

13. CSS 스타일링

현재 상황

지금까지 만든 홈페이지에는 스타일이 거의 없습니다.

이번 챕터에서는 Next.js 애플리케이션을 스타일링하는 다양한 방법을 배워봅니다.

이번 챕터에서 다루는 내용:

- 글로벌 CSS 파일 추가하기
- Tailwind CSS와 CSS Modules 두 가지 스타일링 방법
- `clsx` 유틸리티를 사용해 조건부 스타일 적용하기

글로벌 스타일(Global Styles)

`/app/ui` 폴더 안에 `global.css` 파일이 있습니다.

이 파일은 애플리케이션 전체에 적용할 수 있는 CSS 규칙(리셋, 링크 스타일 등)을 담을 수 있습니다.

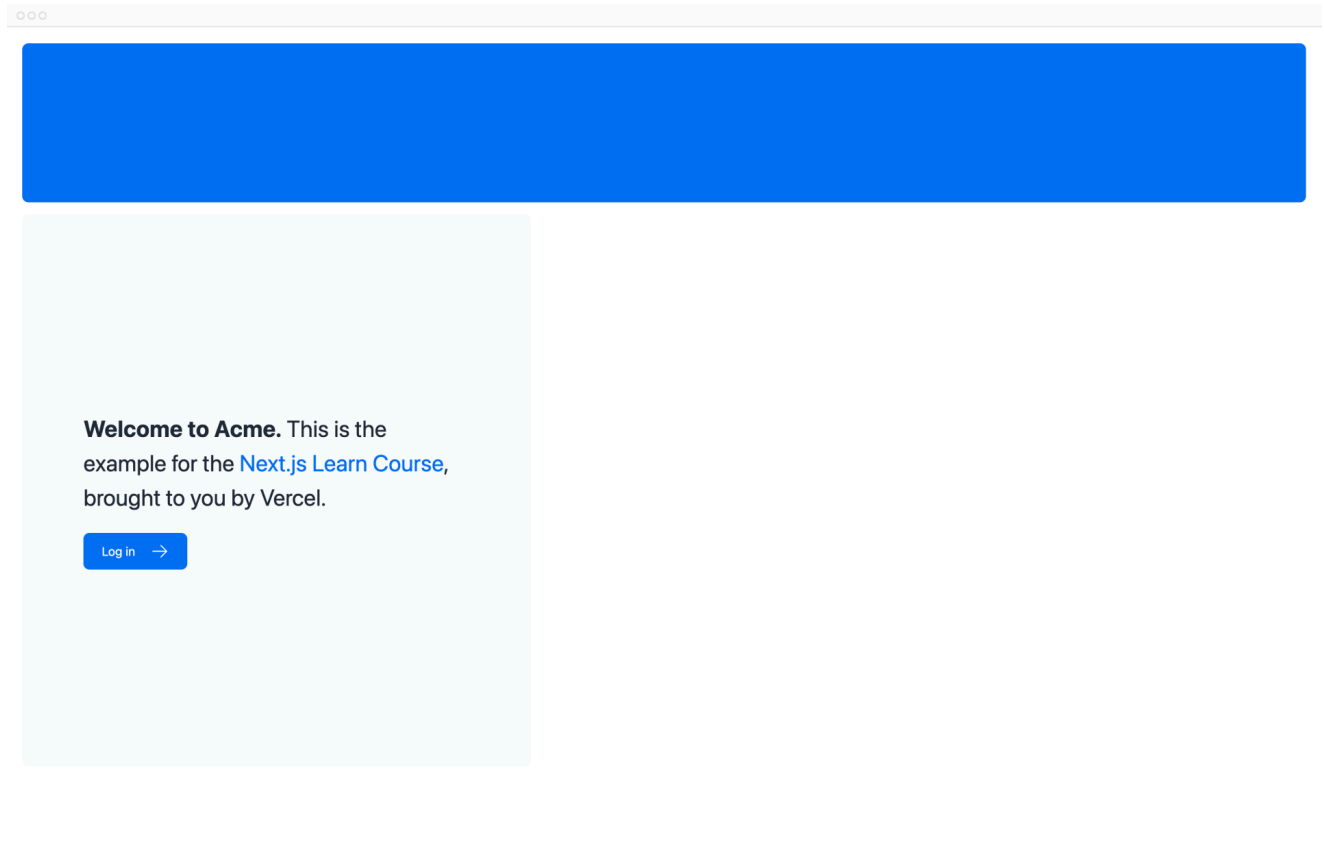
Next.js에서는 보통 최상위 컴포넌트(루트 레이아웃)에 글로벌 CSS를 import합니다.

`/app/layout.tsx`

```
import '@app/ui/global.css';

export default function RootLayout({ children }: { children:
React.ReactNode }) {
  return (
    <html lang="en">
      <body>{children}</body>
    </html>
  );
}
```

개발 서버를 켜 상태로 저장하면, 브라우저에 반영된 결과를 볼 수 있습니다.



그런데 우리가 별도로 CSS를 작성하지 않았는데 스타일이 입혀졌습니다. 왜일까요?

`global.css` 파일을 열어보면 다음과 같은 Tailwind 지시어(`@tailwind`)를 볼 수 있습니다:

```
@tailwind base;
@tailwind components;
@tailwind utilities;
```

이 덕분에 기본적인 Tailwind 스타일이 적용되고 있는 것입니다.

Tailwind CSS

[Tailwind CSS](#)는

React 코드 안에 직접 유틸리티 클래스를 작성해서 빠르게 스타일링할 수 있는 프레임워크입니다.

예를 들어:

```
<h1 className="text-blue-500">I'm blue!</h1>
```

- "text-blue-500" 클래스를 추가하면 텍스트 색이 파란색으로 바뀝니다.

Tailwind는 개별 요소에만 스타일을 적용하기 때문에:

- 요소를 추가/삭제해도 충돌 걱정이 없습니다.
- CSS 파일이 커지는 문제도 없습니다.

참고:

create-next-app 으로 Next.js 프로젝트를 만들 때 "Tailwind 사용할까요?"라는 질문에 Yes 를 선택하면 자동 설치됩니다.

예시

/app/page.tsx

```
import AcmeLogo from '@app/ui/acme-logo';
import { ArrowRightIcon } from '@heroicons/react/24/outline';
import Link from 'next/link';

export default function Page() {
  return (
    <main className="flex min-h-screen flex-col p-6">
      <div className="flex h-20 shrink-0 items-end rounded-lg bg-blue-500 p-4 md:h-52">
        { /* 내용 */ }
      </div>
    </main>
  );
}
```

Tailwind 클래스를 활용해 flex, padding, background color 등을 적용하고 있습니다.

Tailwind로 삼각형 만들어보기

/app/page.tsx 파일에서 <p> 태그 위에 아래 코드를 추가하세요:

```
<div
  className="relative w-0 h-0 border-l-[15px] border-r-[15px] border-b-[26px] border-l-transparent border-r-transparent border-b-black"
/>
```

→ 검은색 삼각형이 생성됩니다!

CSS Modules

Tailwind 대신 전통적인 CSS 파일을 선호한다면, **CSS Modules**를 사용할 수 있습니다.

CSS Modules는 컴포넌트 단위로 CSS를 "자동으로 범위 지정"해줍니다.
스타일 충돌 걱정이 없습니다.

1. `/app/ui` 폴더 안에 `home.module.css` 파일을 새로 만들고 다음 코드를 추가합니다:

```
.shape {  
  height: 0;  
  width: 0;  
  border-bottom: 30px solid black;  
  border-left: 20px solid transparent;  
  border-right: 20px solid transparent;  
}
```

2. `/app/page.tsx` 파일에서 import 하고 사용합니다:

```
import styles from '@app/ui/home.module.css';  
  
export default function Page() {  
  return (  
    <main className="flex min-h-screen flex-col p-6">  
      <div className={styles.shape} />  
      { /* ... */ }  
    </main>  
  );  
}
```

결과: Tailwind 버전과 똑같은 삼각형이 렌더링됩니다.

조건부 클래스 토글: **clsx** 라이브러리

상태(state)나 조건에 따라 클래스를 다르게 적용하고 싶을 때는
[clsx](#) 라이브러리를 사용할 수 있습니다.

예시: 송장 상태 표시 컴포넌트

`/app/ui/invoices/status.tsx`

```
import clsx from 'clsx';  
  
export default function InvoiceStatus({ status }: { status: string }) {  
  return (  
    <span  
      className={clsx(  

```

```
      'inline-flex items-center rounded-full px-2 py-1 text-sm',
      {
        'bg-gray-100 text-gray-500': status === 'pending',
        'bg-green-500 text-white': status === 'paid',
      }
    )}
  >
  { /* 텍스트 표시 */ }
</span>
);
}
```

"pending" 이면 회색, "paid" 면 초록색으로 스타일링됩니다.

기타 스타일링 방법

Next.js에서는 Tailwind, CSS Modules 외에도 다양한 방법으로 스타일링할 수 있습니다:

- Sass (`.scss` 파일)
- **CSS-in-JS** 라이브러리:
 - [styled-jsx](#) (Next.js 기본 내장)
 - [styled-components](#)
 - [emotion](#)

📖 자세한 내용은 [Next.js CSS 가이드](#) 참고
