(CHAPTER)

5

利用 HTML+CSS 製作基本版型

認識完基本 HTML 及 CSS 後,接下來我們來學習如何製作基本版型吧!基本的網頁版型包含單欄式版型、雙欄式版型,以及登入版型,這三個版面都是最簡單的!現在就讓我們透過製作基礎版型熟練整體網頁的架構,並從中學習 HTML 元素與 CSS 吧!

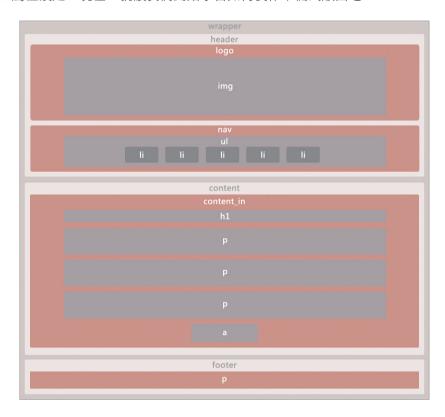
學習重點

單欄式版型單欄式響應式版面雙欄式版型登入版型



單欄式版面

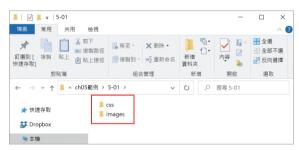
單欄式版型意味著只有一個欄位。此版型的架構相當簡單,其網頁的內容只需要由上往下堆疊即可。在此範例中,僅有 nav 元素的部分較為複雜,需透過較多的 css 屬性設定。現在,就讓我們開始學習如何製作單欄式版面吧!





5-1-1 前置作業

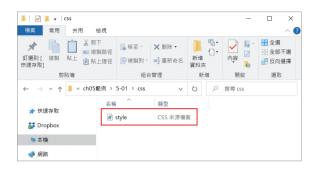
step 新增專案資料夾 5-01,並於專案資料夾下新增 css 與 images 資料夾。



新增 index.html,放於專案資料夾下。



step 新增 style.css,放於專案資料夾的 css 資料夾中。



複製範例中 CH5/5-01/images 資料夾下的所有圖片,放於專案資料夾的 images 資料夾下。



step 開啟 index.html,套用 style.css。

[index.html]

- · 「*」為通用選擇器,意思為選擇每個元素。而我們為了解決不同瀏覽器中每個元素的 margin 和 padding 預設值不同的問題,因此透過通用選擇器,將所有元素的 margin (外距) 與 padding (內距) 皆設為 0px。
- · font-family: Microsoft JhengHei 用以設定文字字型為微軟正黑體。
- · margin: 0px auto 用以設定元素於父元素中水平置中。在此,由於 wrapper 為最外層包覆網頁的 div 元素,而我們為了使 wrapper 於視窗中水平置中,因此設定 wrapper 的上下外距設定為 0px,左右外距為 auto。
- 於 wrapper 元素中新增 header 元素、id 為 content 的 div 元素以及 footer 元素。

[index.html]

```
<div id="wrapper">
    <header></header>
    <divid="content"></div>
    <footer></footer>
</div>
```

NOTE

- · header 元素內主要放置橫幅圖片與網頁導覽列。
- · content 內主要放置網頁的內容文字。
- · footer 元素內主要放置網頁版權聲明。

5-1-3 製作 header 區塊

[index.html]

ostep 05 設定 nav 元素的樣式。

[style.css]

```
nav{
   background-color: #B63D32;
   text-align: center;
}
```

NOTE

- #B63D32 是 酒 紅 色 的 顏 色 代 碼。 在 此 , 我 們 設 定 background-color:#B63D32 , 使得 nav 元素的背景顏色為酒紅色。
- text-align 屬性用於設定內容文字的對齊方式,在此我們設定 nav 元素的內容文字置中對齊。

 $\stackrel{\text{step}}{\text{06}}$ 於 nav 元素內,新增 ul 與 li 元素,並在每個 li 元素之中加入一個 a 元素。

[index.html]

step 07 設定 ul 與 li 元素的樣式。

[style.css]

```
ul{
    display: inline-block;
    margin: 0px;
}
li{
    list-style-type: none;
    float: left;
    text-align: center;
}
```

```
li a{
    display: block;
    padding: 15px 20px;
    color: #FFFFFF;
    text-decoration: none;
}
li a:hover{
    color: #FFC90E;
}
```

- · display 屬性用於設定元素的區框類型,最常被設定的值有 block 與 inline。 設定為 block 的元素稱為「區塊元素」,而設定為 inline 的元素稱為「行內元素」。inline-block 與 inline 相似,但是卻可以設定元素的 width 與 height 屬性。
- · 由於 ul 預設的 display 屬性為 block,在此我們應該將 display 屬性設定為 inline-block,如此才可以讓 ul 元素於 nav 元素中水平置中。
- · list-style-type 屬性用於設定項目清單的樣式。在此,我們設定項目清單的樣式為無樣式 (none)。
- · hover 用以設定滑鼠滑過連結時的效果,在這裡我們設定 li 元素中的 a 元素,文字顏色變成 #FFC90E (黃色)。

按下儲存並於瀏覽器中開啟「index.html」,檢視 header 的製作。



5-1-4 製作 content 區塊

[index.html]

```
<header>
 <div id="logo">
   <img src="images/banner.jpg" />
 </div>
 <nav>
   <a href="#">首頁 </a>
   <a href="#"> 最新消息 </a>
   <a href="#">合作店家 </a>
   <a href="#"> 關於我們 </a>
   <a href="#"> 聯絡我們 </a>
 </nav >
</header>
<div id="content">
   <div id="content_in">
   </div>
</div>
```

NOTE

content_in 用於包覆 content 中的內容文字。我們透過 content_in 調整文字 內容與 content 之間的間隙。

step | 設定 content_in 的樣式。

[style.css]

```
#content_in{
   padding: 30px 50px;
   text-align: center;
   background-color: #F4F4ED;
}
```

padding: 30px 50px 用以設定上下內距為 30px,左右內距為 50px。透過如此設定,可讓 $content_in$ 與 $content_in$ 中的元素或是內容文字之間具有間隙。此外,padding 的設定值並非絕對,所以使用者可依照自己的喜好,設定 $content_in$ 的 padding 值。

step | 於 content_in 中,新增 h1 元素並輸入「歡迎光臨」。

[index.html]

```
<div id="content">
    <div id="content_in">
    <h1> 歡迎光臨 </h1>
    </div>
</div>
```

otep 04 設定 h1 元素的樣式。

[style.css]

```
#content h1{
    color: #B63D32;
    margin-bottom: 10px;
}
```

NOTE

- · color: #B63D32 用於設定 h1 元素的文字顏色為酒紅色。
- margin-bottom屬性用於設定元素的下外距。在此,我們將 h1 元素的下外距設定為 10px,用以使得 h1 元素下方的元素與 h1 元素之間有 10px 的間隙。

[index.html]

step | 設定 p 元素的樣式。 **06** |

[style.css]

```
#content_in p{
   font-size: 15px;
   margin-bottom: 20px;
}
```

NOTE

- · font-size 屬性用於設定文字的大小。
- · margin-bottom 屬性用於設定 p 元素的下外距。

^{step} │ 於各個 p 元素內輸入內容文字。

[index.html]

step 08

於p元素下方,新增a元素。

[index.html]

ostep 09

設定 a 元素的樣式。

[style.css]

```
#content_in a{
    display: block;
    width: 130px;
    height: 40px;
    line-height: 40px;
    margin: 0 auto;
    text-decoration: none;
    background-color: #B63D32;
    color: #FFFFFF;
}
```

NOTE

- · 由於 a 元素預設之 display 為 inline,因此 a 元素的高度與寬度都無法設定。為了設定 a 元素的高度與寬度,我們將 display 設定為 block (區塊),用以設定 a 元素的寬度為 130px,高度為 40px。
- · 此外,我們希望 a 元素的內容文字於 a 元素中垂直置中,因此我們將行高 (line-height) 設定成與 a 元素的高度 (height) 相同,即皆設定為 40px,以 達成內容文字於 a 元素中垂直置中的效果。
- text-decoration 屬性用於設定文字的特效。在此,我們設定 text-decoration 為不顯示任何文字特效 (none)。

按下儲存並於瀏覽器中開啟「index.html」,檢視 content 區塊的製作。



5-1-5 製作 footer 的區塊

step 於 footer 元素中新增一個 p 元素。

[index.html]

```
<footer>
    Copyright @ 2017 Itoeat All rights reserved.
</footer>
```

設定 footer 元素的樣式。 02

[style.css]

```
footer{
   background-color: #71552A;
   text-align: center;
}
footer p{
   font-size: 10px;
   color: #FFFFFF;
```

```
padding: 10px 0px;
}
```

padding: 10px 0px 用以設定上下內距為 10px,左右內距為 0px,透過如此設定,可讓 p 元素與其內容文字之間的上下間距皆為 10px。

按下儲存並於瀏覽器中開啟「index.html」,檢視 footer 元素的製作。





使用 GitHub Copilot 快速製作單欄式 響應式版面

接下來,我們使用 GitHub Copilot 快速製作單欄式響應式版面,透過文字的方式 讓 GitHub Copilot 生成相似的範例!

GitHub Copilot 的操作輸出不一定會跟本書相同,煩請讀者在實際操作時以實際產生的程式碼為主。讀者可以自行更改提詞或產生的程式碼,更能貼近讀者需求。

step 新增專案資料夾 5-01-01 01

資料夾建立方式如 5-1-1 前置作業,只是我們不需要 index.html 及 style. css 檔,images 資料夾需有 banner 圖片,接下來,開啟 Visual Studio Code,將 5-01-01 資料夾開啟。



step 開始對話 02

接著,參考 5-1 單欄式版面的配置。由於我們還未在工作區建立 html 及 css 檔案,於聊天視窗中,輸入「@workspace 請設計一個寬度為 1024px 的單欄式響應式網頁,包含 index.html 和對應的 style.css。網頁主體使用 wrapper 包覆,其中依序包含 header、content 和 footer。

header 裡有 logo 區塊和 nav 區塊上下兩個區塊,logo 區塊需放置 banner.jpg 圖片; nav 區塊需包含「首頁」、「最新消息」、「合作店家」、「關於我們」和「聯絡我們」五個選單項目; content 要顯示「歡迎光臨」文字以及一個「了解更多」按鈕; footer 則需顯示今年的版權宣告,頁面內容不足時仍然位於頁面底部。

在 style.css 檔案,整體網頁背景色為 #F4F4ED,header 裡 nav 區塊和了解更多按鈕的背景色為 #B63D32,footer 的背景色為 #71552A。另外,所有的文字都需要置中對齊。」按下 Enter 鍵後,GitHub Copilot 會產生 html 及 css 檔案。

03 點選「更多操作」

```
</>
⟨⟨⟩ Workspace
✓ 正在收集工作區資訊
好的,以下是符合您需求的 index.html 和 style.css 檔
案內容:
index.html
<!DOCTYPE html>
<html lang="zh-tw">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=d</pre>
    <title>單欄式響應式網頁</title>
    <link rel="stylesheet" href="css/style</pre>
</head>
<body>
    <div class="wrapper">
        <header>
            <img src="images/banner.jpg" a</pre>
        </header>
        <nav>
            <l
                <a href="#">首頁</a></!
                <a href="#">最新消息</a
                <a href="#">合作店家</a>
                <a href="#">關於我們</a
                <a href="#">聯絡我們</a
            </nav>
         div class="content"s
```

104 點選「插入至新檔案」

儲存成 index.html 檔案。css 檔案也是相同操作,並儲存為 style.css,儲存至 css 資料夾裡。



將 index.html 開啟,大致上就完成囉!細節都可以按照 5-1 的教學去做調整。



NOTE

雖然 GitHub Copilot 讓生成網頁的步驟變得簡單,但讀者還是要對程式碼有基礎的認識,因為 GitHub Copilot 生成的程式碼不一定準確也不一定能符合讀者 想要的效果,如果有程式碼的基礎,讀者就可以自行輸入想要的需求,自行修改程式,所以建議讀者不要太依賴 GitHub Copilot,還是要自行判斷!