

# Props

## Props란?

- 컴포넌트 끼리의 정보교환 방식!

## Props의 특징

- props는 반드시 [부모] → [자식] 방향으로만 흐른다(단방향).
- props는 반드시 읽기 전용으로 취급하며, 저장된 값이 변경되지 않는다.  
→ 저장된 값을 변경하려면, 부모의 값을 변경해줘야 한다.

< 예시 >

`Child` 컴포넌트에서 `Mother` 의 이름을 알 수 있을까요?

```
import React from "react";

function App() {
  return <GrandFather />;
}

function GrandFather() {
  return <Mother />;
}

function Mother() {
  const name = '홍부인';
  return <Child />;
}

function Child() {
  return <div>아들입니다.</div>;
}

export default App;
```

`motherName` 이라는 이름으로 `name` 값을 `Child` 컴포넌트에게 전달

→ 이 과정을 “**Props** 로 정보를 전달했다” 라고 표현

```
import React from "react";

function App() {
  return <GrandFather />;
}

function GrandFather() {
  return <Mother />;
}

function Mother() {
  const name = '홍부인';
  return <Child motherName={name} />; // 💡 "props로 name을 전달했다."
}

function Child() {
  return <div>아들입니다.</div>;
}

export default App;
```

`props`의 값(객체형태)을 받아오다.

```
function Child(props){
  console.log(props) // {motherName: '홍부인'}
  return <div>아들입니다.</div>
}
```

## Prop drilling

- props의 문제점 중 하나!
- 부모의 데이터가 많은 자식들을 거쳐 마지막 자식 컴포넌트에 자료를 내려주는 것을 말한다.
- 해결방법 : redux, Mobx, recoil 등 전역 상태관리 라이브러리 사용

< 예시 >

```
export default function App() {
  return (
    <div className="App">
```

```

    <FirstComponent content="Who needs me?" />
  </div>
);
}

function FirstComponent({ content }) {
  return (
    <div>
      <h3>I am the first component</h3>;
      <SecondComponent content={content} />|
    </div>
  );
}

function SecondComponent({ content }) {
  return (
    <div>
      <h3>I am the second component</h3>;
      <ThirdComponent content={content} />
    </div>
  );
}

function ThirdComponent({ content }) {
  return (
    <div>
      <h3>I am the third component</h3>;
      <ComponentNeedingProps content={content} />
    </div>
  );
}

function ComponentNeedingProps({ content }) {
  return <h3>{content}</h3>;
}

```

## Props Children

### Children란?

- props의 일종이지만, 다른 역할을 수행
  - props : 문자열, 숫자, 배열, 객체 등의 데이터를 전달
  - children : 컴포넌트 태그 사이에 작성된 내용을 전달

< 예시 >

```
import React from "react";

function App() {
  return <User>안녕하세요</User>;
}

function User(props) {
  console.log(props) // children : 안녕하세요
  console.log(props.children) // 안녕하세요
  return <div>{props.children}</div>;
}

export default App;
```

## Props와 구조분해할당

### 구조분해할당이란?

- props의 일종이지만, 선언 방법이 다름
  - props : props를 선언
  - 구조분해할당 : 함수나 클래스내에서 변수를 선언
- 코드의 간결성, 가독성을 높일 수 있음

< 예시 >

props 방식

```
function Todo(props){
  return <div>{props.title}</div>
}
```

구조분해할당 방식

```
function Todo({ title }){
  return <div>{title}</div>
}
```

# DefaultProps

- 부모 컴포넌트에서 props를 보내주지 않아도 설정될 초기 값

< 예시 >

```
import React from 'react';

function Child({ name }){
  return <div>내 이름은 {name} 입니다. </div>
}

// 이렇게 설정합니다. // 초기 값이 설정되지 않았을때는 undefined
Child.defaultProps={
  name: '기본 이름'
}

export default Child
```