

9월 5일 (일)

1. 코딩애플 - 리액트강의 - Part 2. 쇼핑몰 프로젝트

1) 쇼핑몰 프로젝트 : 프로젝트 생성 & Bootstrap 설치

yarn 은 npm , npx 보다 빠르게 모듈을 설치한다.

yarn에서 npm run start 는 yarn start로 사용한다.

HTML/ CSS 디자인 쌩코딩이 싫다면 Bootstrap 이용하기

리액트 전용 부트스트랩을 이용해야 한다.

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows a project structure with files like `App.js`, `App.css`, `App.test.js`, etc., under the `src` directory.
- Code Editor:** Displays the `App.js` file content:

```
import React from 'react';
import './App.css';

function App() {
  return (
    <div className="App">
    </div>
  );
}

export default App;
```
- Terminal:** Shows a Windows PowerShell window with the following text:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

새로운 크로스 플랫폼 PowerShell 사용 https://aka.ms/pscore6

PS C:\Users\harry\Desktop\React class\shop> yarn add react-bootstrap bootstrap
```
- Status Bar:** Shows file information: Line 7, Column 7, Spaces: 2, UTF-8, LF, JavaScript.

React-bootstrap 설치하기

터미널에 sudo npm install react-bootstrap bootstrap
또는 sudo yarn add react-bootstrap bootstrap

```

File Edit Selection View Go Run Terminal Help
EXPLORER OPEN EDITORS App.js - shop - Visual Studio Code
src > JS App.js src > App
import React from 'react';
import './App.css';

function App() {
  return (
    <div className="App">
      </div>
  );
}

export default App;

```

TERMINAL

```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

새로운 크로스 플랫폼 PowerShell 사용 https://aka.ms/pscore6

PS C:\Users\harry\Desktop\React class\shop> yarn add react-bootstrap bootstrap
yarn add v1.22.4
[1/4] Resolving packages...
[2/4] Fetching packages...
info fsevents@2.1.2: The platform "win32" is incompatible with this module.
info "fsevents@2.1.2" is an optional dependency and failed compatibility check. Excluding it from installation.
info fsevents@1.2.11: The platform "win32" is incompatible with this module.

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

2: node

Windows PowerShell

Copyright (C) Microsoft Corporation. All rights reserved.

새로운 크로스 플랫폼 PowerShell 사용 <https://aka.ms/pscore6>

PS C:\Users\harry\Desktop\React class\shop> yarn add react-bootstrap bootstrap

yarn add v1.22.4

[1/4] Resolving packages...

[2/4] Fetching packages...

info fsevents@2.1.2: The platform "win32" is incompatible with this module.

info "fsevents@2.1.2" is an optional dependency and failed compatibility check. Excluding it from installation.

info fsevents@1.2.11: The platform "win32" is incompatible with this module.

권한이 없다는 오류가 뜨면 sudo 를 붙여서 다운로드한다.

다운로드 후 public 에 있는 index.html 파일에 아래 호스팅하는 CDN형식의 링크를 입력하고 자료를 받아오는 것이다.

```

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/css/bootstrap.min.css" integrity="sha384-KyZXEAg3QhqLMpG8r+8fhAXLRk2vvoC2f3B09zVXn8CA5QIVfZ0J3BCsw2P0p/We" crossorigin="anonymous"/>

```

안정적인 운영을 위해서는 해당 부트스트랩 파일을 다운로드 한 후 public 폴더나 app.css 파일에 첨부하여 사용한다.

2) 평화로운 쇼핑몰 레이아웃 디자인 시간



이런식으로 짤 것 , 쌩코딩하면 시간이 오래 걸리니 부트스트랩 사용

The screenshot shows a browser window with the URL react-bootstrap.github.io/components/alerts. The page title is "Alerts". The main content area displays three alert examples: a primary alert (blue background), a secondary alert (gray background), and a success alert (green background). Each alert contains the text "This is a [type] alert—check it out!". On the left sidebar, under the "Components" section, the "Alerts" item is selected. A large watermark "React-bootstrap 사이트 들어가서 원하는 UI 검색" is overlaid on the right side of the page.

리액트 사이트 컴포넌트에서 원하는 레이아웃을 검색한다.

The screenshot shows a Visual Studio Code interface with a browser tab at the top displaying "Failed to compile". The code editor shows an error in `App.js` at line 7:8: "'Navban' is not defined". The code is a React component definition:

```

function App() {
  return (
    <div any assName="App">
      <Navbar bg="light" expand="lg">
        <Container>
          <Navbar.Brand href="#home">React-Bootstrap</Navbar.Brand>
          <Navbar.Toggle aria-controls="basic-navbar-nav" />
          <Navbar.Collapse id="basic-navbar-nav">
            <Nav className="me-auto">
              <Nav.Link href="#home">Home</Nav.Link>
              <Nav.Link href="#link">Link</Nav.Link>
              <NavDropdown title="Dropdown" id="basic-nav-dropdown">
                <NavDropdown.Item href="#action/3.1">Action</NavDropdown.Item>
                <NavDropdown.Item href="#action/3.2">Another action</NavDropdown.Item>
                <NavDropdown.Item href="#action/3.3">Something</NavDropdown.Item>
                <NavDropdown.Divider />
                <NavDropdown.Item href="#action/3.4">Separated link</NavDropdown.Item>
              </NavDropdown>
            </Nav>
          </Navbar.Collapse>
        </Container>
      </Navbar>
    
```

A message at the bottom of the code editor says "Search for the keywords to learn more about this error".

맘에 드는 예제를 선정하여 그대로 복사한다. 하지만 에러가 뜨게 된다.

Importing Components

You should import individual components like: `react-bootstrap/Button` rather than the entire library. Doing so pulls in only the specific components that you use, which can significantly reduce the amount of code you end up sending to the client.

```

import Button from 'react-bootstrap/Button';
// or less ideally
import { Button } from 'react-bootstrap';

```

Browser globals

We provide `react-bootstrap.js` and `react-bootstrap.min.js` bundles with all components exported on the `window.ReactBootstrap` object. These bundles are available on `unpkg`, as well as in the npm package.

```

<script
  src="https://unpkg.com/react/umd/react.production.min.js"
  crossorigin></script>

```

```

import React from 'react'
import { Button } from 'react-bootstrap';
import './App.css';

function App() {
  return (
    <div className="App">
      <Navbar bg="light" expand="lg">
        <Container>
          <Navbar.Brand href="#home">React-Bootstrap</Navbar.Brand>
          <Navbar.Toggle aria-controls="basic-navbar-nav" />
          <Navbar.Collapse id="basic-navbar-nav">
            <Nav className="me-auto">
              <Nav.Link href="#home">Home</Nav.Link>
              <Nav.Link href="#link">Link</Nav.Link>
              <NavDropdown title="Dropdown" id="basic-nav-dropdown">
                <NavDropdown.Item href="#action/3.1">Action</NavDropdown.Item>
                <NavDropdown.Item href="#action/3.2">Another action</NavDropdown.Item>
                <NavDropdown.Item href="#action/3.3">Something</NavDropdown.Item>
                <NavDropdown.Divider />
                <NavDropdown.Item href="#action/3.4">Separated link</NavDropdown.Item>
              </NavDropdown>
            </Nav>
          </Navbar.Collapse>
        </Container>
      </Navbar>
    </div>
  );
}

export default App;

```

컴포넌트를 import를 해 와야지 사용이 가능하다.

```

import React from 'react'
import { Navbar, Container, Nav } from 'react-bootstrap';
import './App.css';

function App() {
  return (
    <div className="App">
      <Navbar bg="light" expand="lg">
        <Container>
          <Navbar.Brand href="#home">React-Bootstrap</Navbar.Brand>
          <Navbar.Toggle aria-controls="basic-navbar-nav" />
          <Navbar.Collapse id="basic-navbar-nav">
            <Nav className="me-auto">
              <Nav.Link href="#home">Home</Nav.Link>
              <Nav.Link href="#link">Link</Nav.Link>
              <NavDropdown title="Dropdown" id="basic-nav-dropdown">
                <NavDropdown.Item href="#action/3.1">Action</NavDropdown.Item>
                <NavDropdown.Item href="#action/3.2">Another action</NavDropdown.Item>
                <NavDropdown.Item href="#action/3.3">Something</NavDropdown.Item>
                <NavDropdown.Divider />
                <NavDropdown.Item href="#action/3.4">Separated link</NavDropdown.Item>
              </NavDropdown>
            </Nav>
          </Navbar.Collapse>
        </Container>
      </Navbar>
    </div>
  );
}

export default App;

```

용량을 아끼기 위해 {}를 사용해서 일부만 임포트해온다.

```

<NavDropdown.Item href="#action/3.4">Separated link</NavDropdown.Item>
          </NavDropdown>
        </Nav>
      </Navbar.Collapse>
    </Container>
  </Navbar>
<Jumbotron className="background">
  <h1>20% Season Off</h1>
  <p>
    This is a simple hero unit, a simple jumbotron-style component for calling extra attention to featured content or information.
  </p>
  <p>
    <Button variant="primary">Learn more</Button>
  </p>
</Jumbotron>

</div>
);
}

export default App;

```

이미지는 public이나 src에 저장하면 된다. 차이점은 아래와 같다.

```

    <NavDropdown.Item href="#action/3.4">Separated link</NavDropdown.Item>
  </NavDropdown>
  </Nav>
  </Navbar.Collapse>
</Container>
</Navbar>

<Jumbotron className="background">
  <h1>20% Season Off</h1>
  <p>
    This is a simple hero unit, a simple jumbotron-style component for calling extra attention to featured content or information.
  </p>
  <p>
    <Button variant="primary">Learn more</Button>
  </p>
</Jumbotron>

</div>
);
}

export default App;

```

public 은 문법 같은 걸 써야해서 약간 귀찮을 수 있다. 폴경로를 다 써야지 가져올 수 있다. src에 하위폴더 만들어서 보관해라

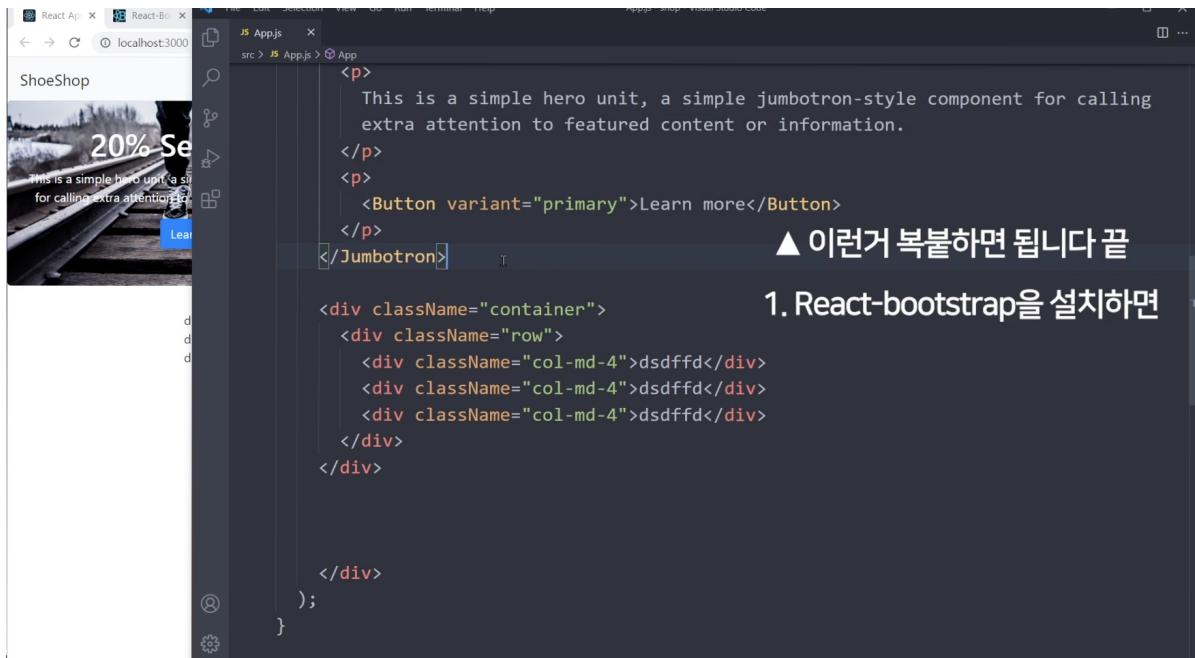
```

<p>
  This is a simple hero unit, a simple jumbotron-style component for calling extra attention to featured content or information.
</p>
<p>
  <Button variant="primary">Learn more</Button>
</p>
</Jumbotron>

<div className="container">
  <div className="row">
    <div className="col-md-4">dsdff</div>
    <div className="col-md-4">dsdff</div>
    <div className="col-md-4">dsdff</div>
  </div>
</div>
)
}

```

상품목록 넣는 프레임 짜기 (문법과 같다.)



This screenshot shows a Visual Studio Code interface with two tabs open: 'React API' and 'React-Bootstrap'. The current file is 'App.js' under the 'src' directory. The code displays a hero unit component from React-Bootstrap:

```
<p>
  This is a simple hero unit, a simple jumbotron-style component for calling extra attention to featured content or information.
</p>
<p>
  <Button variant="primary">Learn more</Button>
</p>
</Jumbotron>

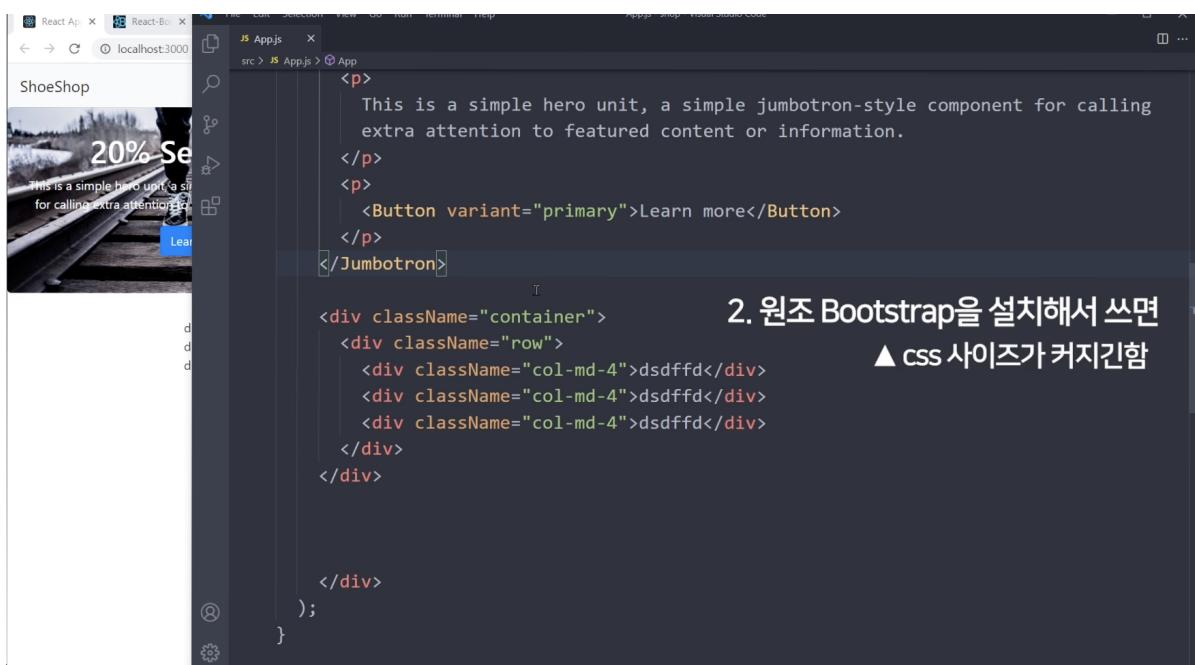
<div className="container">
  <div className="row">
    <div className="col-md-4">dsdfffd</div>
    <div className="col-md-4">dsdfffd</div>
    <div className="col-md-4">dsdfffd</div>
  </div>
</div>

</div>
);
```

The browser preview on the left shows a hero unit with a background image of a railway track, a '20% Off' discount banner, and a 'Learn more' button.

▲ 이런거 복붙하면 됩니다 끝

1. React-bootstrap을 설치하면



This screenshot shows the same setup as the first one, but the code has been modified to use Bootstrap's grid system directly:

```
<p>
  This is a simple hero unit, a simple jumbotron-style component for calling extra attention to featured content or information.
</p>
<p>
  <Button variant="primary">Learn more</Button>
</p>
</Jumbotron>

<div className="container">
  <div className="row">
    <div className="col-md-4">dsdfffd</div>
    <div className="col-md-4">dsdfffd</div>
    <div className="col-md-4">dsdfffd</div>
  </div>
</div>

</div>
);
```

The browser preview remains the same, showing the hero unit with the铁路轨道背景和优惠券。

2. 원조 Bootstrap을 설치해서 쓰면

▲ css 사이즈가 커지긴 함

```
<p>
    This is a simple hero unit, a simple jumbotron-style component for calling extra attention to featured content or information.
</p>
<p>
    <Button variant="primary">Learn more</Button>
</p>
</Jumbotron>

```

▼ 원조는 이런 식으로 class명을 부여해서 씁니다

```
<div className="container">
    <div className="row">
        <div className="col-md-4">dsdffd</div>
        <div className="col-md-4">dsdffd</div>
        <div className="col-md-4">dsdffd</div>
    </div>
</div>

```

```
);
```

2. 원조 Bootstrap을 설치해서 쓰면

3) 코드가 넘나 길어진다면 import / export 사용해보기

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, GO, Run, Terminal, Help.
- Editor:** The active editor is 'data.js' under the 'SHOP' folder. The code is as follows:

```
src > JS data.js > [e] default
var name = 'Kim';

export default name

// [
//   {
//     id : 0,
//     title : "White and Black",
//     content : "Born in France",
//     price : 120000
//   },
//   {
//     id : 1,
//     title : "Red Knit",
//     content : "Born in Seoul",
//     price : 110000
//   },
//   {
//     id : 2,
//     title : "Grey Yordan",
//     content : "Born in the States",
//   }
]
```

A callout box highlights the `export default name` line with the text "내보내기 : export default 변수명".

export default 는 딱 한 번만 사용이 가능

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, GO, Run, Terminal, Help.
- Editor:** The active editor is 'App.js'. The code is as follows:

```
import React, {useState} from 'react';
import { Navbar, Nav, NavDropdown, Jumbotron } from 'react-bootstrap';
import './App.css';
import ??? from './data.js';
```

The code contains several syntax errors, indicated by red squiggly lines under the imports and the question mark placeholder.

```

App.js
import React, {useState} from 'react';
import { Navbar, Nav, NavDropdown, Button, Jumbotron } from 'react-bootstrap';
import './App.css';
import name from './data.js';

function App() {
  let [shoes, shoes변경] = useState([]);
  return (
    <div className="App">
      <Navbar bg="light" expand="lg">
        <Navbar.Brand href="#home">ShoeShop</Navbar.Brand>
        <Navbar.Toggle aria-controls="basic-navbar-nav" />
        <Navbar.Collapse id="basic-navbar-collapse">
          <Nav className="ml-auto">
            <Nav.Link href="#home">Home</Nav.Link>
          </Nav>
        </Navbar.Collapse>
      </Navbar>
    </div>
  );
}

export default App;

```

```

data.js
var name = 'Kim';

export default name;

```

내보내기 : export default 변수명
가져오기 : import 변수명 from 경로

default 변수를 import 할 때 작명은 하고싶은대로 작명

```
App.js
import React, {useState} from 'react';
import { Navbar, Nav, NavDropdown, Button, Jumbotron } from 'react-bootstrap';
import './App.css';

function App() {
  let [shoes, shoes변경] = useState([]);
  return (
    <div className="App">
      <Navbar bg="light" expand="lg">
        <Navbar.Brand href="#home">ShoeShop</Navbar.Brand>
        <Navbar.Toggle aria-controls="basic-navbar-nav" />
        <Navbar.Collapse id="basic-navbar-nav">
          <Nav className="mr-auto">
            <Nav.Link href="#home">Home</Nav.Link>
            <Nav.Link href="#link">Link</Nav.Link>
          <NavDropdown title="Dropdown" >
            ...
          </NavDropdown>
        </Nav>
      </Navbar.Collapse>
    </div>
  );
}

export default App;
```

```
data.js
var name = 'Kim';
var name2 = 'Park';

export { name, name2 }
```

디풀트는 한 번만 가능하니, 여러 개의 변수를 사용하고자 할 땐 {}를 사용하여 변수 이름 그대로를 써준다.

import 할 때도 {} 안에 변수이름을 그대로 써줘야 한다.

The screenshot shows a code editor with two tabs open: `App.js` and `data.js`.

App.js:

```
src > JS App.js > ...
import React, {useState} from 'react';
import { Navbar, Nav, NavDropdown, Button, Jumbotron } from 'react-bootstrap';
import './App.css';
import [ name, name2 ] from './data.js';

function App() {
  let [shoes, shoes변경] = useState([]);
  return (
    <div className="App">
      <Navbar bg="light" expand="lg">
        <Navbar.Brand href="#home">ShoeShop</Navbar.Brand>
        <Navbar.Toggle aria-controls="basic-navbar-nav" />
        <Navbar.Collapse id="basic-navbar-nav">
          <Nav className="mr-auto">
            <Nav.Link href="#home">Home</Nav.Link>
            <Nav.Link href="#">Link</Nav.Link>
          <NavDropdown title="Dropdown">
```

data.js:

```
src > JS data.js > ...
var name = 'Kim';
var name2 = 'Park';

export { name, name2 }
```

A tooltip is displayed over the `name` variable in the `App.js` code, containing the following text:

내보내기 : `export { 변수1, 변수2 }`
가져오기 : `import { 변수1, 변수2 } from 경로`

The screenshot shows a code editor with two files open: `App.js` and `data.js`.
`App.js` contains React component code with a template string that includes a variable `{ name }`.
`data.js` contains a JavaScript object with two properties: `name` and `name2`, both set to strings.
A tooltip in the center of the screen contains the following Korean text:
내보내기 : `export { 변수1, 변수2 }`
가져오기 : `import { 변수1, 변수2 } from 경로`

리턴문 안에서 사용할 때는 중괄호를 써서 사용한다.

4) 숙제 해설 : 상품목록 Component화 + 반복문

```


<div className="row">
    <div className="col-md-4">
      
      <h4>상품명</h4>
      <p>상품설명 & 가격</p>
    </div>
    <div className="col-md-4">
      
      <h4>상품명</h4>
      <p>상품설명 & 가격</p>
    </div>
  </div>


```

component로 만들기

반복된 태그를 컴포넌트화 해주자 ! 가독성을 높이고 디버깅을 쉽게 할 수 있다.

```


<div className="row">
    <div className="col-md-4">
      <Card>
        <div className="col-md-4">
          
          <h4>상품명</h4>
          <p>상품설명 & 가격</p>
        </div>
        <div className="col-md-4">
          
          <h4>상품명</h4>
          <p>상품설명 & 가격</p>
        </div>
      </Card>
    </div>
  </div>


```

[component 제작법]

1. function 컴포넌트이름(){}
2. return (<div></div>)
3. 필요한 곳에 <컴포넌트이름 />

ReferenceError: shoes is not defined

```

Card
C:\Users\harry\Desktop\React class\shop\src\app.js:75
    72 |   return (
    73 |     <div className="col-md-4">
    74 |       
    75 |       <p>{shoes[0].title}</p>
    76 |       <p>{shoes[0].content} & {shoes[0].price}</p>
    77 |     </div>
    78 |   )
  View compiled
  render@file:///C:/Users/harry/Desktop/React class/shop/src/app.js:14:80
  C:\Users\harry\Desktop\React class\shop\src\index.html:1:14
  > 14800 |   <div>
  14801 |     <h1>ShoeShop</h1>
  14802 |
  > 14803 |   var children = Component(props, secondArg);
  a render phase update
  14804 |   ...
  14805 |   // If (workInProgress.expirationTime === renderExpirationTime)
  14806 |   // Keep rendering in a loop for as long as it continues to
  View compiled
  mountIndeterminateComponent
C:\Users\harry\Desktop\React class\shop\src\index.html:1:14
  > 14807 |   setIsRendering(true);
  14808 |   ReactCurrentOwner.current = workInProgress;
  14809 |   value = renderIndeterminate(null, workInProgress,
  context, renderExpirationTime);
  14810 |   setIsRendering(false);
  14811 |   // React DevTools reads this flag.
  14812 |
  View compiled
  beginWork
C:\Users\harry\Desktop\React class\shop\src\index.html:1:14
  > 14813 |   switch (workInProgress.tag) {
  14814 |     case IndeterminateComponent:
  14815 |       {
  > 14816 |         return mountIndeterminateComponent(currentWorkInProgress.type, renderExpirationTime);
  14817 |       }

```

shoes라는 state는
부모 컴포넌트가 가지고 있음

ReferenceError: shoes is not defined

```

Card
C:\Users\harry\Desktop\React class\shop\src\app.js:75
    72 |   return (
    73 |     <div className="col-md-4">
    74 |       
    75 |       <p>{shoes[0].title}</p>
    76 |       <p>{shoes[0].content} & {shoes[0].price}</p>
    77 |     </div>
  View compiled
  render@file:///C:/Users/harry/Desktop/React class/shop/src/app.js:14:80
  C:\Users\harry\Desktop\React class\shop\src\index.html:1:14
  > 14800 |   <div>
  14801 |     <h1>ShoeShop</h1>
  14802 |
  > 14803 |   var children = Component(props, secondArg);
  a render phase update
  14804 |   ...
  14805 |   // If (workInProgress.expirationTime === renderExpirationTime)
  14806 |   // Keep rendering in a loop for as long as it continues to
  View compiled
  mountIndeterminateComponent
C:\Users\harry\Desktop\React class\shop\src\index.html:1:14
  > 14807 |   setIsRendering(true);
  14808 |   ReactCurrentOwner.current = workInProgress;
  14809 |   value = renderIndeterminate(null, workInProgress,
  context, renderExpirationTime);
  14810 |   setIsRendering(false);
  14811 |   // React DevTools reads this flag.
  14812 |
  View compiled
  beginWork
C:\Users\harry\Desktop\React class\shop\src\index.html:1:14
  > 14813 |   switch (workInProgress.tag) {
  14814 |     case IndeterminateComponent:
  14815 |       {
  > 14816 |         return mountIndeterminateComponent(currentWorkInProgress.type, renderExpirationTime);
  14817 |       }

```

shoes라는 state는
부모 컴포넌트가 가지고 있음
그걸 자식 컴포넌트가 사용하고 싶으면
props로 전송해줘야 함

ReferenceError: shoes is not defined

```
C:\Users\harry\Desktop\React class\shop>node src/App.js:79

72 | return (
73 |   <div className="col-md-4">
74 |     
75 |     <div><{shoes[0].title}>/</div>
76 |     <p>{shoes[0].content}& {shoes[0].price}</p>
77 |   </div>
78 | )
View compiled
renderChildren
C:\Users\harry\Desktop\React class\shop>node_modules/react-dom.development.js:14480
14480 | }
14481 | }
14482 |
14483 | var children = Component(props, secondary); // a render phase update
14484 | if (isWorkInProgress.expirationTime === renderExpirationTime) {
14485 |   // keep rendering in a loop for as long as it continues to
continue to
View compiled
mountIndeterminateComponent
C:\Users\harry\Desktop\React class\shop>node_modules/react-dom.development.js:17482
17479 |
17480 |   setIsRendering(true);
17481 |   ReactCurrentOwner.current = workInProgress;
17482 |   renderExpirationTime = null;
17483 |   renderExpirationTime = renderExpirationTime;
17484 |   // React DevTools reads this flag.
17485 |
View compiled
beginWork
C:\Users\harry\Desktop\React class\shop>node_modules/react-dom.development.js:18896
18896 | switch (workInProgress.tag) {
18894 |   case IndeterminateComponent:
18895 |   {
18896 |     return mountIndeterminateComponent(currentElement, type, renderExpirationTime);
18897 |   }
18898 | }
18899 | 
```

<div className="container">
 <div className="row">
 <Card shoes={shoes}/>
 <div className="col-md-4">

 <h4>상품명</h4>
 <p>상품설명 & 가격</p>
 </div>
 <div className="col-md-4">

 <h4>상품명</h4>
 <p>상품설명 & 가격</p>
 </div>
 </div>
</div>

[props 전송법]

1. <자식컴포넌트 보낼이름={전송할state}>/>

[props 전송법]

1. <자식컴포넌트 보낼이름={전송할state} />



```
https://react-bootcamp.tistory.com/100
File Edit Selection View Go Run Terminal Help
App.js - shop - Visual Studio Code
src > JS App.js > Card
</div>
</div>
</div>
);
}

function Card(props){
  return (
    <div className="col-md-4">
      
      <h4>{ shoes[0].title }</h4>
      <p>{ shoes[0].content } & { shoes[0].price }</p>
    </div>
  )
}

export default App;
```

[props 전송법]

1. <자식컴포넌트 보낼이름={전송할state} />
2. function 자식컴포넌트(props){}

[props 전송법]

1. <자식컴포넌트 보낼이름={전송할state} />

2. function 자식컴포넌트(props){}



The screenshot shows a React application running in a browser and its corresponding code in Visual Studio Code. The browser window displays three cards, each featuring a different title: "White and Black" (repeated twice) and "Born in France & 120000". Each card also includes an image of a black sneaker with a red, white, and blue stripe on the side.

The code in `App.js` uses the `Card` component multiple times with different props:

```
<div className="container">
  <div className="row">
    <Card shoes={shoes}/>
    <Card shoes={shoes}/>
    <Card shoes={shoes}/>
  </div>
</div>

</div>
);
}

function Card(props){
  return (
    <div className="col-md-4">
      
      <h4>{ props.shoes[0].title }</h4>
      <p>{ props.shoes[0].content } & { props.shoes[0].price }</p>
    </div>
  );
}
```

A large yellow annotation box covers the right side of the code editor, containing Korean text:

Card마다
각각 다른 제목을 가져야 하는데..?
- 각각 컴포넌트마다
다른 제목을 전달해주면 되지 않을까요



The screenshot shows a browser window with the URL <https://react-boom-001001. firebaseapp.com/> and a Visual Studio Code editor side-by-side.

Browser View:

- A hero unit with a large image of a black and white striped sneaker.
- The text "20% Season Offer" in large orange font.
- A subtext: "This is a simple hero unit, a simple jumbotron component for calling extra attention to certain information."
- A blue button labeled "Learn more".

Code Editor (VS Code):

File structure: `src > JS > App.js`

App.js Content:

```
<Card shoes={shoes[0]} />
<Card shoes={shoes[1]} />
<Card shoes={shoes[2]} />
</div>
</div>

</div>
);

}

function Card(props){
return [
  <div className="col-md-4">
    
    <h4>{ props.shoes.title }</h4>
    <p>{ props.shoes.content } & { props.shoes.price }</p>
  </div>
]
}
```

Code Editor (VS Code):

File structure: `src > JS > data.js`

data.js Content:

```
export default [
  {
    id : 0,
    title : "White and Black",
    content : "Born in France",
    price : 120000
  },
  {
    id : 1,
    title : "Red Knit",
    content : "Born in Seoul",
    price : 110000
  },
  {
    id : 2,
    title : "Grey Yordan",
    content : "Born in the States",
    price : 130000
  }
]
```

Bottom Bar:

<Card/> 마다 다른 상품 데이터 전달하기

`shoes[0]` 은 하나의 객체로 `props.shoes.title` 이라고 하면 `shoes`는 자동으로 `shoes[0]` 을 가리키는 것이다.

이번에 반복된 컴포넌트를 반복문으로 처리해보자.

<Card/> 3개를 반복문에 담아보세요

```


<div className="row">
    <Card shoes={shoes[0]} />
    <Card shoes={shoes[1]} />
    <Card shoes={shoes[2]} />
  </div>
</div>
);
}

function Card(props){
  return (
    <div className="col-md-4">
      
      <h4>{ props.shoes.title }</h4>
      <p>{ props.shoes.content } & { props.shoes.price }</p>
    </div>
  )
}


```

[map 반복문]

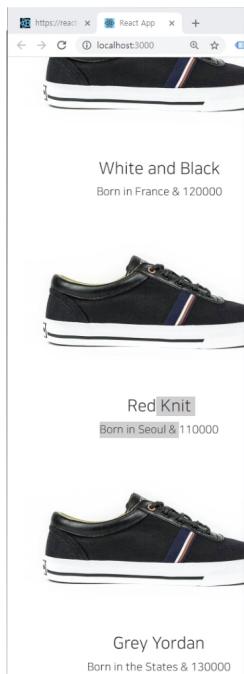
```


<div className="row">
    [
      shoes.map((a,i)=>{
        return <Card shoes={shoes[i]} />
      })
    ]
  </div>
</div>
);
}

function Card(props){
  return (
    <div className="col-md-4">
      
      <h4>{ props.shoes.title }</h4>
      <p>{ props.shoes.content } & { props.shoes.price }</p>
    </div>
  )
}


```

1. 자료.map()=>{}
 2. return (반복시킬 HTML)
 3. 반복하며 변해야하는 숫자체크



The screenshot shows a React application running in a browser window. The code in the editor is as follows:

```

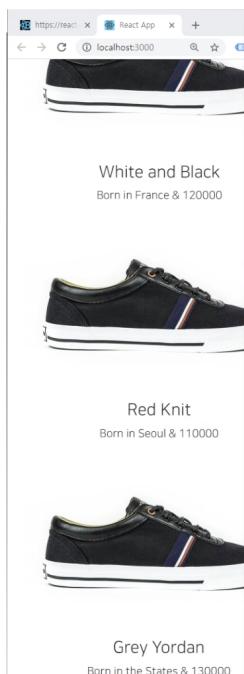

<div className="row">
    {
      shoes.map((a,i)=>{
        return <Card shoes={a}/>
      })
    }
  </div>
</div>
};

function Card(props){
  return (
    <div className="col-md-4">
      
      <h4>{ props.shoes.title }</h4>
      <p>{ props.shoes.content } & { props.shoes.price }</p>
    </div>
  )
}


```

또는 a 를 집어넣어도 된다. a는 객체 shoes[0],[1],[2] 를 차례대로 나타내기 때문이다.

이미지 src 에 데이터바인딩 하기



The screenshot shows the same React application with the code modified to bind the image source to the `shoes[i].src` property. A note on the right side of the code editor reads:

src=" " 에다가 데이터바인딩하려면?
src={} 이렇게하면 변수명 함수명 넣기 가능

The code is now:

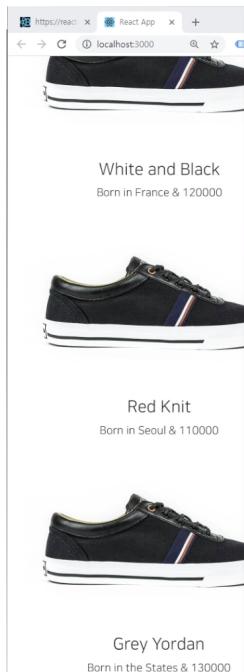
```


<div className="row">
    {
      shoes.map((a,i)=>{
        return <Card shoes={shoes[i]}/>
      })
    }
  </div>
</div>
};

function Card(props){
  return (
    <div className="col-md-4">
      <img src={[ 'https://codingapple1.github.io/shop/shoes???.'+jpg ]} width="100%" />
      <h4>{ props.shoes.title }</h4>
      <p>{ props.shoes.content } & { props.shoes.price }</p>
    </div>
  )
}


```

??? 에 1, 2, 3 ... 0| 오도록 할 것이다.



The screenshot shows a React application running in a browser at <https://react:localhost:3000>. The application displays three shoe cards. The first card is for 'White and Black' shoes, born in France for 120000. The second card is for 'Red Knit' shoes, born in Seoul for 110000. The third card is for 'Grey Yordan' shoes, born in the States for 130000. The code in the editor is as follows:

```

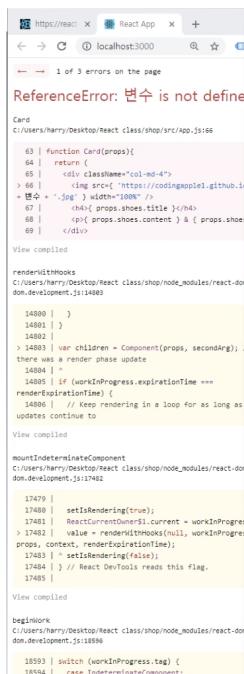
    {
      shoes.map((a,i)=>{
        return <Card shoes={shoes[i]}/>
      })
    }
  </div>
</div>
);
}

function Card(props){
  return (
    <div className="col-md-4">
      <img src={ 'https://codingapple1.github.io/shop/shoes' + 변수 + '.jpg' } width="100%" />
      <h4>{ props.shoes.title }</h4>
      <p>{ props.shoes.content } & { props.shoes.price }</p>
    </div>
  )
}

```

**텍스트 중간에 변수를 넣고싶으면
'문자' + 변수 + '문자'**

이런 식으로 문자열로 끓어주면 된다.



The screenshot shows a browser window with a `ReferenceError: 변수 is not defined` error. The error occurs in the `render` method of the `App` component. The code is as follows:

```

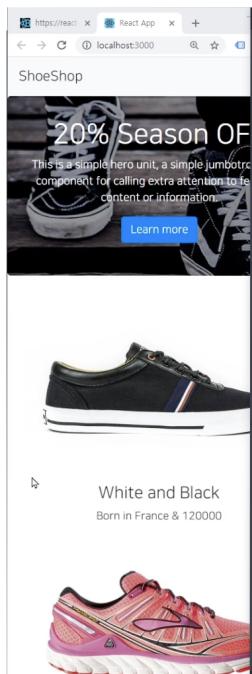
  {
    shoes.map((a,i)=>{
      return <Card shoes={shoes[i]}>/>
    })
  }
</div>
</div>
);
}

function Card(props){
  return (
    <div className="col-md-4">
      <img src={ 'https://codingapple1.github.io/shop/shoes' + 변수 + '.jpg' } width="100%" />
      <h4>{ props.shoes.title }</h4>
      <p>{ props.shoes.content } & { props.shoes.price }</p>
    </div>
  )
}

```

▲ 반복문 돌때마다 0,1,2가 되는 변수

**부모가 가지고 있는 state, 변수 이런건
props로 전송해야 쓸 수 있죠**



The screenshot shows a React application titled "ShoeShop". On the left, there's a hero unit with a black background featuring a pair of white sneakers and the text "20% Season OFF". Below the hero unit are two shoe cards. The first card shows a black and white sneaker with the text "White and Black" and "Born in France & 120000". The second card shows a pink and white sneaker.

```

    JS App.js | index.html
    src > JS App.js > Card
    {
      shoes.map((a,i)=>{
        return <Card shoes={shoes[i]} i={i}/>
      })
    }
  
```

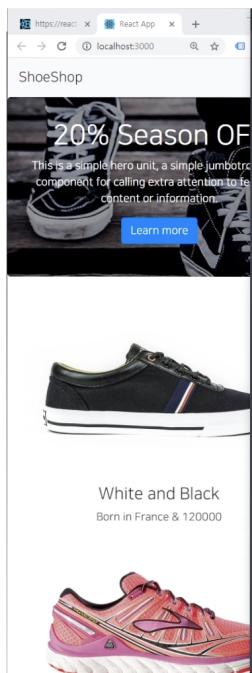
▲ 반복문 돌 때마다 0,1,2가 되는 변수

```

    function Card(props){
      return (
        <div className="col-md-4">
          <img src={ 'https://codingapple1.github.io/shop/shoes' + (props.i + 1) + '.jpg' } width="100%" />
          <h4>{ props.shoes.title }</h4>
          <p>{ props.shoes.content } & { props.shoes.price }</p>
        </div>
      )
    }
  
```

반복문 돌 때마다 1,2,3이 되는 변수▼

왜 이런 짓을 하는가?



The screenshot shows the same ShoeShop application as before. The hero unit and shoe cards are identical.

```

    JS App.js | index.html
    src > JS App.js > App
    <div className="container" >
      <div className="row" >
        {
          shoes.map((a,i)=>{
            return <Card shoes={shoes[i]} i={i}/>
          })
        }
      </div>
    </div>
  
```

Q. 왜 이 짓거리를 나에게 시키는가?
- 서버에서 사이트 구성에 필요한 데이터 받아온 후
- HTML로 보여주는게 프론트엔드 개발자의 역할

```

    function Card(props){
      return (
        <div className="col-md-4">
          <img src={ 'https://codingapple1.github.io/shop/shoes' + (props.i + 1) + '.jpg' } width="100%" />
          <h4>{ props.shoes.title }</h4>
          <p>{ props.shoes.content } & { props.shoes.price }</p>
        </div>
      )
    }
  
```

반복문을 돌리는 이유

ReferenceError: i is not defined

```

App
C:/Users/harry/Desktop/React class/shop/src/App.js:16
54 |     return <Card shoes={shoes[i]} i={i}>
54 |   }
55 |
> 56 |   <Card shoes={shoes[1]} i={1}>
57 |   <Card shoes={shoes[1]} i={1}>
58 |   <Card shoes={shoes[1]} i={1}>
59 | 
```

View compiled

```

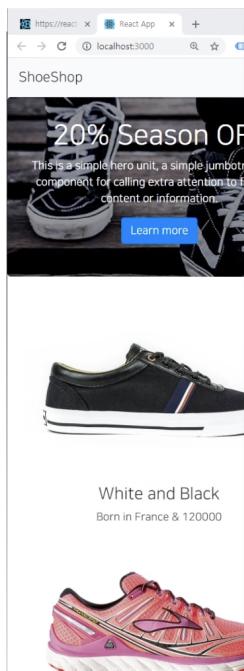
renderInListMode
C:/Users/harry/Desktop/React class/shop/node_modules/react-dom/development.js:1480
1480 |     }
1481 |   }
1482 |
> 1483 |   var children = Component(props, secondArg);
| there was a render phase update
1484 | 
1485 |   if (workInProgress.expirationTime ===
renderExpirationTime) {
1486 |     // Keep rendering in a loop for as long as it
| updates continue to
View compiled
mountIndeterminateComponent
C:/Users/harry/Desktop/React class/shop/node_modules/react-dom/development.js:1482
1482 |     setRendering(true);
1483 |     if (workInProgress.expirationTime > workInProgress.props.context.renderExpirationTime) {
1484 |       value = renderInHost(null, workInProgress);
1485 |     } else {
1486 |       setRendering(false);
1487 |     }
1488 |   } // React DevTools reads this flag.
1489 |
View compiled
beginWork
C:/Users/harry/Desktop/React class/shop/node_modules/react-dom/development.js:1896
1896 |   switch (workInProgress.tag) {
1897 |     case IndeterminateComponent:
1898 |       return mountIndeterminateComponent(currentlyActiveFiber,
1899 |     )

```

Q.. 어렵게 반복문 만들려도 되지 않을까?
- 데이터가 3개인걸 미리 알고 있으면 3번 손수 복붙해도 OK
- 갑자기 데이터가 4개가 되면? 손수 복붙 몇번해야할지 모름

반복문을 돌린 후 key 는 꼭 넣어주자!

ShoeShop



20% Season OFF

This is a simple hero unit, a simple jumbotron component for calling extra attention to featured content or information.

[Learn more](#)

White and Black
Born in France & 120000

반복시킨 HTML에는
key={} 이게 꼭 필요합니다

```

JS App.js
src > JS App.js > App > shoes.map() callback
App.js - shop - Visual Studio Code

<div className="container">
  <div className="row">
    {
      shoes.map((a,i)=>{
        return <Card shoes={shoes[i]} i={i}> <key={i}>
      })
    }
  </div>
</div>
);

function Card(props){
  return (
    <div className="col-md-4">
      <img src={ 'https://codingapple1.github.io/shop/shoes' + (props.i + 1) + '.jpg' } width="100%" />
      <h4>{ props.shoes.title }</h4>
      <p>{ props.shoes.content } & { props.shoes.price }</p>
    </div>
  );
}

function App() {
  return (
    <div>
      <h1>ShoeShop</h1>
      <div>
        <h2>20% Season OFF</h2>
        <p>This is a simple hero unit, a simple jumbotron component for calling extra attention to featured content or information.</p>
        <a href="#">Learn more</a>
      </div>
      <div>
        <img alt="A black and white sneaker with a blue stripe." data-bbox="148 648 275 695" />
        <div>
          <h3>White and Black</h3>
          <p>Born in France & 120000</p>
        </div>
        <img alt="A pink and red running shoe." data-bbox="148 758 275 805" />
      </div>
    </div>
  );
}

export default App;

```