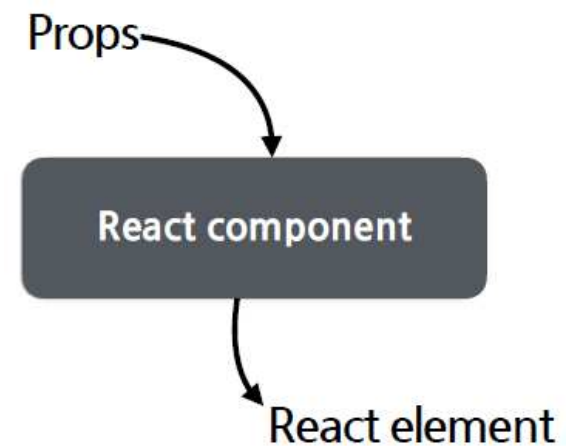
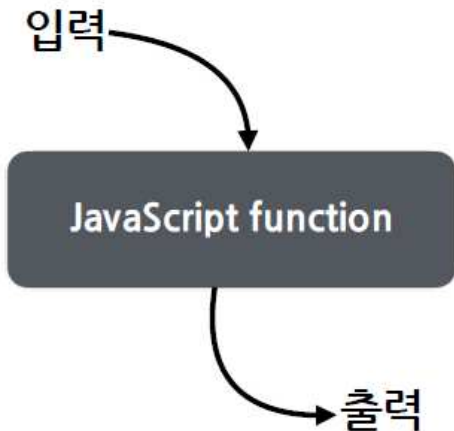
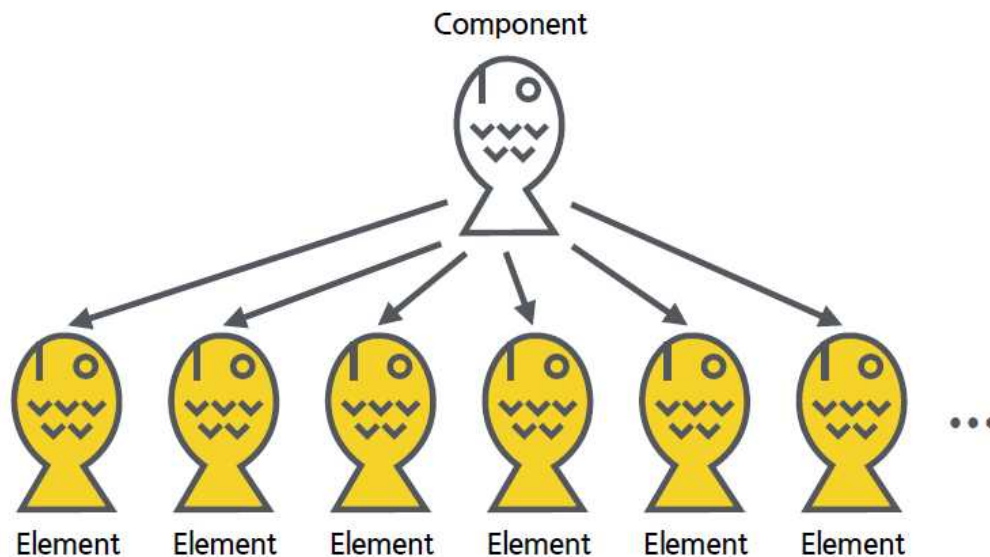
# 목차

1. 컴포넌트에 대해 알아보기
2. **Props**에 대해 알아보기
3. 컴포넌트 만들기
4. 컴포넌트 합성
5. 컴포넌트 추출
6. 실습: 댓글 컴포넌트 만들기
7. 마치며



# 1. 컴포넌트에 대해 알아보기

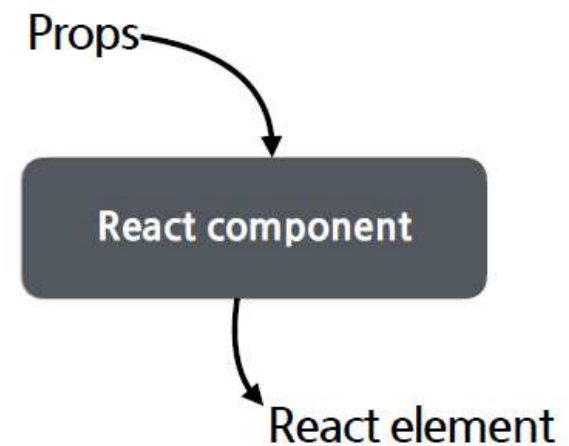
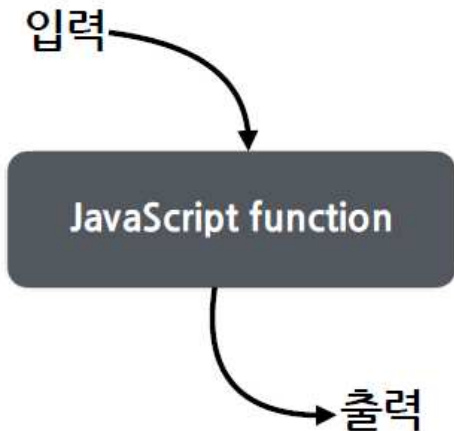
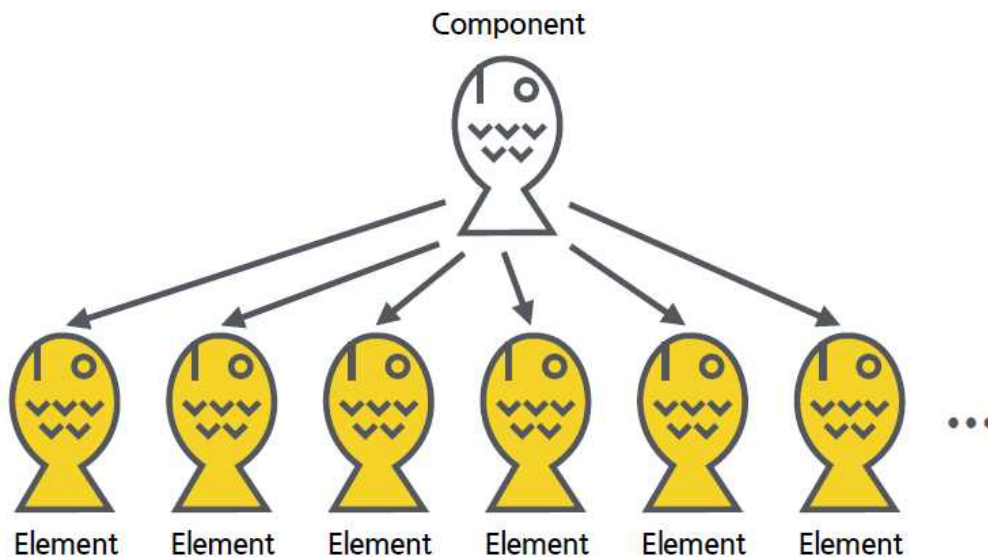
- React는 컴포넌트 기반의 구조
- 컴포넌트는 Props(속성) 입력을 받아 리액트 엘리먼트 생성
- 컴포넌트는 붕어빵틀, 엘리먼트는 붕어빵





## 2. Props에 대해 알아보기

- React는 컴포넌트 기반의 구조
- 컴포넌트는 Props(속성) 입력을 받아 리액트 엘리먼트 생성
- 컴포넌트는 붕어빵틀, 엘리먼트는 붕어빵





## 2. Props 자료형

```
React.createElement(  
  type,  
  [props], // 자료형은 객체(교재 36쪽) 또는 null 또는 blank  
  [...children] // blank 또는 null 또는 여러 개의 element 또는 string  
)
```

= props가 객체인 경우

- 코드위치: c:\ws\_react\CDN\chapter\_05\_props\Props1.js

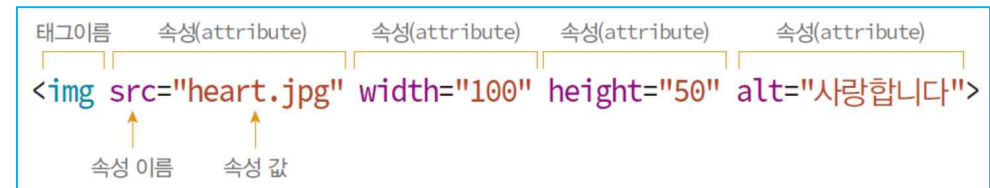
```
React.createElement(  
  "div",  
  {  
    name: "소플",  
    introduction: "안녕하세요, 소플입니다.",  
    viewcount: 1500  
  },  
  null  
)
```

- Props가 render()후 HTML 태그의 attribute가 됨

= props가 null인 경우

- 코드위치: c:\ws\_react\CDN\chapter\_05\_props\Props2.js

```
React.createElement("h1", null, "안녕, 리액트!")
```



```
Elements Console Sources Network Performance Memory Applicatio  
tml>  
<head> </head>  
<body>  
▼ <div id="root"> == $0  
  <div name="소플" introduction="안녕하세요, 소플입니다." viewcount="1500"></div>  
  ...
```

```
Elements Console  
<html>  
  <head> </head>  
  ▼ <body>  
    ▼ <div id="root"> == $0  
      <h1>안녕, 리액트!</h1>
```



## 2. Props 자료형

```
React.createElement(  
  type,  
  [props], // 자료형은 객체(교재 36P) 또는 null 또는 blank  
  [...children] // blank 또는 null 또는 여러 개의 element 또는 string  
)
```

= props가 blank인 경우

- 코드위치: c:\ws\_react\CDN\chapter\_03\_reactJS\index.js  
root.render(React.createElement(Library));



## 2. Children 자료형

React.createElement(

type,

[props], // 자료형은 객체(교재 36P) 또는 null 또는 blank

[...children] // blank 또는 null 또는 여러 개의 element 또는 string

)

= children이 여러 개의 element 경우

- 코드위 치: c:\ws\_react\CDN\chapter\_05\_children\Child3.js
- Children이 render()후 child HTML 태그가 됨

= children이 여러 개의 string 인 경우

- 코드위 치: c:\ws\_react\CDN\chapter\_05\_children\Child4.js

= children이 1개의 string 인 경우

- 코드위 치: c:\ws\_react\CDN\chapter\_05\_children\Child5.js

```
Elements Console Sources
<html>
  <head> </head>
  <body>
    <div id="root"> == $0
      <div>현재 시간: 오후 10:07:03</div>
    </div>
```

```
Elements Console
<html>
  <head> </head>
  <body>
    <div id="root"> == $0
      <div>
        <h1>안녕, 리액트!</h1>
        <h2>
          "현재 시간: "
          "오후 9:58:53"
        </h2>
      </div>
```

```
Elements Console
<html>
  <head> </head>
  <body>
    <div id="root"> == $0
      <div>
        <h1>안녕, 리액트!</h1>
        <h2>
          "현재 시간: "
          "오후 10:02:48"
        </h2>
      </div>
```





## 2. Children 자료형

```
React.createElement(  
  type,  
  [props], // 자료형은 객체(교재 36P) 또는 null  
  [...children] // blank 또는 null 또는 여러 개의 element 또는 string  
)
```

= children이 blank 경우

- 코드위 치: c:\ws\_react\CDN\chapter\_05\_children\Child1.js

```
React.createElement(  
  "div",  
  {  
    name: "소플"  
  },  
  )
```

= children이 null인 경우

- 코드위 치: c:\ws\_react\CDN\chapter\_05\_children\Child2.js

```
Elements Console  
<html>  
  <head> </head>  
  <body>  
    <div id="root"> == $0  
      <div name="소플"></div>  
      ...
```

```
Elements Console  
<html>  
  <head> </head>  
  <body>  
    <div id="root"> == $0  
      <div name="소플"></div>
```



## 2. type 자료형

React.createElement(

type, // 자료형은 HTML tag 또는 리액트 Component

[props], // 자료형은 객체(교재 36P) 또는 null

[...children] // blank 또는 null 또는 여러 개의 element 또는 string

)

= type이 HTML tag인 경우

- 코드위치: c:\ws\_react\CDN\chapter\_05\_children\Child1.js

React.createElement("div", {name: "소플"})

= type이 리액트 Component인 경우

- 코드위치: c:\ws\_react\CDN\chapter\_05\_children\index.js  
root.render(React.createElement(Child1));

```
Elements Console
<html>
  <head> </head>
  <body>
    <div id="root"> == $0
      <div name="소플"></div>
      ...
```



### 3. 컴포넌트 만들기

- 컴포넌트의 2가지 종류

- Function component

```
function Child4(props) {  
  return React.createElement(  
    "div", null, "현재 시간");  
}
```

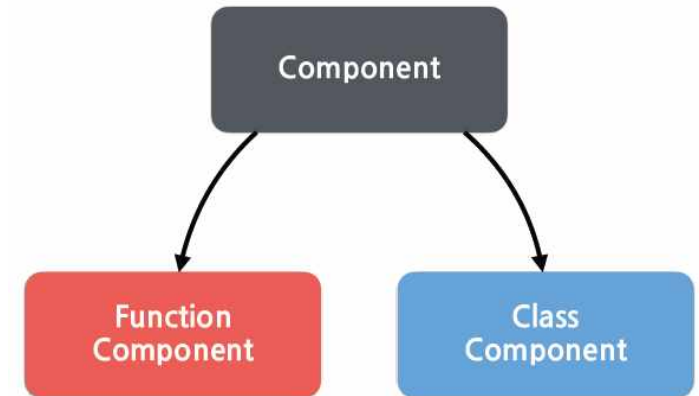
- Class component

```
class Welcome extends React.component {  
  render() {  
    return React.createElement(  
      "div", null, "현재 시간: ");  
  }  
}
```

학생> class는 무엇인가요?

교수> ES6에서 생긴 자료형으로 Java의 class에 해당. 교재 6장에서 자세히 나옴

[https://www.w3schools.com/js/js\\_class\\_inheritance.asp](https://www.w3schools.com/js/js_class_inheritance.asp)





### 3. 컴포넌트 만들기 (컴포넌트 이름짓기)

- 컴포넌트의 이름은 항상 대문자로 시작  
소문자는 **HTML tag**로 인식

```
function Child4(props) {  
  return React.createElement(  
    "div", null, "현재 시간");  
}
```



### 3. 컴포넌트 만들기 (컴포넌트 렌더링)

- 교재 154쪽 코드 분석
- 코드위치: c:\ws\_react\CDN\chapter\_05\_3\_render\render.js
- 12번줄에서 Welcome은 component
- 15번줄에서 Welcome component에 props를 입력하여 출력으로 리액트 엘리먼트 생성
- 20번줄에서 리액트 엘리먼트 렌더링하여 크롬 화면 표시

```
12 function Welcome(props) {  
13   |   return React.createElement("h1", null, "안녕, ", props.name);  
14 }  
15 const element = React.createElement(Welcome, {  
16   |   name: "인제"  
17 });  
18  
19 const root = ReactDOM.createRoot(document.getElementById('root'));  
20 root.render(element);
```



## 4. 컴포넌트 합성

- 컴포넌트의 합성은 여러 개의 컴포넌트를 합쳐서 하나의 컴포넌트를 만드는 것
- 교재 155쪽 코드 분석
- 코드위치: c:\ws\_react\CDN\chapter\_05\_4\_합성\합성.js
- App 컴포넌트는 Welcome 컴포넌트를 합쳐서 하나의 컴포넌트 만듦

### App Component

**Welcome Component**  
props={ name: "Mike" }

**Welcome Component**  
props={ name: "Steve" }

**Welcome Component**  
props={ name: "Jane" }



## 5. 컴포넌트 추출

- 컴포넌트의 추출은 큰 컴포넌트에서 일부를 추출해서 새로운 컴포넌트를 만드는 것
- 교재 157쪽 코드 분석
- 코드위치: c:\ws\_react\CDN\chapter\_05\_5\_추출\Comment\_157.jsx
- 1차 추출
- Avatar\_158.jsx와 Comment\_159.jsx
- 2차 추출
- UserInfo\_160.jsx와 Comment\_160.jsx

학생> 컴포넌트를 어느 정도 수준까지 추출하나요?

교수> 다른 곳에 재사용이 가능한 형태로 추출





# 휴식

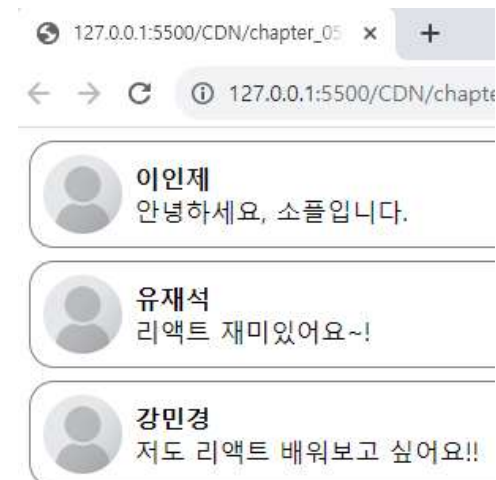
1. 컴포넌트에 대해 알아보기
2. **Props**에 대해 알아보기
3. 컴포넌트 만들기
4. 컴포넌트 합성
5. 컴포넌트 추출
6. 실습: 댓글 컴포넌트 만들기
7. 마치며





## 6. 실습: 댓글 컴포넌트 만들기 (CommentList.js)

- 한줄씩 설명합니다
- 코드위치: c:\ws\_react\CDN\chapter\_05\_6\_reactJS
- CommentList.js 48번 줄  
comment => { }  
동일: function (comment) { }  
function 생략, parentheses () 생략
- 괄호의 영어 명칭 정리  
<https://en.wikipedia.org/wiki/Bracket>



Brackets			
<b>()</b>	<b>[]</b>	<b>{ }</b>	<b>&lt; &gt;</b>
Round brackets	Square brackets	Curly brackets	Angle brackets
or	or	or	or
parentheses	brackets	braces	chevrons
or	(US)		
brackets			
(UK)			



## 6. 실습: 댓글 컴포넌트 만들기(CommentList.js)

- CommentList.js 48번 줄  
comments.map()
- [https://www.w3schools.com/jsref/jsref\\_map.asp](https://www.w3schools.com/jsref/jsref_map.asp)  
map() creates a new array from calling a function for every array element
- CommentList2.js 50번 줄 newArray
- F12에서 55번 줄 중단점  
54번 줄 중단점, 56번 줄 중단점  
54번 줄 중단점 해제
- comments는 객체(object) Array
- newArray는 element Array

```
▼ Module
  ▶ Comment: f Comment(props)
  ▼ comments: Array(3)
    ▶ 0: {name: '미인제', comment: '안녕하세요, 소플입니다.'}
    ▶ 1: {name: '유재석', comment: '리액트 재미있어요~!'}
    ▶ 2: {name: '강민경', comment: '저도 리액트 배워보고 싶어요!!'}
    length: 3
    ▶ [[Prototype]]: Array(0)
  ▼ newArray: Array(3)
    ▶ 0: {$$typeof: Symbol(react.element), key: null, ref: null, p
    ▶ 1: {$$typeof: Symbol(react.element), key: null, ref: null, p
    ▶ 2: {$$typeof: Symbol(react.element), key: null, ref: null, p
    length: 3
    ▶ [[Prototype]]: Array(0)
```



## 6. 실습: 댓글 컴포넌트 만들기 (Comment.js)

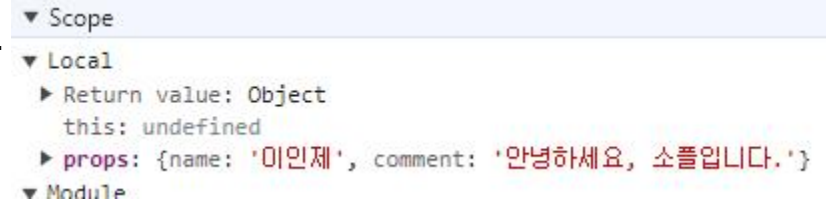
학생> 교수님~ Comment.js가 상당히 복잡한데요.

교수> 자세히 살펴보겠습니다.

먼저 **props**의 내용을 살펴보겠습니다.

F12에서 70번줄 중단점

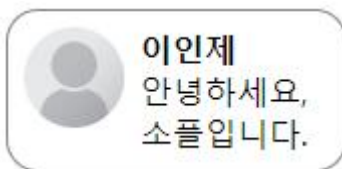
- Local창에 **props**의 내용을 볼 수 있습니다





## 6. 실습: 댓글 컴포넌트 만들기 (Comment.js)

- 크롬 화면과 F12 elements과 Comment.js 코드와 비교해봅니다
- 먼저 크롬화면과 F12 elements 창을 비교해 보면 elements창의 내용을 hovering해 보면 해당 내용이 크롬화면에 표시됩니다



```
▼<div style="margin: 8px; padding: 8px; display: flex; flex-direction: row; border-radius: 16px;"> flex == $0
  ▼<div>
    
  </div>
  ▼<div style="margin-left: 8px; display: flex; flex-direction: column; justify-content: space-between;">
    <span style="color: black; font-size: 16px; font-weight: bold;">이인제</span>
    <span style="color: black; font-size: 16px;">안녕하세요, 소플입니다.</span>
  </div>
</div>
```



## 6. 실습: 댓글 컴포넌트 만들기 (Comment.js)

- F12 elements과 Comment.js 코드와 비교해봅니다
- 57번줄이 ① 영역 style: styles.wrapper
- 59번줄이 ② 영역 style: styles.imageContainer
- 61번줄이 <img> 태그 style: styles.image
- 64번줄이 ③ 영역 style: styles.contentContainer
- 66번줄이 <span> 태그 style: styles.nameText
- 68번줄이 <span> 태그 style: styles.commentText

```
56 function Comment(props) {
57   return React.createElement("div", {
58     style: styles.wrapper
59   }, React.createElement("div", {
60     style: styles.imageContainer
61   }, React.createElement("img", {
62     src: "https://upload.wikimedia.org/wikipedia
63     style: styles.image
64   })), React.createElement("div", {
65     style: styles.contentContainer
66   }, React.createElement("span", {
67     style: styles.nameText
68   }, props.name), React.createElement("span", {
69     style: styles.commentText
70   }, props.comment)));
71 }
```

1 `<div style="margin: 8px; padding: 8px; display: flex; flex-direction: row; border-radius: 16px;"> flex == $0`

2 `<div>`  
``  
`</div>`

3 `<div style="margin-left: 8px; display: flex; flex-direction: column; justify-co" <span style="color: black; font-size: 16px; font-weight: bold;">미인제</span> <span style="color: black; font-size: 16px;">안녕하세요, 소플입니다.</span> </div>`  
`</div>`



# 이제 나는 할 수 있다

1. 나는 컴포넌트를 만들 수 있다
2. 나는 컴포넌트가 만들어낸 **react element**가 **render()**를 통해서 **F12 element**창에서 **HTML tag**로 변환된 것을 확인할 수 있다
3. 나는 **Props**를 만들 수 있다
4. 나는 **Props**가 **F12 element**창에서 **HTML tag**의 **attributes**로 변환된 것을 확인할 수 있다
5. 나는 **children**이 **F12 element**창에서 **child HTML tag**로 변환된 것을 확인할 수 있다
6. 나는 컴포넌트를 합성할 수 있다.  
여러 개의 컴포넌트를 합쳐서 하나의 컴포넌트를 만들 수 있다
7. 나는 컴포넌트를 추출할 수 있다  
큰 컴포넌트에서 일부를 추출해서 새로운 컴포넌트를 만들 수 있다

`React.createElement(`

`type,`

`[props], // 자료형은 객체(교재 36쪽) 또는 null 또는 blank`

`[...children] // blank 또는 null 또는 여러 개의 element 또는 string`

`)`