

本堂課學習提要

- 認識 HTML 與 CSS
- 區分內部樣式表與外部樣式表
- 利用 Div 標籤規劃網頁區塊
- 製作層層包覆的 Div 標籤
- 利用 CSS 改變 Div 標籤的外觀及大小、位置

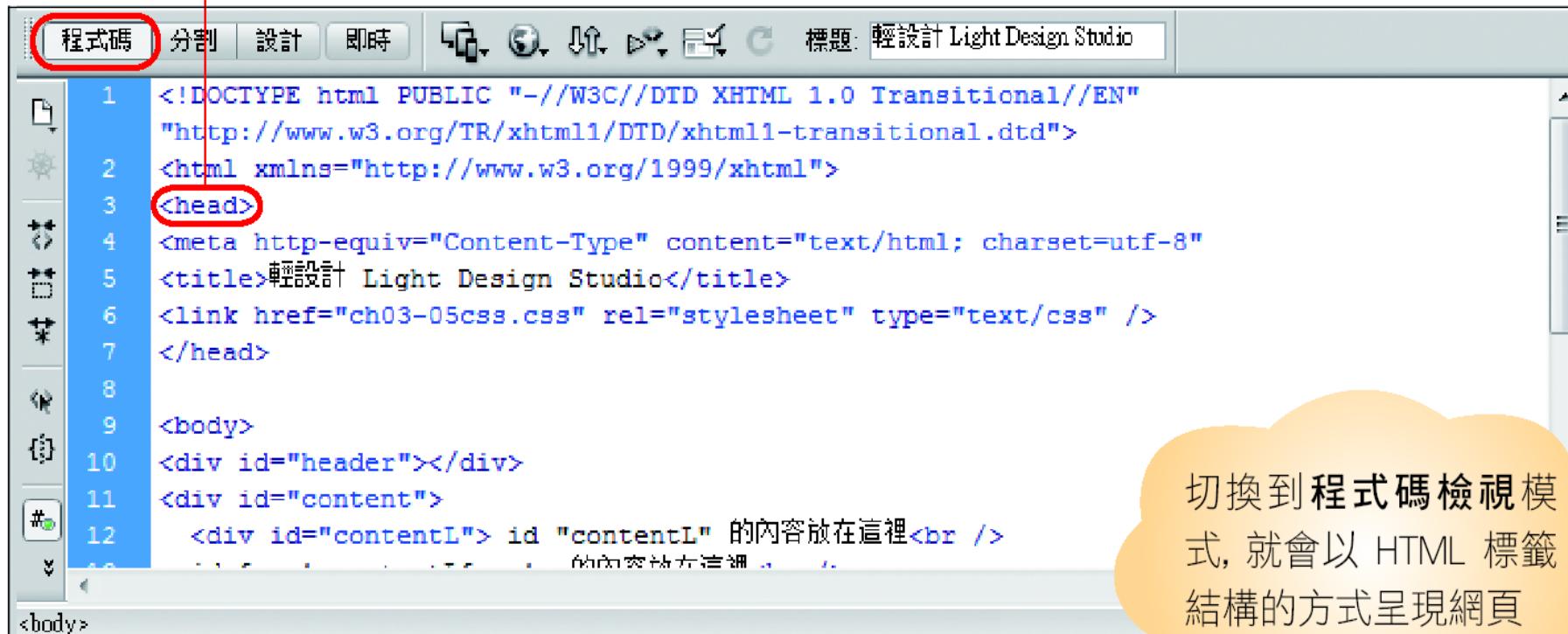
3-1 認識 HTML 與 CSS

- 認識 HTML
- 認識 CSS
- HTML 與 CSS 的版本選擇：認識 HTML 5 與 CSS 3.0

認識 HTML

標準：W3C，1999公布HTML 4.01版

由 "<" 與 ">" 包在一起的就稱為標籤



The screenshot shows a web editor interface with the following details:

- Toolbar:** Includes tabs for "程式碼" (Code), "分割" (Split), "設計" (Design), and "即時" (Live). The "程式碼" tab is highlighted with a red circle.
- Title Bar:** Displays the title "輕設計 Light Design Studio".
- Code View:** Shows the HTML code in a syntax-highlighted format. The code includes a DOCTYPE declaration, an XML namespace declaration, and a head section containing a meta tag and a title. The head tag is also highlighted with a red circle.
- Content Area:** Shows the body section with a header and content divs. A callout bubble points to the content area with the text: "切換到 程式碼 檢視模式，就會以 HTML 標籤結構的方式呈現網頁".

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
<title>輕設計 Light Design Studio</title>
<link href="ch03-05css.css" rel="stylesheet" type="text/css" />
</head>

<body>
<div id="header"></div>
<div id="content">
    <div id="contentL"> id "contentL" 的內容放在這裡<br />
        . . .
    </div>
</div>
</body>
```

切換到 **程式碼** 檢視模式，就會以 HTML 標籤結構的方式呈現網頁

認識 HTML

step01

Untitled-1 ×

程式碼 **分割** 設計 即時

標題: 無標題文件

```
1 <!DOCTYPE html PUBLIC
2   "-//W3C//DTD XHTML 1.0
3   Transitional//EN"
4   "http://www.w3.org/TR/xhtml1/DTD/
5  /xhtml1-transitional.dtd">
6 <html xmlns=
7   "http://www.w3.org/1999/xhtml">
8 <head>
9 <meta http-equiv="Content-Type"
10 content="text/html;
11 charset=utf-8" />
<title>無標題文件</title>
</head>
<body>
</body>
</html>
```

<body> 標籤組之間
專門用來放置網頁
內容，也就是瀏覽
器中會看到的畫面

這兩個插入點會同步變化

認識 CSS

標準：CSS 2.1 版



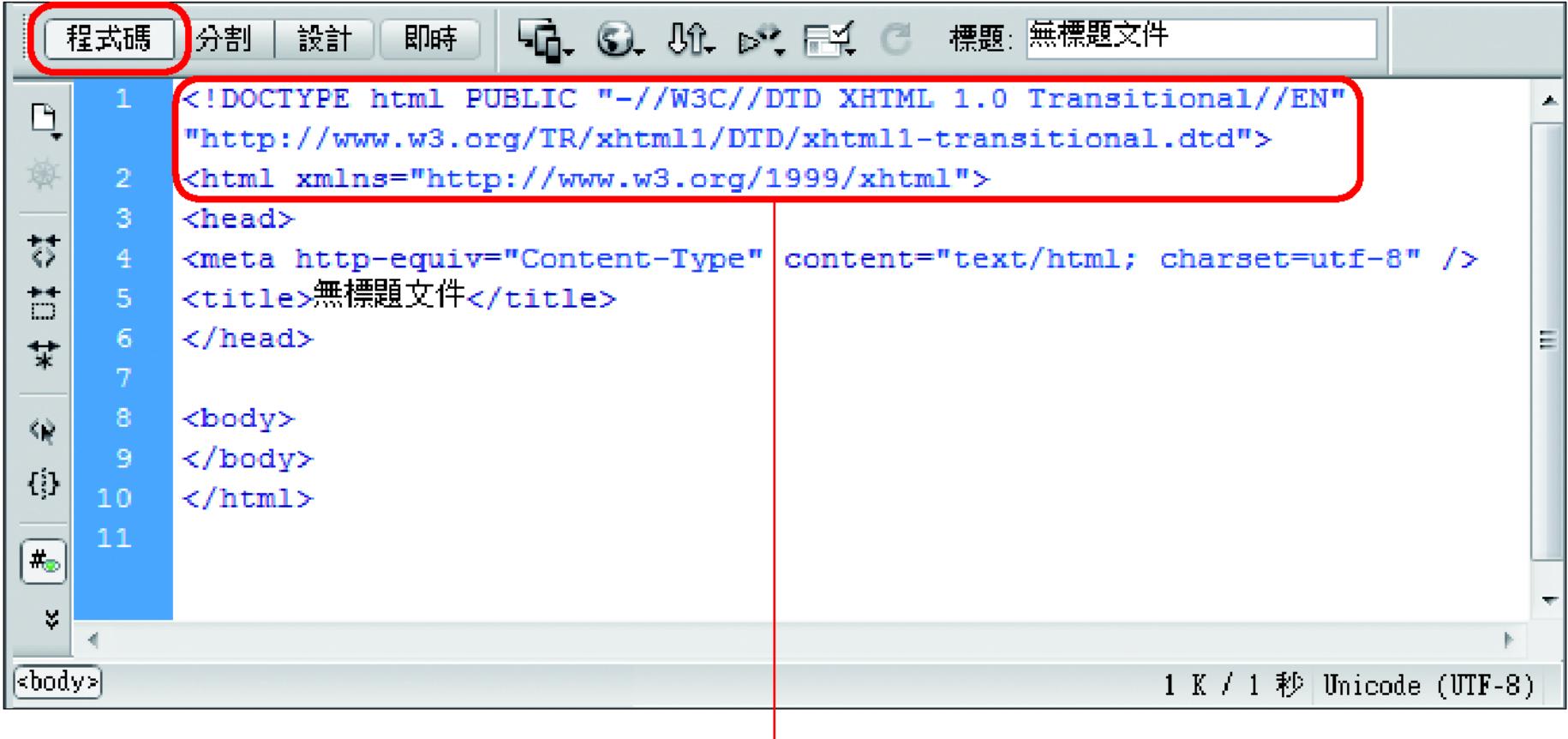
單純的 HTML 標籤架構網頁，
沒有 CSS 的輔助，樣式變化較少



設定 CSS 樣式後，網頁編排更具彈性，例如
兩欄式版面、不同顏色與大小的文字、…等

HTML 與 CSS 的版本選擇： 認識 HTML 5 與 CSS 3.0

HTML5與CSS3於2014在W3C確定成為最新標準。



```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
3 <html xmlns="http://www.w3.org/1999/xhtml">  
4 <head>  
5 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
6 <title>無標題文件</title>  
7 </head>  
8 <body>  
9 </body>  
10 </html>  
11
```

<body>

1 K / 1 秒 Unicode (UTF-8)

以預設值建立的網頁會加上這 3 行，代表本網頁使用 1999 年推出的 HTML 4.01 版

HTML 與 CSS 的版本選擇： 認識 HTML 5 與 CSS 3.0

支援 HTML 5 語法的瀏覽器版本 (含更新版本)



Internet Explorer



Mozilla Firefox



Google Chrome



Apple Safari



Opera

IE 9

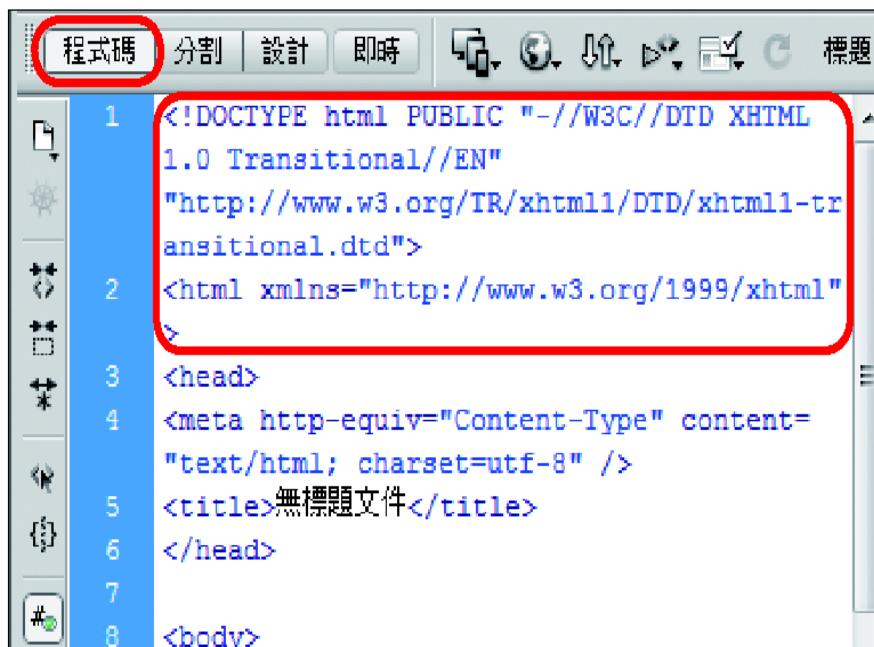
Firefox 3.6

Chrome 12

Safari 4

Opera 10

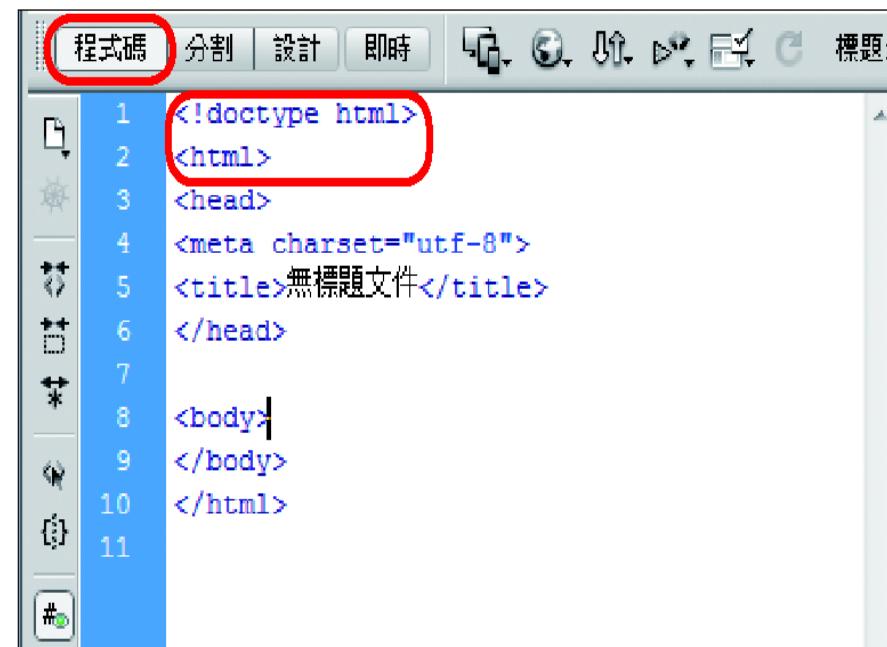
HTML 與 CSS 的版本選擇： 認識 HTML 5 與 CSS 3.0



程式碼 | 分割 | 設計 | 即時 | 標題:

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML  
1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-tr  
ansitional.dtd">  
2 <html xmlns="http://www.w3.org/1999/xhtml"  
>  
3 <head>  
4 <meta http-equiv="Content-Type" content=  
"text/html; charset=utf-8" />  
5 <title>無標題文件</title>  
6 </head>  
7  
8 <body>
```

HTML 4 網頁會加上一連串的宣告



程式碼 | 分割 | 設計 | 即時 | 標題:

```
1 <!doctype html>  
2 <html>  
3 <head>  
4 <meta charset="utf-8">  
5 <title>無標題文件</title>  
6 </head>  
7  
8 <body>  
9 </body>  
10  
11
```

HTML 5 網頁將宣告簡化了

3-2 利用 Div 標籤劃分網頁區塊



利用 Div 標籤劃分網頁區塊

- 初步規劃網頁架構
- 插入 Div 標籤

初步規劃網頁架構

用來放置網頁內容的區域, 包含左右兩欄

表頭

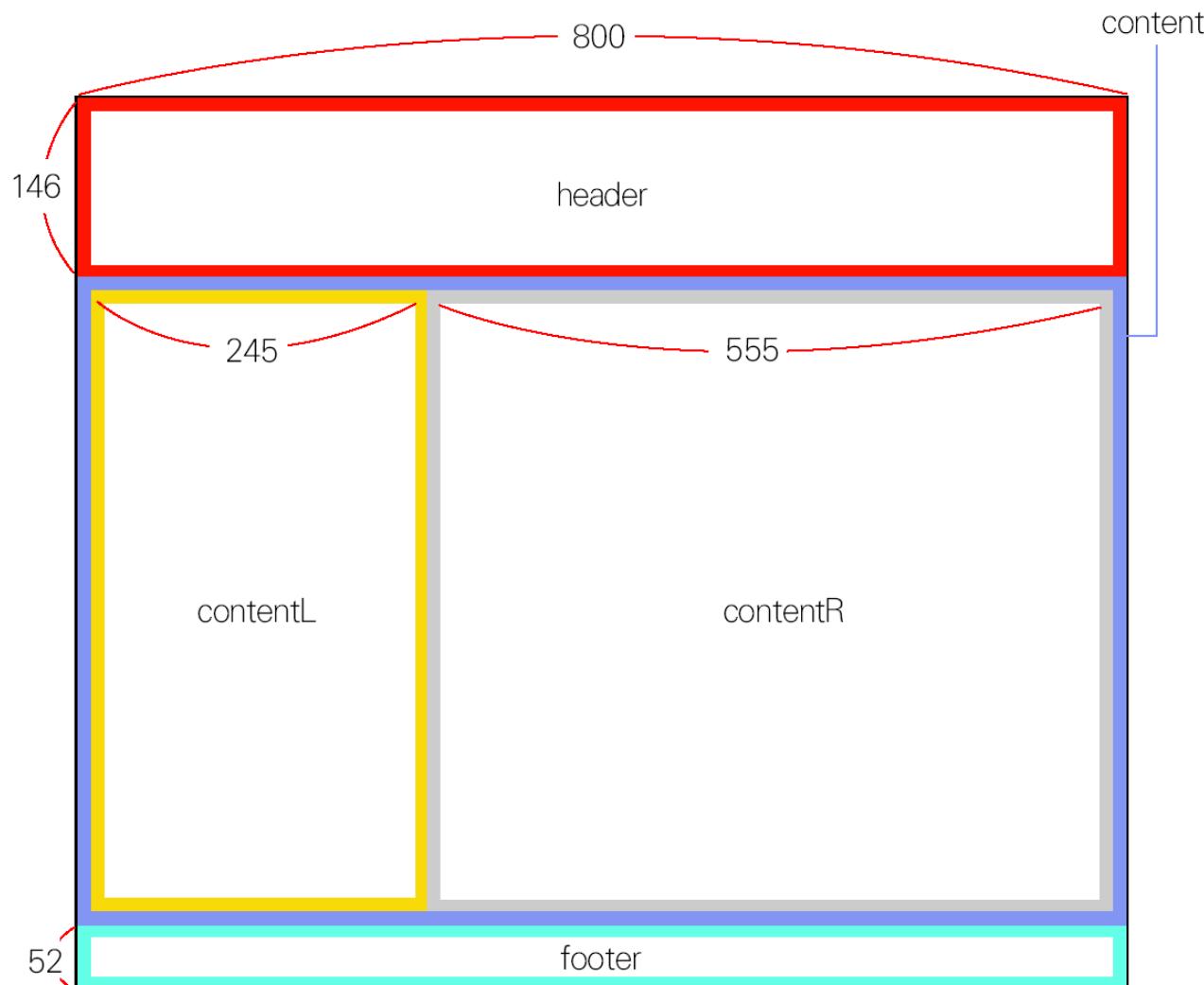


根據內容將網頁的平面設計稿先劃分大區塊

初步規劃網頁架構

- 命名的第 1 個字元務必使用英文字，第 2 個字元之後用英文或數字皆可。
- 區塊名稱不可包含空格與特殊符號，當然也不可使用中文。
- 命名時盡量取個容易辨識的名稱，例如表頭區(頁首區)域通常會命名為“header”、放置內容的區域命名為“content”、頁尾區域則命名為“footer”、...等，這樣在設定 CSS 樣式時較容易辨識。

初步規劃網頁架構



其中的 contentL、contentR 區塊是包含在 content 區塊中

插入 Div 標籤

- 由上而下依序定義區塊
- 建立層層包覆的區塊

建立由上而下依序定義區塊

建立main0201.html，並載入html5格式，建立區塊

```
mainOLD.html main.html main0201.html  
main0201.html > html  
1 <!DOCTYPE html>  
2 <html lang="en">  
3 <head>  
4     <meta charset="UTF-8">  
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">  
6     <title>Light Design Studio輕設計網站工作室</title>  
7 </head>  
8 <body>  
9     <div id="header">header</div>  
10    <div id="content">content</div>  
11    <div id="footer">footer</div>  
12 </body>  
13 </html>
```

建立content的子區塊

```
8 <body>
9   <div id="header">header</div>
10  <div id="content">
11    <div id="contentL">contentL</div>
12    <div id="contentR">contentR</div>
13    content</div>
14  <div id="footer">footer</div>
15 </body>
16 </html>
```

建立link css 到website0201.css

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Light Design Studio輕設計網站工作室</title>
7     <link rel="stylesheet" href="website0201.css">
8 </head>
9 <body>
10    <div id="#">按一下以顯示 4 項定義。
11    <div id="content">
12        <div id="contentL">contentL</div>
13        <div id="contentR">contentR</div>
14        content</div>
15        <div id="footer">footer</div>
16    </body>
17 </html>
```

練習二

- 練習建一個三欄架構的DIV網頁版面
- 請開新HTML檔，檔名：main0202.html，並符合下列要求
- 設定title標籤：Cute Design Studio可愛網頁工作室
- 需包括頁首header、頁尾footer
- 所有css規則請記錄在website0202.css中
- 內容頁content需包括contentL、contentM、contentR

利用 CSS 美化 Div 標籤

- 設定 Div 標籤的背景及寬高
- 利用「**float**」屬性讓 Div 標籤左右並排
- 利用「**clear**」屬性清除「**float**」屬性的影響

設定 header Div 標籤的背景及寬高

step01



The screenshot shows a Visual Studio Code interface with multiple tabs. The active tab is 'website0201.css'. A red box highlights this tab and the entire code block below it. The code defines a style for the '#header' selector, setting its background image to 'images/header.gif', repeating it once, and specifying a width of 800px and a height of 146px.

```
1 #header {  
2     background-image: url("images/header.gif");  
3     background-repeat: no-repeat;  
4     width: 800px;  
5     height: 146px;  
6 }
```

執行結果

header 內容刪除

contentL
contentR
content
footer



設定 content Div 標籤的背景及寬高

```
7 #content{  
8     background-image:url("images/content_bg.gif");  
9     background-repeat:repeat-y;  
10    width:800px;  
11}
```

content 執行結果



設定 footer Div 標籤的背景及寬高

step06

```
12 #footer{  
13     background-image: url("images/footer.gif");  
14     background-repeat: no-repeat;  
15     width: 800px;  
16     height: 52px;  
17 }
```

footer 執行結果



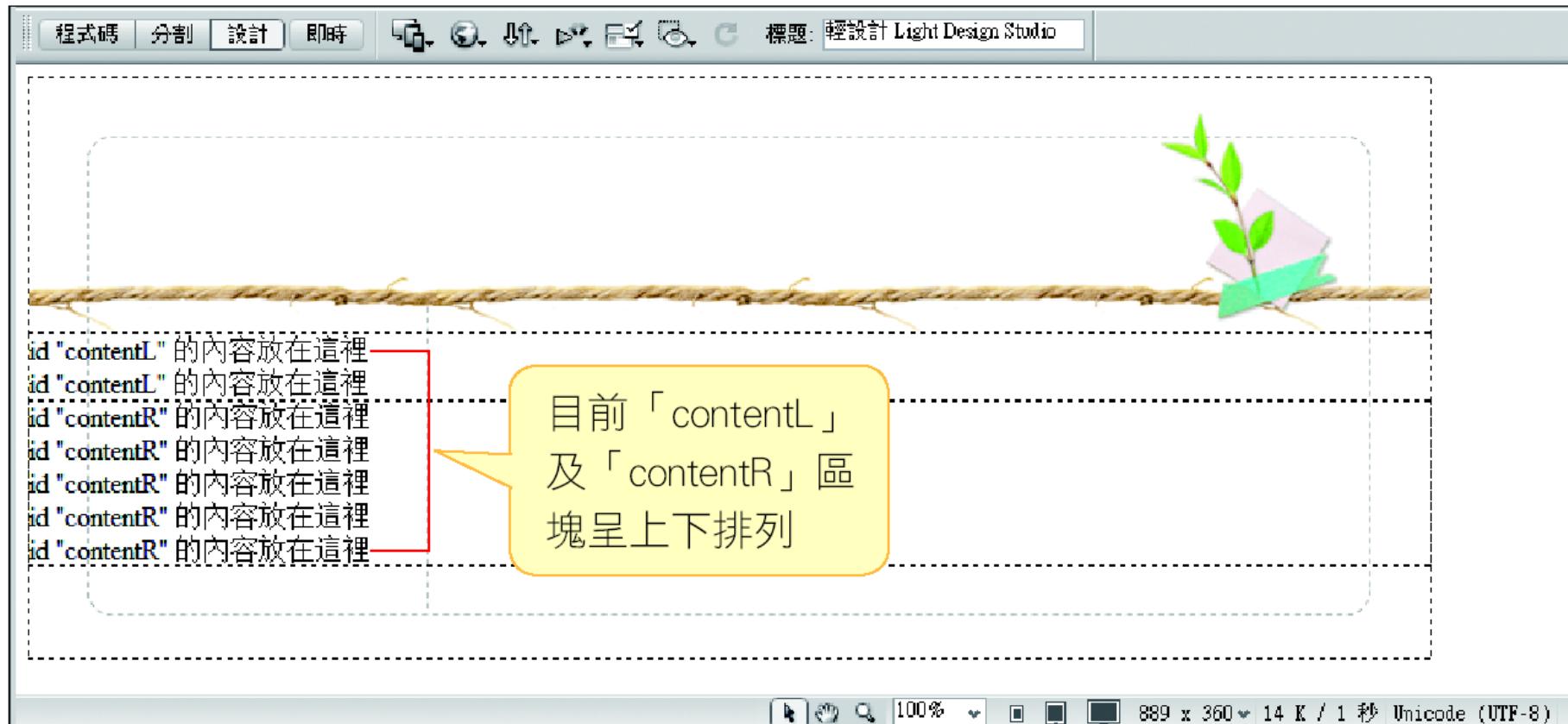
練習三

- ▣ 利用images1圖片，使用相同的方式，將三欄式的架構，加上相關背景圖。
- ▣ header:header_bg.jpg ,no-repeat ,Width:800 Height:129
- ▣ 消除header內容文字
- ▣ content:content_bg.jpg ,repeat-y ,Width:800
- ▣ 消除content內容文字
- ▣ footer:footer_bg.jpg ,no-repeat ,Width:800 Height:52
- ▣ 消除footer內容文字

執行結果



利用「flex」屬性讓 Div 標籤左右並排



為了方便看出 **float** 屬性的作用，我們預先在「contentL」及「contentR」區塊中增加多行文字內容，並讓「contentR」區塊包含較多行的文字，以便觀察出箇中端倪

建立練習資料

```
13    <div id="content">
14        <div id="contentL">
15            id "contentL" 的內容放這裡<br>
16            id "contentL" 的內容放這裡<br>
17        </div>
18        <div id="contentR">
19            id "contentR" 的內容放這裡<br>
20            id "contentR" 的內容放這裡<br>
21            id "contentR" 的內容放這裡<br>
22            id "contentR" 的內容放這裡<br>
23            id "contentR" 的內容放這裡<br>
24        </div>
25    </div>
```

利用「flex」屬性讓 Div 標籤左右並排

step 01

將#content的方框設定flex功能

```
19 #content {  
20   background-image : url(images/content_bg.gif);  
21   background-repeat: repeat-y;  
22   width ..... : 800px;  
23   display ..... : flex;  
24   flex-direction:row;  
25   flex-wrap:nowrap;  
26   justify-content: flex-start;  
27   align-items:stretch;  
28   align-content:stretch;  
29 }
```

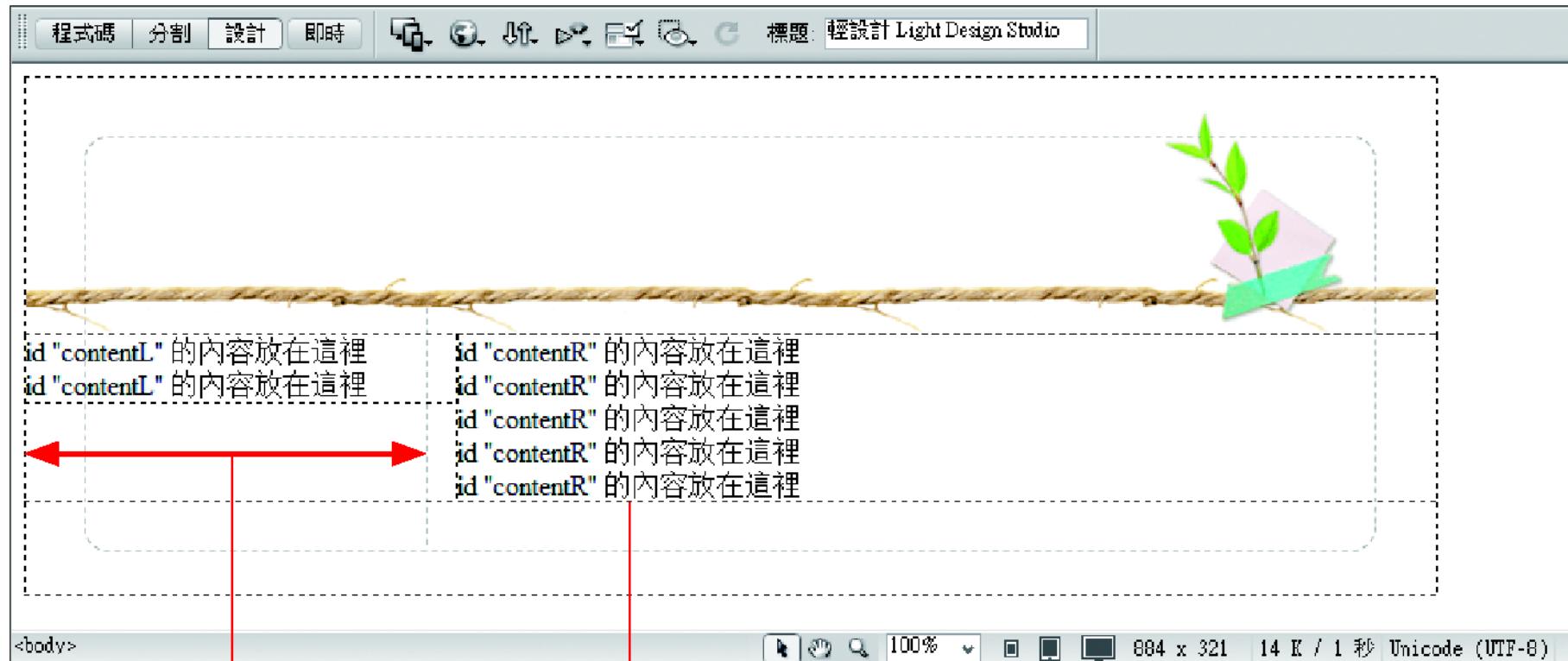
利用「flex」屬性讓 Div 標籤左右並排

step 02

設定**contentL** 與 **contentR** 中的**Width**寬

```
38  #contentL {  
39      width: 245px;  
40  }  
41  #contentR {  
42      width: 555px;  
43  }  
44
```

利用「flex」屬性讓 Div 標籤左右並排



245 像素寬

超過「contentL」區塊的內容也會乖乖地跟左邊保持一定的距離了 (到此的完成結果可參考 ch03-04.html)

練習四

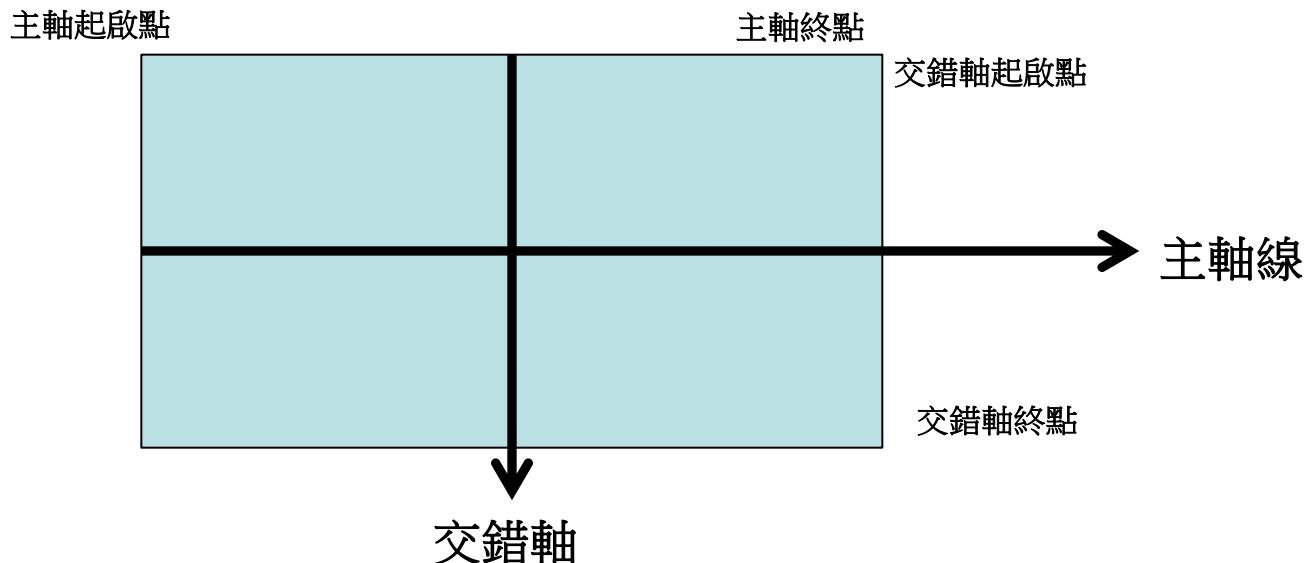
- 設定三欄利用flex浮動的功能，讓content內的DIV三欄左右並排。
- contentL:245px
- contentM:450px
- contentR:105px

margin和padding功能說明

- 設定兩個DIV分別為a，300*300，設背景顏色，display:flex，包含b為150*150，設背景顏色，文字設定白色，b內含文字為：內容文字呈現，用以練習測試，margin和padding功能。
 - 設定b的margin為top:40px，left:40px。
 - 設定b的padding為top:20px，left:20px。

display:flex功能說明

- 4. Div 標籤設定flex具有自成一排的特性，若想讓區塊並排，必須替上層區塊設定 flex功能屬性；不過當區塊設有flex屬性後，其他的相依屬性也可能產生影響作用，以下建立用示意圖幫你釐清觀念。假設利用 Div 標籤建立了 A1、A2、A3、A4、A5 五個區塊，因此會以上下並列的方式存在：



建立test001.html

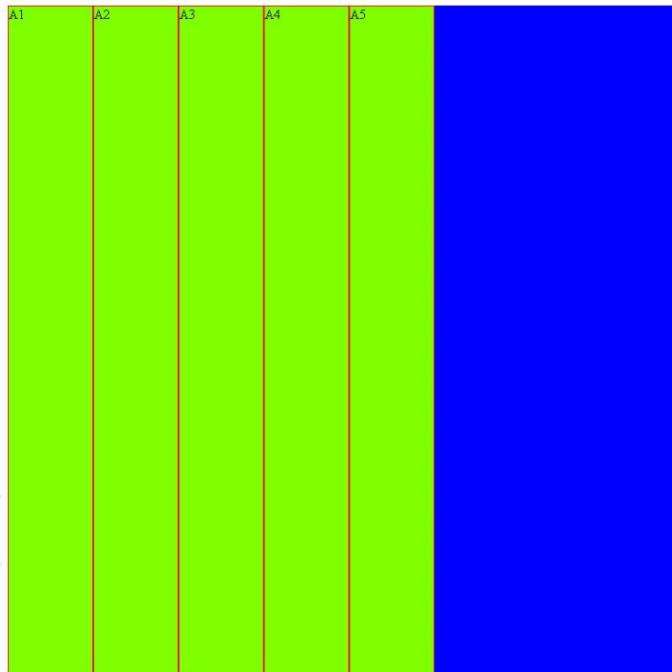
```
main.html X main0301.html # website0301.css test001.html X
test001.html > html > head
content="IE=edge">
6   <meta name="viewport" content="width=device-width,
initial-scale=1.0">
7   <title>css flex texting</title>
8   <style type="text/css">
9     .content{
10       background-color: blue;
11       width: 800px;
12       height: 800px;
13     }
14     .item{
15       background-color: chartreuse;
16       width: 100px;
17       border: solid 1px red;
18     }
19   </style>
20 </head>
21 <body>
22   <div class="content">
23     <div class="item">A1</div>
24     <div class="item">A2</div>
25     <div class="item">A3</div>
26     <div class="item">A4</div>
27     <div class="item">A5</div>
28   </div>
```

執行結果

A1
A2
A3
A4
A5

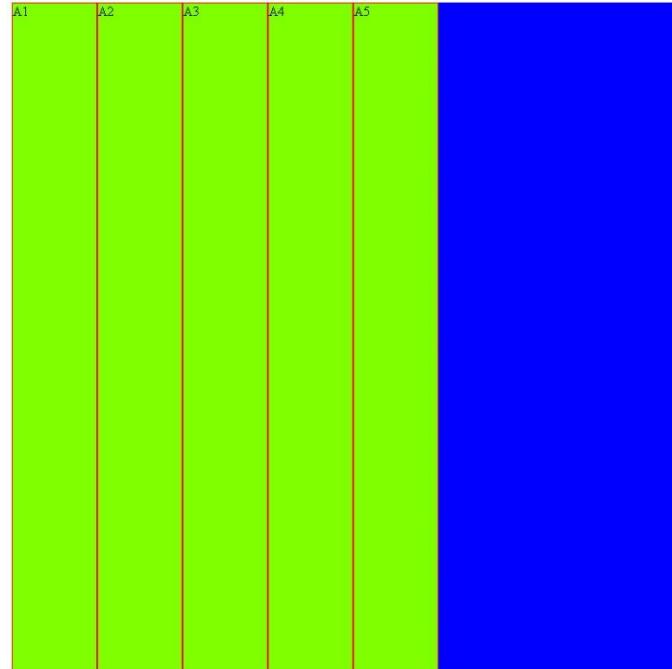
建立flex功能與相關屬性

```
<title>css flex texting</title>
<style type="text/css">
    .content{
        background-color: blue;
        width: 800px;
        height: 800px;
        display:flex
    }
    .item{
        background-color: chartreuse;
        width: 100px;
        border: solid 1px red;
    }
</style>
```



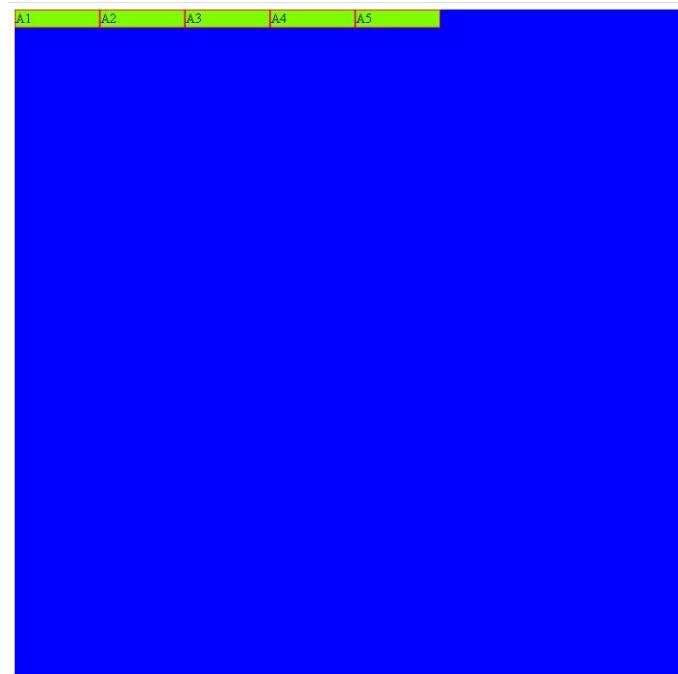
flex預設相關屬性內容

```
<title>css flex text</title>
<style type="text/css">
    .content{
        background-color: blue;
        width: 800px;
        height: 800px;
        display: flex;
        flex-direction: row;
        flex-wrap: nowrap;
        justify-content: flex-start;
        align-items:stretch;
        align-content:stretch;
    }
    .item{
        background-color: chartreuse;
        width: 100px;
        border: solid 1px red;
    }
</style>
```



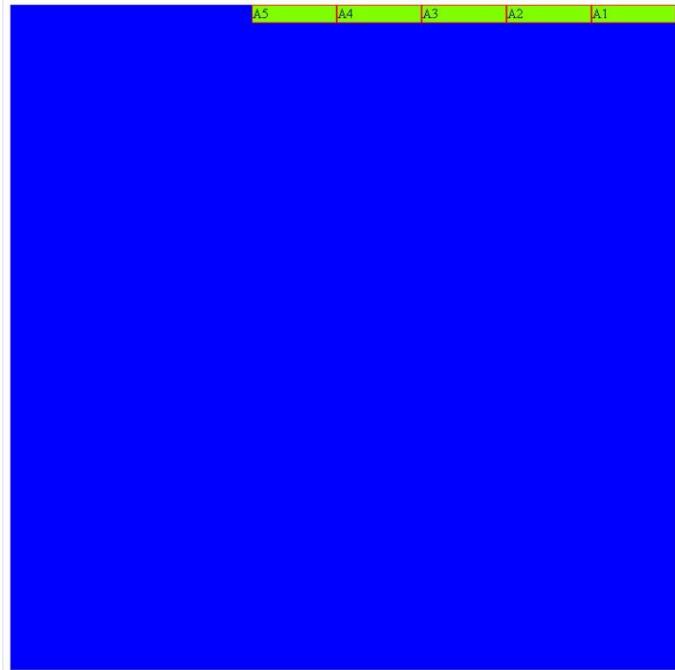
先調改align-item屬性內容，準備測試

```
<title>css flex texting</title>
<style type="text/css">
    .content{
        background-color: blue;
        width: 800px;
        height: 800px;
        display: flex;
        flex-direction: row;
        flex-wrap: nowrap;
        justify-content: flex-start;
        align-items: flex-start;
        align-content: stretch;
    }
    .item{
        background-color: chartreuse;
        width: 100px;
        border: solid 1px red;
    }
</style>
```



測試更改flex-direction屬性內容

```
<style type="text/css">
  .content{
    background-color: blue;
    width: 800px;
    height: 800px;
    display: flex;
    /* flex-direction: row | row-reverse |
    column | column-reverse; content中內容項目的
    排列方式與方向 */
    flex-direction: row;
    flex-wrap: nowrap;
    justify-content: flex-start;
    align-items: flex-start;
    align-content: stretch;
  }
  .item{
    background-color: chartreuse;
    width: 100px;
    border: solid 1px red;
  }
</style>
```



增加div A6-A9，更改flex-wrap屬性

```
29 <body>
30     <div class="content">
31         <div class="item">A1</div>
32         <div class="item">A2</div>
33         <div class="item">A3</div>
34         <div class="item">A4</div>
35         <div class="item">A5</div>
36         <div class="item">A6</div> //增加的div
37         <div class="item">A7</div>
38         <div class="item">A8</div>
39         <div class="item">A9</div> //增加的div
40         <div class="item">A10</div>
41     </div>
42 </body>
```

增加div A6-A9，更改flex-wrap屬性

```
<style type="text/css">
  .content{
    background-color: blue;
    width: 800px;
    height: 800px;
    display: flex;
    /* flex-direction: row | row-reverse |
       column | column-reverse; content中內容項目的
       排列方式與方向 */
    flex-direction: row;
    /* flex-wrap: nowrap | wrap |
       wrap-reverse; items 在有需要的時候自動換行
       */
    flex-wrap: wrap;
    justify-content: flex-start;
    align-items: flex-start;
    align-content: stretch;
  }
  .item{
    background-color: chartreuse;
    width: 100px;
    border: solid 1px red;
  }
</style>
```

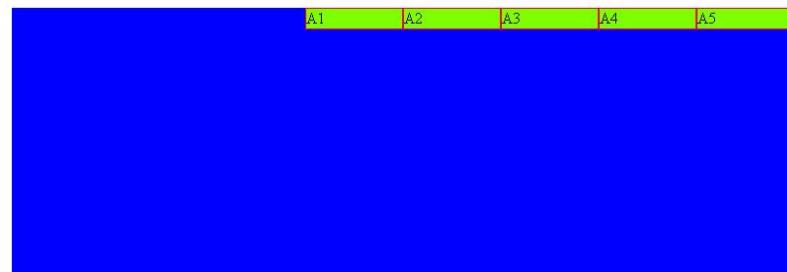


註解div A6-A9

```
30 <body>
31     <div class="content">
32         <div class="item">A1</div>
33         <div class="item">A2</div>
34         <div class="item">A3</div>
35         <div class="item">A4</div>
36         <div class="item">A5</div>
37         <!-- <div class="item">A6</div>
38             <div class="item">A7</div>
39             <div class="item">A8</div>
40             <div class="item">A9</div>
41             <div class="item">A10</div> -->
42     </div>
43 </body>
```

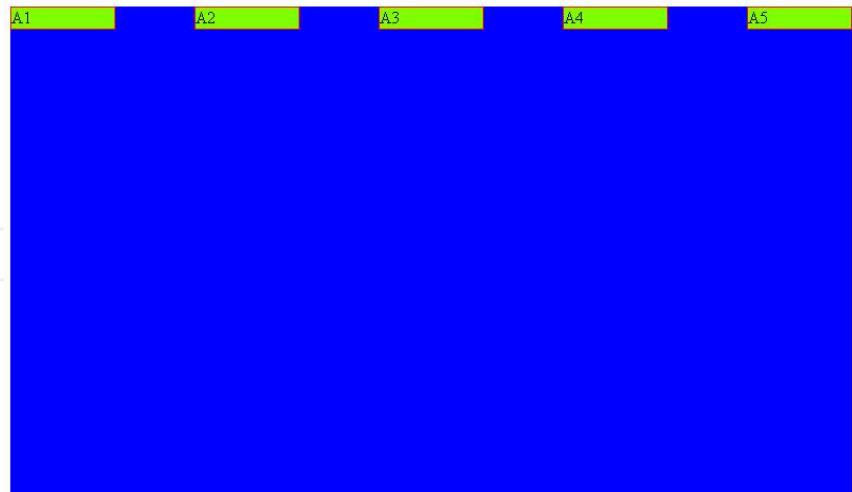
註解div A6-A9，更改justify-content

```
...     flex-direction: row;
...     /* flex-wrap: nowrap | wrap |
...        wrap-reverse; items 在有需要的時候自動換行
...     */
...     flex-wrap: wrap;
...     /* justify-content:flex-start|flex-end|
...        center|space-between|space-around|
...        space-evenly;content主軸作對齊的方式 */
...     justify-content: flex-end;
...     align-items: flex-start;
...     align-content: stretch;
...
... }
...
.item{
    background-color: chartreuse;
    width: 98px;
    border: solid 1px red;
}
</style>
</head>
```



註解div A6-A9，更改justify-content2

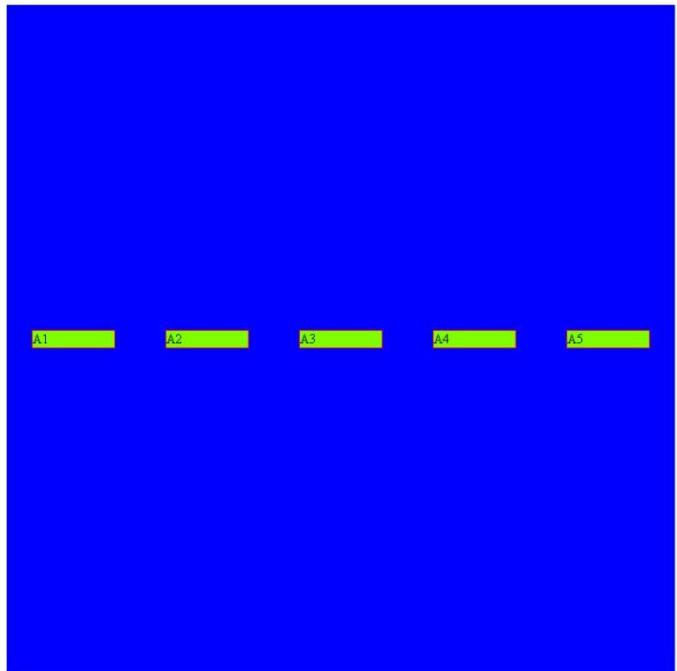
```
        flex-direction: row;
        /* flex-wrap: nowrap | wrap |
        wrap-reverse; items 在有需要的時候自動換行
        */
        flex-wrap: wrap;
        /* justify-content:flex-start|flex-end|
        center|space-between|space-around|
        space-evenly;content主軸作對齊的方式 */
        justify-content: space-between;
        align-items: flex-start;
        align-content: stretch;
    }
.item{
    background-color: chartreuse;
    width: 98px;
    border: solid 1px red;
}
</style>
```



更改align-item屬性內容

```
.....flex-wrap: wrap;
/* justify-content:flex-start|flex-end|
center|space-between|space-around|
space-evenly;content主軸作對齊的方式 */
justify-content: space-around;
/* align-items: flex-start | flex-end | .
center | baseline | stretch;content交錯軸作
對齊的方式 */
align-items:center;
align-content:stretch;
}

.item{
background-color: #chartreuse;
width: 98px;
border: solid 1px red;
}
</style>
```



align-item:baseline屬性測試1

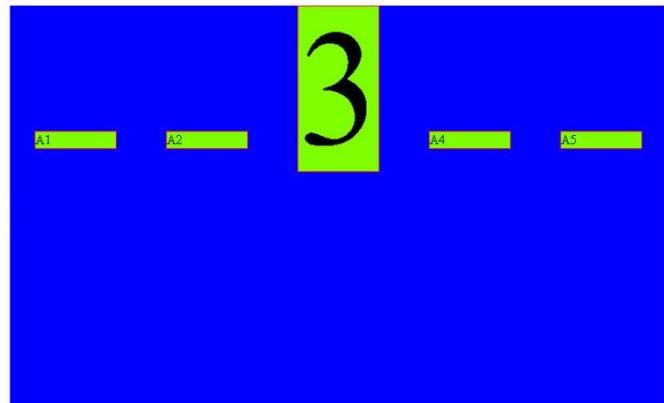
```
....<title>css flex texting</title>
....<style type="text/css">
....  .fontset{
....    font-size: 200px;
....    line-height: 200px;
....  }
....  .content{
....    background-color: ■blue;
....    width: 800px;
....    height: 800px;
....    display: flex;
....    /* flex-direction: row | row-reverse |
....    column | column-reverse; content中內容項目的
....    排列方式與方向*/
....
```



```
35 <body>
36     <div class="content">
37         <div class="item">A1</div>
38         <div class="item">A2</div>
39         <div class="item fontset">3</div> (highlighted)
40         <div class="item">A4</div>
41         <div class="item">A5</div>
42         <!-- <div class="item">A6</div>
43         <div class="item">A7</div>
44         <div class="item">A8</div>
45         <div class="item">A9</div>
46         <div class="item">A10</div> -->
47     </div>
48 </body>
```

align-item:baseline屬性測試3

```
.....flex-wrap: wrap;
...../* justify-content:flex-start|flex-end|
center|space-between|space-around|
space-evenly;content主軸作對齊的方式 */
justify-content: space-around;
/* align-items: flex-start | flex-end | 
center | baseline | stretch;content交錯軸作
對齊的方式 */
align-items:baseline;
align-content:stretch;
}
.item{
background-color: chartreuse;
width: 98px;
border: solid 1px red;
}
/style>
```



align-content測試，設定content

```
test002.html > html > head > style > .content
13      }
14      .content {
15          background-color: blue;
16          width: 800px;
17          height: 1200px; //高度
18          display: flex;
19          flex-direction: row;
20          /* flex-direction:row|row-reverse|column|
21             column-reverse; content中內容的排列方式與方向 */
22          flex-wrap: wrap;
23          /* flex-wrap:nowrap|wrap|wrap-reverse; items在有
24             需要的時候自動換行 */
25          justify-content: space-around;
26          /* justify-content:flex-start|flex-end|center|
27             space-between|space-around|space-evenly; content
28             主軸作用對齊的方式 */
29          align-items: stretch; //子元素垂直對齊
30          /* align-items:flex-start|flex-end|center|
31             baseline|stretch; content交錯軸作用對齊的方式 */
32          align-content: stretch;
33      }
34      .item {
35          background-color: chartreuse;
36          width: 200px;
37          border: solid 1px red;
38      }
39  
```



align-content測試1，增加item內容

```
35 <body>
36   <div class="content">
37     <div class="item"><h2>A1</h2>
38       <p>Lorem ipsum dolor sit amet consectetur·
39         adipisicing elit. Dolor ducimus illum·
40         accusamus minima. Nam repellat maiores·
41         molestias ipsum assumenda preferendis?·
42         Voluptates, veritatis minima velit aperiam·
43         soluta commodi tempore dignissimos dolore.</
44           p>
45     </div>
46     <div class="item"><h2>A2</h2>
47       <p>Lorem ipsum dolor sit amet consectetur·
48         adipisicing elit. Dolor ducimus illum·
49         accusamus minima. Nam repellat maiores·
50         molestias ipsum assumenda preferendis?·
51         Voluptates, veritatis minima velit aperiam·
52         soluta commodi tempore dignissimos dolore.</
53           p></div>
54     <div class="item"><h2>A3</h2>
55       <p>Lorem ipsum dolor sit amet consectetur·
56         adipisicing elit. Dolor ducimus illum·
57         accusamus minima. Nam repellat maiores·
58         molestias ipsum assumenda preferendis?
```

align-content測試2，更改css內容

```
..... justify-content: space-around;
..... /* align-items: flex-start | flex-end | 
..... center | baseline | stretch;content交錯軸作
..... 對齊的方式 */
..... align-items:stretch;
..... /* align-content: flex-start | flex-end | 
..... center | space-between | space-around | 
..... stretch;content內交錯軸整體元件的對齊 */
..... align-content:center;
}
.item{
    background-color: chartreuse;
    width: 98px;
    border: solid 1px red;
}
</style>
```

練習：使用flex功能調出下列版面



復原預設參數

```
13
14     .content {
15         background-color: blue;
16         width: 800px;
17         height: 800px; 高度
18         display: flex;
19         flex-direction: row;
20         /* flex-direction:row|row-reverse|column|
21            column-reverse; content中內容的排列方式與方向 */
22         flex-wrap: wrap;
23         /* flex-wrap:nowrap|wrap|wrap-reverse; items在有
24            需要的時候自動換行 */
25         justify-content: space-around;
26         /* justify-content:flex-start|flex-end|center|
27            space-between|space-around|space-evenly; content
28            主軸作用對齊的方式 */
29         align-items: stretch; 垂直
30         /* align-items:flex-start|flex-end|center|
31            baseline|stretch; content交錯軸作用對齊的方式 */
32         align-content: stretch; 水平
33         /* align-content:flex-start|flex-end|center|
34            space-between|space-around; stretch; content內容
35            交錯軸整體元件的對齊 */
36     }
37
38     .item {
39         background-color: chartreuse;
40         width: 98px; 寬度
41         border: solid 1px red;
42     }
```



設定為下列flex結果

A1	A2	A3	A4	A5	A6	A7	A8
<p> Lorem ipsum dolor sit amet consectetur, adipisicing elit. Asperiores laboriosam dignissimos illum voluptate odio, deleniti natus numquam vitae ab earum necessitatibus dolorum hic quasi corrupti mollitia perferendis. Beatae, sunt sint?</p>	<p> Lorem ipsum dolor sit amet consectetur, adipisicing elit. Asperiores laboriosam dignissimos illum voluptate odio, deleniti natus numquam vitae ab earum necessitatibus dolorum hic quasi corrupti mollitia perferendis. Beatae, sunt sint?</p>	<p> Lorem ipsum dolor sit amet consectetur, adipisicing elit. Asperiores laboriosam dignissimos illum voluptate odio, deleniti natus numquam vitae ab earum necessitatibus dolorum hic quasi corrupti mollitia perferendis. Beatae, sunt sint?</p>	<p> Lorem ipsum dolor sit amet consectetur, adipisicing elit. Asperiores laboriosam dignissimos illum voluptate odio, deleniti natus numquam vitae ab earum necessitatibus dolorum hic quasi corrupti mollitia perferendis. Beatae, sunt sint?</p>	<p> Lorem ipsum dolor sit amet consectetur, adipisicing elit. Asperiores laboriosam dignissimos illum voluptate odio, deleniti natus numquam vitae ab earum necessitatibus dolorum hic quasi corrupti mollitia perferendis. Beatae, sunt sint?</p>	<p> Lorem ipsum dolor sit amet consectetur, adipisicing elit. Asperiores laboriosam dignissimos illum voluptate odio, deleniti natus numquam vitae ab earum necessitatibus dolorum hic quasi corrupti mollitia perferendis. Beatae, sunt sint?</p>	<p> Lorem ipsum dolor sit amet consectetur, adipisicing elit. Asperiores laboriosam dignissimos illum voluptate odio, deleniti natus numquam vitae ab earum necessitatibus dolorum hic quasi corrupti mollitia perferendis. Beatae, sunt sint?</p>	<p> Lorem ipsum dolor sit amet consectetur, adipisicing elit. Asperiores laboriosam dignissimos illum voluptate odio, deleniti natus numquam vitae ab earum necessitatibus dolorum hic quasi corrupti mollitia perferendis. Beatae, sunt sint?</p>

每個元件自我對齊align-self屬性

- align-self: auto|stretch|center|flex-start|flex-end|baseline
- default:auto
- 各別調整子元素在交錯軸線的位置，屬性與 align-item 相同。假如我們已經在父元素上設定 align-item，但要其中一個內容物的位置需要調整成其他對齊方式時，可採用 align-self 。

每個元件自我對齊align-self屬性1

```
.....flex-wrap: wrap;  
...../* justify-content:flex-start|flex-end|center|space-between|  
.....space-around|space-evenly;content主軸作對齊的方式 */  
.....justify-content: space-around;  
...../* align-items: flex-start | flex-end | center | baseline | stretch;  
.....content交錯軸作對齊的方式 */  
.....align-items:stretch;  
...../* align-content: flex-start | flex-end | center | space-between |  
.....space-around | stretch;content內交錯軸整體元件的對齊 */  
.....align-content:stretch;  
}  
.item{ █ ...  
}  
.alignSelfStart{  
..../* align-self: auto|stretch|center|flex-start|flex-end|baseline 調整子  
.....元素在交錯軸線的位置 */  
....align-self:center;  
}  
</style>
```

每個元件自我對齊align-self屬性2

```
40 <body>
41   <div class="content">
42     <div class="item alignSelfStart"><h2>A1</h2>
43       <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Dolor
44         ducimus illum accusamus minima. Nam repellat maiores molestias ipsum
45         assumenda preferendis? Voluptates, veritatis minima velit aperiam
46         soluta commodi tempore dignissimos dolore.</p>
47     </div>
48     <div class="item"><h2>A2</h2> ...
49     </div>
50     <div class="item alignSelfStart"><h2>A3</h2>
51       <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Dolor
52         ducimus illum accusamus minima. Nam repellat maiores molestias ipsum
53         assumenda preferendis? Voluptates, veritatis minima velit aperiam
54         soluta commodi tempore dignissimos dolore.</p></div>
55     <div class="item"><h2>A4</h2>
56       <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Dolor
57         ducimus illum accusamus minima. Nam repellat maiores molestias ipsum
58         assumenda preferendis? Voluptates, veritatis minima velit aperiam
59         soluta commodi tempore dignissimos dolore.</p></div>
```

測試執行

A2	A4	A5	A6	A7	A8
Lore ipsum dolor sit amet consectetur adipisicing elit. Dolor ducimus illum accusamus minima. Nam repellat maiores molestias ipsum assumenda perferendis? Voluptates, veritatis minima velit aperiam soluta commodi tempore dignissimos dolore.	Lore ipsum dolor sit amet consectetur adipisicing elit. Dolor ducimus illum accusamus minima. Nam repellat maiores molestias ipsum assumenda perferendis? Voluptates, veritatis minima velit aperiam soluta commodi tempore dignissimos dolore.	Lore ipsum dolor sit amet consectetur adipisicing elit. Dolor ducimus illum accusamus minima. Nam repellat maiores molestias ipsum assumenda perferendis? Voluptates, veritatis minima velit aperiam soluta commodi tempore dignissimos dolore.	Lore ipsum dolor sit amet consectetur adipisicing elit. Dolor ducimus illum accusamus minima. Nam repellat maiores molestias ipsum assumenda perferendis? Voluptates, veritatis minima velit aperiam soluta commodi tempore dignissimos dolore.	Lore ipsum dolor sit amet consectetur adipisicing elit. Dolor ducimus illum accusamus minima. Nam repellat maiores molestias ipsum assumenda perferendis? Voluptates, veritatis minima velit aperiam soluta commodi tempore dignissimos dolore.	Lore ipsum dolor sit amet consectetur adipisicing elit. Dolor ducimus illum accusamus minima. Nam repellat maiores molestias ipsum assumenda perferendis? Voluptates, veritatis minima velit aperiam soluta commodi tempore dignissimos dolore.
A1 Lore ipsum dolor sit amet consectetur adipisicing elit. Dolor ducimus illum accusamus minima. Nam repellat maiores molestias ipsum assumenda perferendis? Voluptates, veritatis minima velit aperiam soluta commodi tempore dignissimos dolore.	A3 Lore ipsum dolor sit amet consectetur adipisicing elit. Dolor ducimus illum accusamus minima. Nam repellat maiores molestias ipsum assumenda perferendis? Voluptates, veritatis minima velit aperiam soluta commodi tempore dignissimos dolore.				

order測試

```
..... .alignSelfStart{  
.....     /* align-self: auto|stretch|center|flex-start|flex-end|baseline 調整子  
.....     元素在交錯軸線的位置 */  
.....     align-self:center;  
..... }  
..... .orderCtl{  
.....     /* order:0|0-N|default:0|控制區塊順序位子 */  
.....     order:-1;  
..... }  
..... </style>  
</head>  
<body>  
..... <div class="content">  
>..... <div class="item alignSelfStart"><h2>A1</h2> ...  
..... </div>  
>..... <div class="item"><h2>A2</h2> ...  
..... </div>  
..... <div class="item alignSelfStart orderCtl"><h2>A3</h2>  
.....     <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Dolor  
.....     ducimus illum accusamus minima. Nam repellat maiores molestias ipsum  
.....     assumenda preferendis? Voluptates, veritatis minima velit aperiam  
.....     soluta commodi tempore dignissimos dolore.</p></div>  
..... <div class="item"><h2>A4</h2>
```

測試執行A3區塊放在前

		A2	A4	A5	A6	A7	A8
A3	A1	Lorem ipsum dolor sit amet consectetur adipisicing elit. Dolor ducimus illum accusamus minima. Nam repellat maiores molestias ipsum assumenda perferendis? Voluptates, veritatis minima velit aperiam soluta commodi tempore dignissimos dolore.					
		Lore ipsum dolor sit amet consectetur adipisicing elit. Dolor ducimus illum accusamus minima. Nam repellat maiores molestias ipsum assumenda perferendis? Voluptates, veritatis minima velit aperiam soluta commodi tempore dignissimos dolore.	Lore ipsum dolor sit amet consectetur adipisicing elit. Dolor ducimus illum accusamus minima. Nam repellat maiores molestias ipsum assumenda perferendis? Voluptates, veritatis minima velit aperiam soluta commodi tempore dignissimos dolore.	Lore ipsum dolor sit amet consectetur adipisicing elit. Dolor ducimus illum accusamus minima. Nam repellat maiores molestias ipsum assumenda perferendis? Voluptates, veritatis minima velit aperiam soluta commodi tempore dignissimos dolore.	Lore ipsum dolor sit amet consectetur adipisicing elit. Dolor ducimus illum accusamus minima. Nam repellat maiores molestias ipsum assumenda perferendis? Voluptates, veritatis minima velit aperiam soluta commodi tempore dignissimos dolore.	Lore ipsum dolor sit amet consectetur adipisicing elit. Dolor ducimus illum accusamus minima. Nam repellat maiores molestias ipsum assumenda perferendis? Voluptates, veritatis minima velit aperiam soluta commodi tempore dignissimos dolore.	Lore ipsum dolor sit amet consectetur adipisicing elit. Dolor ducimus illum accusamus minima. Nam repellat maiores molestias ipsum assumenda perferendis? Voluptates, veritatis minima velit aperiam soluta commodi tempore dignissimos dolore.

練習

- 將所有的div調整為置中放置
- A5設定為flex-end,A2設定為flex-start
- A1放置在A8的後面

重點整理

1. HTML (Hyper Text Markup Language) 是一種用來描述網頁結構的語法，其結構是由一組組由“<”與 ">" 組成的「標籤」；CSS (Cascading Style Sheets) 中譯為**串接樣式表**，是一種用來改變網頁外觀樣式的語法。簡單來說，「HTML」的作用是構成網頁架構，「CSS」的作用是美化網頁外觀。

重點整理

2. 要建立網頁版面時，通常會先依設計好的網頁版型劃分出多個區塊，並替每個區塊命名，接著用 Div 標籤在網頁中加入區塊，然後利用 CSS 控制個別的區塊屬性，例如設定寬、高、背景、位置、...等。大致的製作流程可參考下圖：



重點整理

3. CSS 樣式表可區分為**內部樣式表**與**外部樣式表**，下表供你對照它們的特性：

CSS	存在位置	使用時機
內部樣式表	網頁本身的原始碼中	由於影響範圍只有網頁本身，因此適用於網頁數量少的小型網站、或是具獨特性的網頁
外部樣式表	另外儲存在一個附檔名為 *.css 的檔案中	在實務上，我們通常將網站共同的 CSS 彙整在外部樣式表檔中，再將網頁與此樣式表檔連結，尤其當網頁數量繁多時，使用外部樣式表可省去逐頁開啟網頁、修改 CSS 的麻煩