

NEXT.js (홈페이지 : <https://nextjs.org/>)

- => 리액트기반으로 한 Full Stack 웹 애플리케이션 프레임워크
- => Front-end 와 Back-end 두가지 기능을 모두 가지고 있음
- => React , Express.js
- => Server Side Rendering 지원

1. 설치 및 실행
- vscode에서 새창을 실행
 - next-app 폴더 만들고 들어가기
 - 터미널에서 `npx create-next-app@latest .` 탭으로 위치 변경

```
PS F:\next-app> npx create-next-app@latest .
✓ Would you like to use TypeScript? ... No / Yes
✓ Would you like to use ESLint? ... No / Yes
✓ Would you like to use Tailwind CSS? ... No / Yes
✓ Would you like to use `src/` directory? ... No / Yes
✓ Would you like to use App Router? (recommended) ... No / Yes
✓ Would you like to customize the default import alias? ... No / Yes
Creating a new Next.js app in F:\next-app.

Using npm.

Initializing project with template: app-tw
```

개발 서버 실행

`npm run dev` 개발 서버를 시작하려면 실행하세요 .

`http://localhost:3000` 신청서를 보려면 방문하세요 .

파일을 편집 `app/layout.tsx`(또는 `pages/index.tsx`)하고 저장하면 브라우저에서 업데이트된 결과를 볼 수 있습니다.

- 내용보기
1. 쓸모없는 내용 지우기

<pre>import './globals.css' import { Inter } from 'next/font/google' const inter = Inter({ subsets: ['latin'] }) export const metadata = { title: 'Create Next App', description: 'Generated by create next app', } export default function RootLayout({ children }) { return (<html lang="en"> <body className={inter.className}>{children}</body> </html>) }</pre>	<pre>import './globals.css' export const metadata = { title: 'Create Next App', description: 'Generated by create next app', } export default function RootLayout({ children }) { return (<html > <body>{children}</body> </html>) }</pre>
---	--

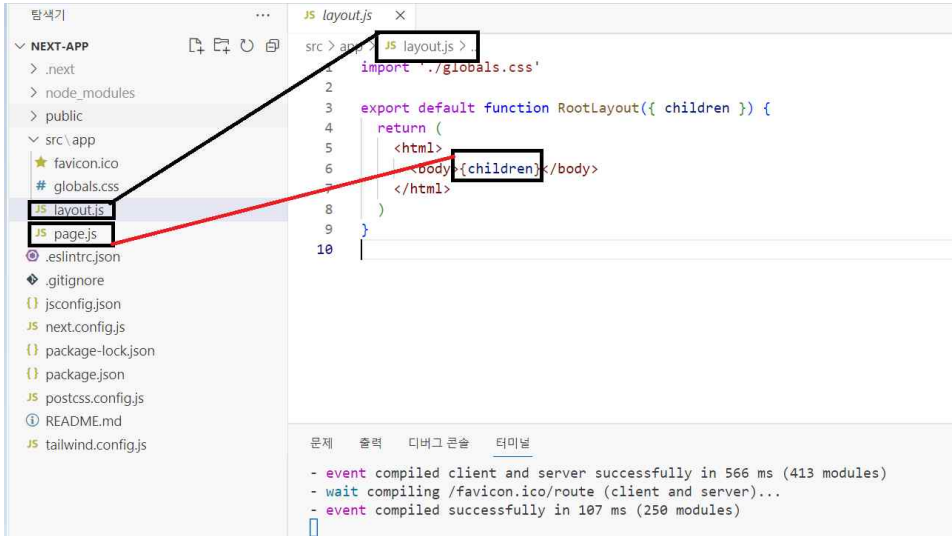
2. 깃허브 처리
- 터미널 없애 gitbash 선택 :
- git 원격저장소 확인 : `$ git remote -v` ;
 - git 원격주소 연결 : `$ git remote add origin 주소` ;
 - git 원격저장소 확인 : `$ git remote -v` ;

2. main => page 찾기 (무조건 page.js)

```
import './globals.css'

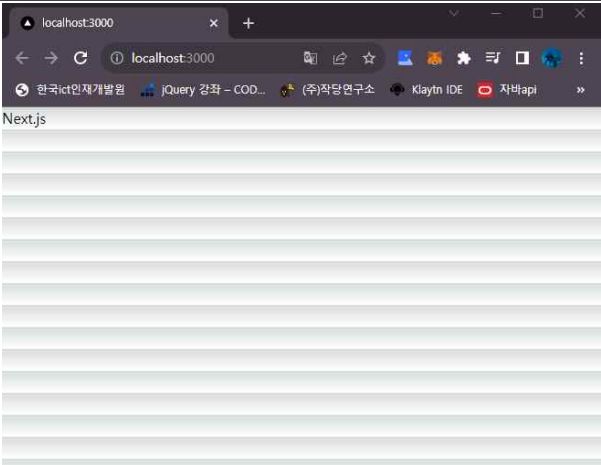
export const metadata = {
  title: 'Create Next App',
  description: 'Generated by create next app',
}

export default function RootLayout({ children }) {
  return (
    <html >
      <body>{children}</body>
    </html>
  )
}
```



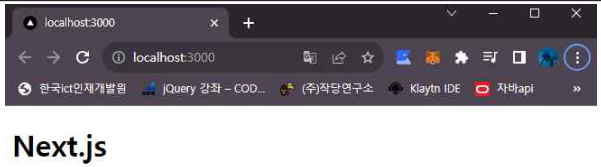
모든 내용을 지우고 입력한다.

```
export default function Home() {
  return (
    <div>
      <h1> Next.js </h1>
    </div>
  )
}
```



css가 아직 존재 한다.
layout.js 안에
import './globals.css' 가 있다.

globals.css 파일의 내용을 모두 삭제하자.



프로젝트 뼈대 만들기

<pre> /src/app/page.js 가 리턴하는 것이 화면에 보인다. page.js export default function Home() { return (<div> <h2>WelCome</h2> Hello, WEB! </div>) }</pre>	<pre> layout.js import './globals.css' export default function RootLayout({ children }) { return (<html> <body> <h1>WEB</h1> html css javascript { /* 위에서 링크된 내용은 모두 children에 들어와야 한다. */ {children} </body> </html>) } }</pre>
--	---

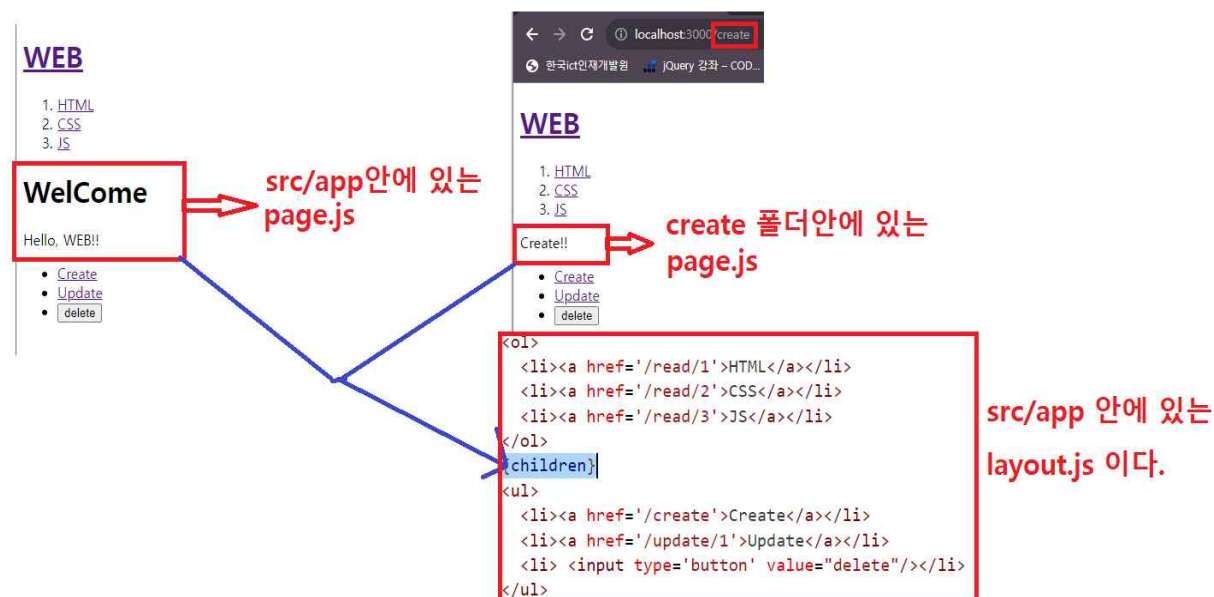
<pre> layout.js import './globals.css' export default function RootLayout({ children }) { return (<html> <body> <h1>WEB</h1> html css javascript { /* 위에서 링크된 내용은 모두 children에 들어와야 한다. */ {children} </body> </html>) } }</pre>	<pre> 내용 추가 하기 // title export const metadata = { title : 'Next.js App', description : 'Generated by nohssam73' } export default function RootLayout({ children }) { return (<html> <body> <h1>WEB</h1> html css javascript { /* 위에서 링크된 내용은 모두 children에 들어와야 한다. */ {children} Create Update <input type='button' value='delete' /> </body> </html>) } }</pre>
---	--

즉, 사용자가 요청한 URL에 맞는 페이지를 보여주는 것이라고 생각할 수 있다.

- 기존 웹페이지는 여러페이지를 사용, 새로운 페이지를 로드하는 방식
- SPA는 새로운 페이지를 로드하지 않고 하나의 페이지안에 필요한 데이터만 가져오는 형태

해당 create 밑에 page.js 는 create 밑에 layout.js 가 있으면
결합을 한다. 만약 없으며 그 위에 있는 부모폴더에서 layout.js를 찾아서 결합한다.

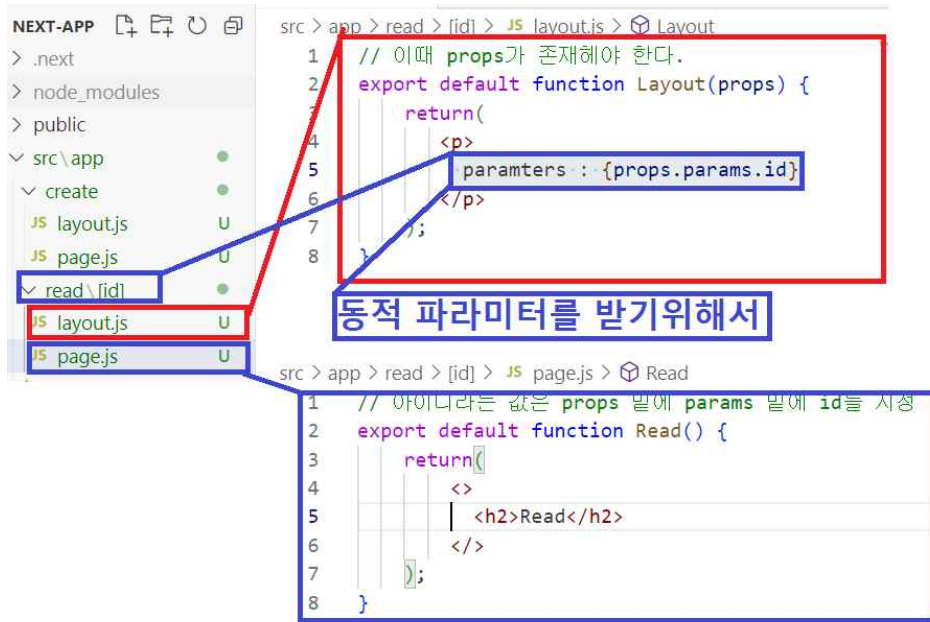
The screenshot shows the VS Code interface with the file explorer on the left and the code editor on the right. The file explorer shows the project structure with 'src/app' expanded, highlighting 'layout.js' and 'page.js'. The code editor shows the content of 'src/app/layout.js' and 'src/app/page.js'. The 'layout.js' code defines a 'Layout' component that takes 'props' and renders a form. The 'page.js' code defines a 'Create' component that renders a form. Arrows indicate the relationship between the file explorer and the code editor.



다이나믹 라우팅

```
<li><a href='/read/1'>html</a></li>
<li><a href='/read/2'>css</a></li>
<li><a href='/read/3'>javascript</a></li>
```

링크가 /read/1, /read/2, /read/3 처럼 /read/ 뒤에 정해지지않고
뭔가 올지 모를 때 다이나믹 라우팅으로 지정해다 된다.
일단 read 폴더를 만들고 아래 [id] 폴더를 더 만들어야 한다.



/read/1, /read/2, /read/3 일 때 처리를 다르게 하자.

```
export default function Layout(props) {
  let msg = null;
  if (props.params.id==='1' ) {
    msg = "HTML 선택";
  } else if (props.params.id==='2' ){
    msg = "CSS 선택"
  }else if (props.params.id==='3' ){
    msg = "JavaScript 선택"
  }
  return(
    <>
      {props.children}
      <h3>paramters : {props.params.id }</h3>
      <h3>{msg}</h3>
      <h3>paramters : {props.params.id !== '1' ? "HTML선택" : props.params.id === '2'? "CSS선택" : "JS선택" }</h3>
    </>
  );
}
```

정적 자원 사용하기 : 이미지 위치 : public 폴더에 위치 시키자.
파일을 읽어오자 : src/app/page.js에서 호출 하면 된다.

NEXT-APP

> .next

> node_modules

> public

- images
 - tree-1.jpgU
 - tree-2.jpgU
- next.svg
- vercel.svg

> src\app

- create
 - JS layout.js
 - JS page.jsM
- read\[id]
 - JS layout.js
 - JS page.js
- ★ favicon.ico
- # globals.css
- JS layout.jsM
- JS page.jsM

src > app > JS page.js > ...

1import Image from "next/image";

2import img1 from '/public/images/tree-2.jpg';

3export default function Home() {

4return (

5<div>

6<h2>WelCome</h2>

7Hello, WEB!

8{/* import를 따로 하지 않으면 width와 height를 반드시 입력해야 된다. */}

9<p><Image src="/images/tree-1.jpg" width={100} height={100}></Image></p>

10<p><Image src={img1} width={100} height={100}></Image></p>

11</div>

12)

13}

14

문제

출력

디버그 콘솔

터미널

a태그로 링크를 실행하는 내용을 개발자 도구에서 확인하면
클릭할때 마다 서버에가서 모든 정보를 다운받아서
속도와 비용이 증가 한다.
a 링크를 => Link 컴포넌트로 변경하자.
Link는 필요한 정보가 가져오고, 한번가져온 정보는 다시 가져오지 않는다.


Create nohsam

localhost:3000/create

1. HTML
2. CSS
3. JS

Create-layout

Create!!



Create

Update

delete

DevTools - localhost:3000/create

소스

요소

콘솔

네트워크

성능

메모리

애플리케이션

보안

필터

Fetch/XHR JS CSS 이미지 미디어 글꼴 문서 WS Wasm 매니페스트 기타

자단된 요청

타사 요청

이름

상태

유형

시작점

크기

시

목표

create

200

document

기타

2.7 kB

1.

layoutcss?v=169335...

200

stylesheet

create

1.2 kB

3.

layoutcss?v=169335...

200

stylesheet

create

1.2 kB

5.

webpackjs?v=16933...

200

script

create

10.8 kB

1.

main-app.js?v=1693...

200

script

create

1.4 MB

4.

faviconico

200

x-icon

image.js!1

9.6 kB

7.

image?url=%2F_next...

200

webp

create

(디스...

2.

app-pages-internalsjs

200

script

webpackjs?v=16...

37.8 kB

1.

pagejs

200

script

webpackjs?v=16...

165 kB

3.

webpack-hmr

101

websocket

use-socketjs...

0 B

다

요청 10건

1.7 MB 전송됨

7.0 MB 리소스

완료: 1.44초

DOMContentLoaded: 641밀리초

로드: 1.48초

콘솔

기본 수준

문맥 없음

DevTools failed to load source map: Could not load content for chrome-extension://fbldlloegpafnldhbmagocccfchci/sourcemaps/contentscript.js.map: 시스템 오류: net::ERR_BLOCKED_BY_CLIENT

DevTools failed to load source map: Could not load content for chrome-extension://fbldlloegpafnldhbmagocccfchci/sourcemaps/image.js.map: 시스템 오류: net::ERR_BLOCKED_BY_CLIENT

Download the React DevTools for a better development experience: <https://reactjs.org/link/react-dev-tools>

Image with src "/_next/static/media/tree-6.344d5ab6.jpg" was detected as the Largest Contentful Paint (LCP). Please add the "priority" property if this image is above the fold. Read more: <https://reactjs.org/docs/opt-reference/next/image-optimization>

Image is missing required "alt" property. Please add Alternative Text to `app-index.js:31` describe the image for screen readers and search engines.

Create nohsam

localhost:3000/read/1

1. HTML
2. CSS
3. JS

Read !!!

paramters : 3

JavaScript 선택

paramters : JS선택

Create

Update

delete

DevTools - localhost:3000/read/1

소스

요소

콘솔

네트워크

성능

메모리

애플리케이션

보안

필터

Fetch/XHR JS CSS 이미지 미디어 글꼴 문서 WS Wasm 매니페스트 기타

자단된 요청

타사 요청

이름

상태

유형

시작점

크기

시

목표

app-pages-internalsjs

200

script

webpackjs?v=16...

37.8 kB

9.

layoutjs

200

script

webpackjs?v=16...

66.8 kB

1.

faviconico

200

x-icon

기타

9.6 kB

5.

webpack-hmr

101

websocket

use-socketjs...

0 B

다

17_rsc=au68s

200

fetch

fetch-server-resp...

449 B

6.

27_rsc=au68s

200

fetch

fetch-server-resp...

474 B

6.

27_rsc=180U

200

fetch

fetch-server-resp...

1.1 kB

6.

37_rsc=mpszj

200

fetch

fetch-server-resp...

474 B

7.

37_rsc=9mqml

200

fetch

fetch-server-resp...

1.1 kB

5.

요청 14건

1.6 MB 전송됨

6.7 MB 리소스

완료: 13.23초

DOMContentLoaded: 705밀리초

로드: 1.53초

콘솔

기본 수준

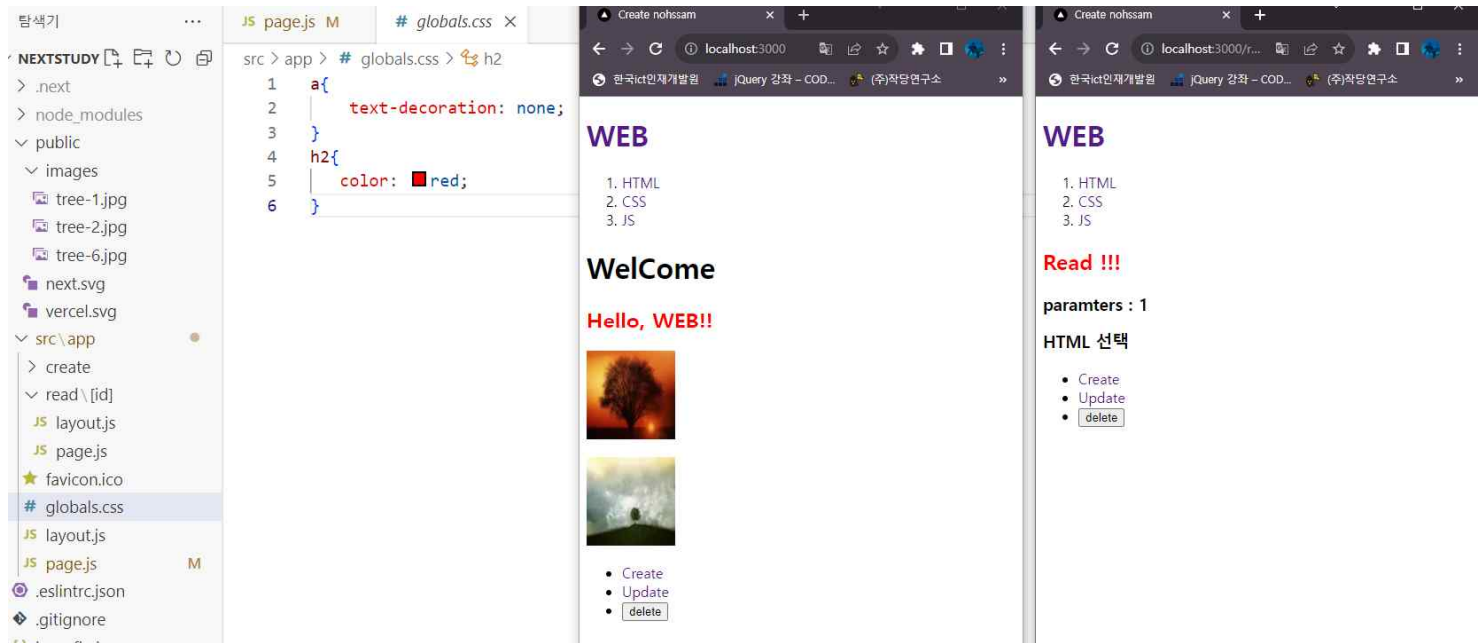
문맥 없음

DevTools failed to load source map: Could not load content for chrome-extension://fbldlloegpafnldhbmagocccfchci/sourcemaps/contentscript.js.map: 시스템 오류: net::ERR_BLOCKED_BY_CLIENT

DevTools failed to load source map: Could not load content for chrome-extension://fbldlloegpafnldhbmagocccfchci/sourcemaps/image.js.map: 시스템 오류: net::ERR_BLOCKED_BY_CLIENT

Download the React DevTools for a better development experience: <https://reactjs.org/link/react-dev-tools>

CSS 처리
src/app 밑에 layout.js를 보면 import './globals.css' 되어 있는 것을 볼 수 있다. (어떤 페이지를 방문하던 간에 적용된다.)



각 컴포넌트 CSS 처리 : 각 컴포넌트 폴더에 css를 만들면 적용된다.

NEXTSTUDY

> .next

> node_modules

> public

> src\app

> create

create.css

JS layout.js

JS page.js

> read\[id]

JS layout.js

JS page.js

★ favicon.ico

globals.css

JS layout.js

JS page.js

.eslintrc.json

.gitignore

jsconfig.json

JS next.config.js

package-lock.json

package.json

JS postcss.config.js

README.md

JS tailwind.config.js

개요

src > app > # globals.css > h2

1 a{

2 text-decoration: none;

3 }

4 h2{

5 color: red;

6 }

src > app > create > # create.css > h2

1 h2{

2 color: skyblue;

3 }

Create nohssam

localhost:3000

WEB


1. HTML


2. CSS

3. JS

WelCome

Hello, WEB!!





Create nohssam

localhost:3000/create

WEB


1. HTML

2. CSS

3. JS

Create-layout

Create!!



Create

Update

delete