

## 第 7 章 - 圖片

### Image 元件

Next 提供了 Image 元件，這是擴充自原本<img>標記的專用於網站圖片的元件，使用的主要目的如下：

- 尺寸大小的最佳化: 針對不同的觀看裝置，自動地提供正確的尺寸大小，使用現代新式的圖檔格式，例如 WebP 與 AVIF
- 視覺上的穩定性: 當圖片載入時，自動地避免排版偏移
- 更快的頁面載入: 圖片只有當它們進入視區時，才會使用瀏覽器原生的延時載入(lazy loading)，並可加上可選的模糊佔位區塊
- 資產靈活性: 按需求使用圖片重訂尺寸大小，也可以針對儲存在遠端伺服器的圖片

註: Cumulative Layout Shift(CLS，累積排版偏移)評量與使用者體驗或網站最佳化有關，更多參考:  
[Cumulative Layout Shift \(CLS\)](#)與[Cumulative Layout Shift \(CLS\)](#)

### 定義

```
// 導入模組
import Image from 'next/image'
//...
// 使用元件
const logo = <Image src="/logo.png" alt="" width={200} height={100} />
```

### 本地端圖片

可以使用 .jpg, .png, .webp，導入後使用，由 Next 自動地決定寬度(width)與高度(height)，或自行定義寬度(width)與高度(height)：

```
import Image from 'next/image'
import profilePic from './me.png'

export default function Page() {
  return (
    <Image
      src={profilePic}
      alt="Picture of the author"
      // width={500} 自動提供
      // height={500} 自動提供
      // blurDataURL="data:..." 自動提供
      // placeholder="blur" // 可選的在載入時使用模糊化(blur-up)後呈現
    />
  )
}
```

### 遠端圖片

寬度(width)與高度(height)是必要的屬性，另外`blurDataURL`也需要手動地加上。

寬度(width)與高度(height)的數值並沒辦法真正決定實際呈現的圖片大小，而是決定了呈現的寬高比例為主。

範例:

```
import Image from 'next/image'

export default function Page() {
  return (
    <Image
      src="https://s3.amazonaws.com/my-bucket/profile.png"
      alt="Picture of the author"
      width={500}
      height={500}
    />
  )
}
```

使用 Image 元件載入時，不能隨意外連到遠端伺服器，要外連圖片需要在 `next.config.js` 中加入 `images` 區段關於網域的設定:

```
images: {
  domains: ['via.placeholder.com', 'localhost'],
}
```

註: 另一種設定方式是使用`remotePatterns`設定區段，它的設定會較詳細和複雜。

## SVG 圖片

第 1 種方式 - 放於 `public` 資料夾中直接使用即可，下面這圖檔位於`public/vercel.svg`:

註: 此種方式無法用屬性值或套用 CSS 類別來更動 SVG 中的樣式(例如顏色等等)，最多改變大小

```
<Image src="/vercel.svg" alt="Vercel Logo" width={100} height={24}
priority />
```

第 2 種方式 - 將需要套用的 `svg` 圖片寫為一個 React 元件:

註: 因為會有標記格式和屬性的差異，可以先用例如`svg2jsx`或`svgviewer`先轉為合法 JSX 語法

```
export default function ArrowLeftIcon({ className }) {
  return (
    <svg
      xmlns="http://www.w3.org/2000/svg"

```

```
    viewBox="0 0 24 24"  
    fill="currentColor"  
    shapeRendering="geometricPrecision"  
    className={className}  
  >  
    <path d="M19 12H5" />  
    <path d="M12 19L5 12L12 5" />  
  </svg>  
)  
}
```

之後在 React 元件中使用:

```
import ArrowLeftIcon from 'components/icons/arrow-left'  
// ...  
return <ArrowLeftIcon className="h-6" />
```

## 注意事項

- Image 元件的 className 無法使用[styled-jsx](#)，建議使用 CSS Module 或 global css

## 更多參考資訊

### 進階搭配使用套件

- [next-images](#)
- [next-optimized-images](#)
- [SVGR](#)

### 圖片最佳化

更多有關這個元件提供的自動圖片最佳化(Automatic Image Optimization)功能，可以參考此[官網文件](#)。

### 最佳化圖檔準備

- 當使用動態與相當大的圖片時，考慮使用 CDN 例如 [Cloudinary](#), [Cloudflare](#) 來儲放與提供