

【Next.js】Layout

太好了！以下是為「Next.js 初學者」量身打造的教學文章，主題為：

□ Next.js App Router 中 layout.tsx、page.tsx、children 的邏輯與實作

□ 為什麼要了解這三個東西？

在 Next.js 的 App Router 架構中，整個畫面的內容是由許多層「外殼 (layout)」與「內頁 (page)」所組合而成。而這個組合的核心，就是這三個角色：

- `layout.tsx`：整體框架的「容器」
- `page.tsx`：實際對應到網址的「頁面內容」
- `children`：layout 內部接收的下一層內容（可能是 page，也可能是下一層 layout）

□ 這三者的關係是什麼？

你可以想像畫面組成就像是便當盒：

- `layout.tsx` 就是便當盒（框架）
- `page.tsx` 是便當裡的菜色（內容）
- `children` 是 `layout.tsx` 中的佔位符，用來「放內容的地方」

□ 換句話說：

“ `page.tsx` 的畫面內容，會自動傳給上層 `layout.tsx` 中的 `children` 來顯示

□ 結構範例

```
app/
├─ layout.tsx    ← 全站框架 Layout
├─ page.tsx     ← 首頁 ` / `
├─ about/
└─   └─ page.tsx ← ` /about ` 頁面
```

□ layout.tsx 實際程式碼

```
export default function RootLayout({ children }: { children: React.ReactNode }) {
  return (
    <html lang="en">
      <body>
        <header>這是全站共用的 Header</header>
        <main>{children}</main> { /* 這裡會被 page.tsx 的內容填進來 */ }
        <footer>這是共用的 Footer</footer>
      </body>
    </html>
  )
}
```

```
</body>
</html>
);
}
```

□ page.tsx 實際程式碼

```
export default function HomePage() {
  return <div>這是首頁的內容</div>;
}
```

□ 瀏覽 / 時會發生什麼事？

1. Next.js 根據 URL 找到 `app/page.tsx`
2. 找到對應的 `layout.tsx` 作為頁面外框
3. 把 `page.tsx` 的內容當作 `children` 傳給 `layout.tsx`
4. 畫面最終呈現：

```
<html lang="en">
  <body>
    <header>這是全站共用的 Header</header>
    <main>
      <div>這是首頁的內容</div>
    </main>
    <footer>這是共用的 Footer</footer>
  </body>
</html>
```

□ 多層 layout 怎麼辦？

可以有巢狀 layout 結構，例如：

```
app/
├── layout.tsx      ← 全站 layout
├── (admin)/
│   ├── layout.tsx ← admin 專區 layout
│   └── dashboard/
│       └── page.tsx ← `dashboard` 頁面
```

流程如下：

1. `dashboard/page.tsx` 被載入
2. 它被傳給 `admin/layout.tsx` 的 `children`
3. 然後再傳到最外層 `app/layout.tsx` 的 `children`
4. 最終包成一整頁畫面

□ 重點整理

元素	說明
<code>layout.tsx</code>	定義頁面外框、接收 <code>children</code>
<code>page.tsx</code>	對應 URL 的實際頁面內容
<code>children</code>	<code>layout.tsx</code> 的佔位符，用來顯示下層內容
傳遞邏輯順序	<code>page.tsx</code> → 傳給 → <code>layout.tsx</code> → 再往上層

□ 初學者常見疑問

□ Q1: `children` 是誰傳給 `layout.tsx` 的？

A: 是 Next.js 自動傳進來的，不需要自己傳。

□ Q2: 可以有多個 `layout.tsx` 嗎？

A: 可以。每一層目錄都可以定義自己的 `layout.tsx`，形成巢狀包裝結構。

□ Q3: 如果我不寫 `layout.tsx` 會怎樣？

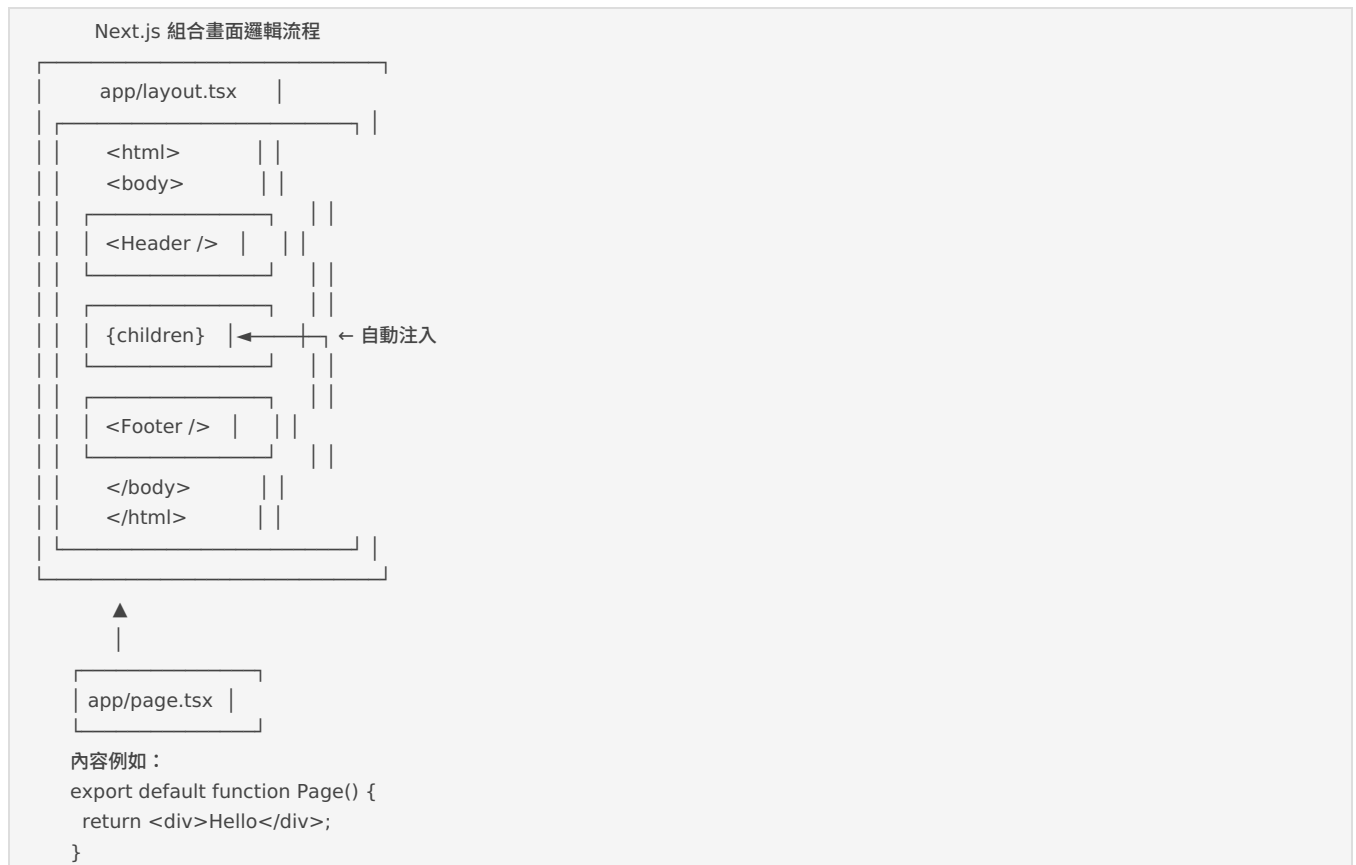
A: 畫面會照樣顯示，但你就無法包共用的元件（像是 Header、Footer、Provider）。

□ 建議實作順序

1. 建立 `app/layout.tsx`：放 header/footer
2. 建立 `app/page.tsx`：顯示首頁
3. 建立 `app/about/page.tsx`：新增新頁面
4. 觀察 `children` 如何把 page 內容塞進 layout

以下是這篇教學的視覺化圖解，讓初學者更清楚了解 `layout.tsx`、`page.tsx`、`children` 在 Next.js App Router 中是如何串接的。

□ 圖解：`layout.tsx` 如何包住 `page.tsx`



□ 圖解說明：

- `layout.tsx` 是整個頁面的外框（包含 `<html>`，`<body>`，Header, Footer）
- `{children}` 是一個「插槽」，Next.js 會自動把 `page.tsx` 的內容放進去
- 你不需要自己寫 props 傳入 `children`，Next.js 幫你處理好
- 如果你有巢狀 layout，每一層都是這樣一層一層包下去

個頁面不要套用全站的 `layout.tsx`，該怎麼做？

這是一個非常常見的情境：

“「在 Next.js App Router 中，某個頁面不要套用全站的 `layout.tsx`，該怎麼做？」

□ 簡短結論

在 App Router 中：

“□ 無法直接讓某個頁面「跳過」`app/layout.tsx`，因為它是全域的根 layout，每一頁都會套用。

但有兩個實務解法可以達到「看起來沒有 layout」的效果：

□ 解法 1：用 Route Group 分離 layout

<https://bookstack.treemanou.com/books/treemanreact/page/nextjsmultiple-root-layouts>

□ 結構：

```
app/
├── layout.tsx           ← 全站 layout (預設套用)
├── (no-layout)/        ← □ 特殊：不繼承全站 layout
│   ├── login/
│   │   ├── layout.tsx ← 空 layout
│   │   └── page.tsx   ← `login`
```

□ `(no-layout)/layout.tsx`

```
export default function NoLayout({ children }: { children: React.ReactNode }) {
  return children; // □ 直接回傳 children，不包任何框架
}
```

“□ 這樣 `/login` 就不會套到 `app/layout.tsx` 了
□ 它只會用 `(no-layout)/layout.tsx`，你可以視為「不使用 layout」

□ 解法 2：在 `layout.tsx` 裡判斷 route 隱藏部

分元件

如果你不想拆資料夾，也可以：

```
'use client'

import { usePathname } from 'next/navigation'
import Header from '@components/Header'
import Footer from '@components/Footer'

export default function RootLayout({ children }: { children: React.ReactNode }) {
  const pathname = usePathname()
  const isPlainPage = pathname.startsWith('/login') || pathname.startsWith('/print')

  return (
    <html>
      <body>
        {!isPlainPage && <Header />}
        <main>{children}</main>
        {!isPlainPage && <Footer />}
      </body>
    </html>
  )
}
```

“ `/login` 和 `/print` 等頁面就會不顯示 Header / Footer
 但 layout 本身還是存在，只是條件渲染內容

比較小結

方式	優點	缺點
<input type="checkbox"/> 用 Route Group 拆 layout	完全不繼承任何 layout	需要多一層目錄
<input type="checkbox"/> layout 判斷條件顯示元件	不需要改目錄結構	layout 實際還是存在，略為複雜

建議

使用情境	建議方式
登入頁、列印頁等需要極簡畫面	<input type="checkbox"/> 用 Route Group 分離 layout
特定頁只想隱藏 header/footer 等元件	<input type="checkbox"/> layout 內用 <code>usePathname()</code> 判斷

如果你需要範例專案或要把這部分整合進教學文章，我可以幫你補上 Markdown + 圖解，要嗎？

🕒 修訂版本 #3

★ 由 treeman 建立於 25 📅 2025 15:50:26

✍ 由 treeman 更新於 25 📅 2025 16:37:32