

Chapter7

圖文編排 1

目錄

- 文繞圖 3
- 相對位置、絕對位置 9
- 疊層 19

文繞圖

- 網頁設計中編排包含圖片、文字的版面時，由於圖片通常比文字高出許多，若將它們排在同一列，圖片旁邊容易出現一大片留白。



排球運動起源於美國。1895年2月9日，由美國麻省霍利奧克基督教青年會幹事威廉·摩根發明。起初，人們分站在網球場球網的兩側，將籃球托來托去，參加人數、擊球次數不限。比賽中網高1.98米。這就是排球的雛形。

文繞圖

- 為了解決這種狀況，我們通常會以文繞圖的方式來進行排版，就是將文字排到圖片旁邊的空白處。



排球運動起源於美國。1895年2月9日，由美國麻省霍利奧克基督教青年會幹事威廉·摩根發明。起初，人們分站在網球場球網的兩側，將籃球托來托去，參加人數、擊球次數不限。比賽中網高1.98米。這就是排球的雛形。

文繞圖

○ 語法：

1. `div{ float: left; }` 圖片置左
2. `div{ float: right; }` 圖片置右

文繞圖程式碼

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Chapter 7</title>
6     <style>
7       div.img_left{
8         float:left;
9         margin: 0 10px 10px 0;
10      }
11      div.img_left img{
12        border: 1px solid #000000;
13        padding: 5px;
14      }
15      div.img_left p{
16        color: #000000;
17        font-size: 25px;
18        margin: 5px 0 0;
19        text-align: center;
20      }
21
22      div.img_right{
23        float:right;
24        margin: 0 10px 10px 0;
25      }
26      div.img_right img{
27        border: 1px solid #000000;
28        padding: 5px;
29      }
30      div.img_right p{
31        color: #000000;
32        font-size: 25px;
33        margin: 5px 0 0;
34        text-align: center;
35      }
36      p.content{
37        color: #000000;
38        font-size: 20px;
39        text-align: left;
40        height:300px;
41      }
42    </style>
  </head>
```


文繞圖程式碼

```
42 </head>
43 <body>
44   <div class="img_left">
45     
46     <p>足球</p>
47   </div>
48   <p class="content">
49     足球是足球運動或足球比賽的簡稱；當然它也指足球比賽中的用球，但在這裡它代表前者。標準的足球比賽由兩隊各派11名隊員參與，
50     包括10名球員及1名守門員，互相在長方形的草地球場上互相對抗、互相進攻。比賽目的是盡量將足球射入對方的球門內，每射入一球就可以得到一分，
51     當比賽完畢後，得分最多的一隊則代表勝出。如果在比賽規定時間內得分相同(比賽持續時間為兩個等長的半場，每半場為45分鐘，共90分鐘。)，
52     則須看比賽章則而定，可以抽籤、延長賽再賽(30分鐘延長賽)或互射十二碼罰球等形式比賽分高下。足球比賽中除了守門員可以在己方禁區內利用手部接觸足球外，
53     球場上每名球員只可以利用手以外的身體其他部分控制足球(開界外球例外)。
54   </p>
55   <div class="img_right">
56     
57     <p>籃球</p>
58   </div>
59   <p class="content">
60     籃球是一個由兩隊參與，在一個長方形籃球場進行的球類運動，每隊出場5名隊員，可將球向任何方向傳、投、拍、滾或運，目的是將籃球投入對方球籃得分，
61     並阻止對方獲得球或得分。籃球是世界上最多觀眾的球類活動之一，從1936年起就是奧運的正式競賽項目之一。在正常比賽時間內，球員若將球投進球籃，即可得分。
62     若球員站在三分線上或線內投籃，可以得到二分，若在三分線外投籃，可以得到三分。在比賽結束時，得分最多的球隊獲勝，但若二隊平手，會進行延長賽。在球賽中，
63     掌控球者可以用持續運球(將球彈到地上，再反彈到自己手中)的方式行進或奔跑，也可以將球傳給其他隊員，但若走步(掌控球者沒有運球，且移動超過指定範圍或指定方式)、
64     翻球或二次運球均是違例。<br>
65     籃球中的違例稱為犯規。個人犯規是指個人違例的情形，若單場中個人犯規累計超過一定次數，此場比賽就需離場休息。若一球隊正要投籃時，防守球隊犯規，
66     進攻球隊可以罰球的機會。有時也會出現技術犯規，多半是球員或教練出現非體育道德行為時，此時另一隊可以罰球，而且罰球後有控球權。<br>
67     籃球的投籃、傳球、運球及籃板球均有其特殊的技巧，球員特定的籃球位置，在進攻或防守時也都有特定的隊形。一般而言最高的球員會是中鋒、大前鋒及小前鋒，
68     較矮的球員或是控球技巧及速度都很好的球員則會是控球後衛或是得分後衛。
69   </p>
70 </body>
71 </html>
```

文繞圖程式碼結果



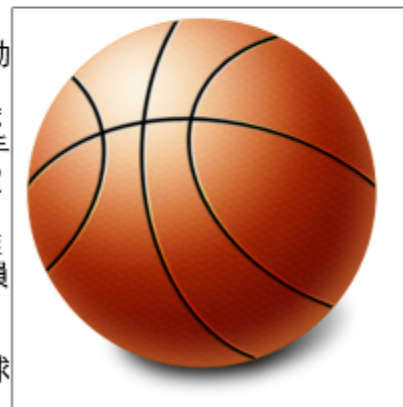
足球

足球是足球運動或足球比賽的簡稱；當然它也指足球比賽中的用球，但在這裡它代表前者。標準的足球比賽由兩隊各派11名隊員參與，包括10名球員及1名守門員，互相在長方形的草地球場上互相對抗、互相進攻。比賽目的是盡量將足球射入對方的球門內，每射入一球就可以得到一分，當比賽完畢後，得分最多的一隊則代表勝出。如果在比賽規定時間內得分相同(比賽持續時間為兩個等長的半場，每半場為45分鐘，共90分鐘。)，則須看比賽章則而定，可以抽籤、延長賽再賽(30分鐘延長賽)或互射十二碼罰球等形式比賽分高下。足球比賽中除了守門員可以在己方禁區內利用手部接觸足球外，球場上每名球員只可以利用手以外的身體其他部分控制足球(開界外球例外)。

籃球是一個由兩隊參與，在一個長方形籃球場進行的球類運動，每隊出場5名隊員，可將球向任何方向傳、投、拍、滾或運，目的是將籃球投入對方球籃得分，並阻止對方獲得球或得分。籃球是世界上最多人觀看的球類活動之一，從1936年起就是奧運的正式競賽項目之一。在正常比賽時間內，球員若將球投進球籃，即可得分。若球員站在三分線上或線內投籃，可以得到二分，若在三分線外投籃，可以得到三分。在比賽結束時，得分最多的球隊獲勝，但若二隊平手，會進行延長賽。在球賽中，掌控球者可以用持續運球（將球彈到地上，再反彈到自己手中）的方式行進或奔跑，也可以將球傳給其他隊員，但若走步（掌控球者沒有運球，且移動超過指定範圍或指定方式）、翻球或二次運球均是違例。

籃球中的違例稱為犯規。個人犯規是指個人違例的情形，若單場中個人犯規累計超過一定次數，此場比賽就需離場休息。若一球隊正要投籃時，防守球隊犯規，進攻球隊可以罰球的機會。有時也會出現技術犯規，多半是球員或教練出現非體育道德行為時，此時另一隊可以罰球，而且罰球後有控球權。

籃球的投籃、傳球、運球及籃板球都有其特殊的技巧，球員特定的籃球位置，在進攻或防守時也都有特定的隊形。一般而言最高的球員會是中鋒、大前鋒及小前鋒，較矮的球員或是控球技巧及速度都很好的球員則會是控球後衛或是得分後衛。



籃球

相對位置、絕對位置

- CSS position位置屬性用來定義元素要在網頁的哪個位置呈現。
- 位置屬性語法：
`{position: 位置參數;}`
- 位置參數有absolute、fixed、relative、static等。

相對位置、絕對位置

○ 位置參數：

1. `absolute`：絕對位置，當網頁往下拉時，元素會跟著改變位置。
2. `fixed`：元素位置固定，相對於瀏覽器而定位。
3. `relative`：相對位置，相對於其他元素的位置。
(以上三項的元素位置由`top`、`bottom`、`left`、`right`決定。)
4. `static`：預設值，忽略`top`、`bottom`、`left`、`right`。

相對位置、絕對位置程式碼

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="utf-8">
5      <title>Chapter 7</title>
6  <style>
7      div.relative {
8          position: relative;
9          left: 50px;
10         border: 8px dashed #66CCFF;
11     }
12 </style>
13 </head>
14 <body>
15     <h1>position: relative;</h1>
16     <p>此位置參數為相對位置</p>
17 <div class="relative">
18         相對於此元素正常的位置，他被調整成離開正常位置距離左邊50px。
19     </div>
20 </body>
21 </html>
```

相對位置、絕對位置程式碼結果

```
position: relative;
```

此位置參數為相對位置

相對於此元素正常的位置，他被調整成離開正常位置距離左邊50px。

相對位置、絕對位置程式碼

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title>Chapter 7</title>
6      <style>
7        div.relative {
8          position: relative;
9          width: 400px;
10         height: 200px;
11         background-color: #FF0000;
12       }
13       div.absolute {
14         position: absolute;
15         top: 40px;
16         right: 20px;
17         width: 100px;
18         height: 100px;
19         background-color: #00CC00;
20       }
21     </style>
22   </head>
23   <body>
24     <h1>position: absolute;</h1>
25     <p>此為兩個div區塊間的位置關係，紅色大的div區塊的位置設定為「position: relative;」，主要目的是讓裡面的綠色小區塊在使用
26       「position: absolute;」時，可以依據紅色的大div區塊邊框來設定位置，這樣兩個div區塊彼此間才會有所關連，否則區塊會根據父層或螢幕為基準。
27       綠色小區塊因為是根據紅色大區塊來呈現位置，並將 position 設為 absolute，所以當我們移動紅色大區塊的位置時，綠色小區塊也會跟著被移動，
28       而且兩個區塊彼此間的位置是不會改變的。</p>
29     <div class="relative">相對位置
30       <div class="absolute">絕對位置</div>
31     </div>
32   </body>
33 </html>
```

相對位置、絕對位置程式碼結果

`position: absolute;`

此為兩個div區塊間的位置關係，紅色大的div區塊的位置設定為「`position: relative;`」，主要目的是讓裡面的綠色小區塊在使用「`position: absolute;`」時，可以依據紅色的大div區塊邊框來設定位置，這樣兩個div區塊彼此間才會有所關連，否則區塊會根據父層或螢幕為基準。綠色小區塊因為是根據紅色大區塊來呈現位置，並將 `position` 設為 `absolute`，所以當我們移動紅色大區塊的位置時，綠色小區塊也會跟著被移動，而且兩個區塊彼此間的位置是不會改變的。

相對位置

絕對位置



相對位置、絕對位置程式碼

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="utf-8">
5      <title>Chapter 7</title>
6  <style>
7      div.fixed {
8          position: fixed;
9          top: 0;
10         right: 0;
11         width: 300px;
12         border: 5px dotted #CC99FF;
13     }
14 </style>
15 </head>
16 <body>
17     <h1>position: fixed;</h1>
18     <p>此元素被設為固定位置，代表即使頁面上下移，他都會停留在同一個位置</p>
19     <div class="fixed">
20         此元素被設為固定位置
21     </div>
22 </body>
23 </html>
```

相對位置、絕對位置程式碼結果

```
position: fixed;
```

此元素被設為固定位置

此元素被設為固定位置，代表即使頁面上下移，他都會停留在同一個位置

相對位置、絕對位置程式碼

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title>Chapter 7</title>
6      <style>
7        div.static {
8          position: static;
9          border: 5px double #FF0066;
10        }
11      </style>
12    </head>
13    <body>
14      <h1>position: static;</h1>
15      <p>此元素被設為預設值，不會受到上下左右屬性的影響，根據瀏覽器預設的配置自動排版在頁面上。</p>
16      <div class="static">
17        此元素被設為預設值
18      </div>
19    </body>
20  </html>
```

相對位置、絕對位置程式碼結果

`position: static;`

此元素被設為預設值，不會受到上下左右屬性的影響，根據瀏覽器預設的配置自動排版在頁面上。

此元素被設為預設值

疊層

- 我們把電腦螢幕左右方向當作x軸，上下方向當作y軸，則我們面對電腦螢幕的這個方向為z軸。
- z-index 可以用來設置元素的z方向位置，z-index 數字越大的在越上面，數字越小的則在越下面。

疊層

- 要使用多個元素重疊的效果，必須使用到 `position` 來設定你的元素位置，若沒有設定則元素會在網頁上呈現為一般的區塊，沒辦法重疊起來。
- `position` 可設定為 `static`、`absolute`、`relative`、`fixed` 以及 `top`、`right`、`bottom`、`left`，其他的目前在 `z-index` 沒有效果。

疊層程式碼

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title>Chapter 7</title>
6      <style>
7        div.top{
8          position: absolute;
9          left: 0px;
10         top: 0px;
11         height: 200px;
12         width: 200px;
13         background-color: #6666FF;
14         z-index: 1;
15       }
16       p.under{
17         height:300px;
18       }
19       div.under{
20         position: absolute;
21         left: 0px;
22         top: 300px;
23         height: 200px;
24         width: 200px;
25         background-color: #6666FF;
26         z-index: -1;
27       }
28       p.top{
29         height: 300px;
30       }
31     </style>
32   </head>
33   <body>
34     <div class="top"> </div>
35     <p class="under">此處為文字</p>
36     <div class="under"> </div>
37     <p class="top">此處為文字</p>
38   </body>
39 </html>
```

疊層程式碼 結果

- 上圖是將色塊的z-index設為1，圖片遮住了文字。
- 下圖是將色塊的z-index設為-1，圖片在文字的下方。

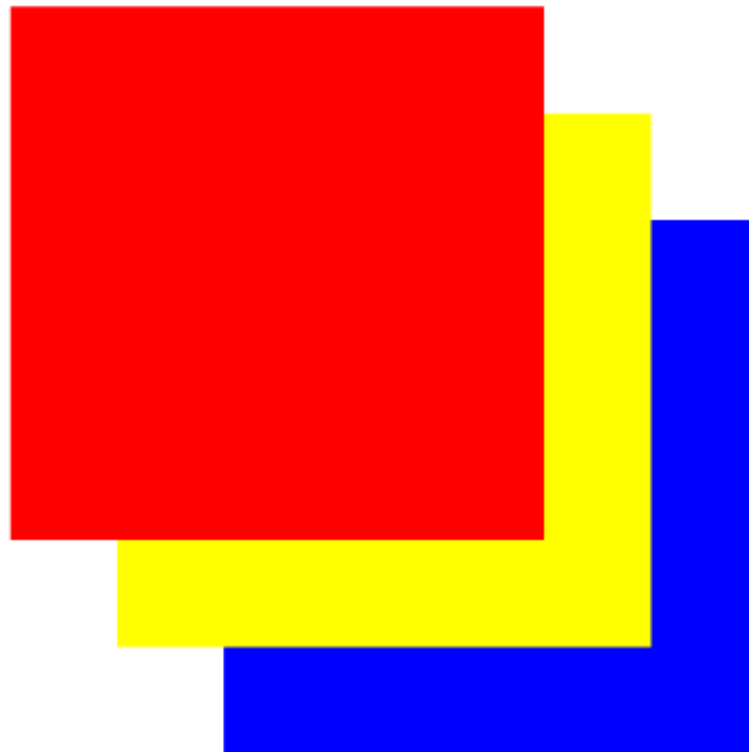
此處為文字

疊層程式碼

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title>Chapter 7</title>
6      <style>
7        div.top{
8          position: absolute;
9          left: 20px;
10         top: 20px;
11         height: 200px;
12         width: 200px;
13         background-color: #FF0000;
14         z-index: 3;
15       }
16       div.middle{
17         position: absolute;
18         left: 60px;
19         top: 60px;
20         height: 200px;
21         width: 200px;
22         background-color: #FFFF00;
23         z-index: 2;
24       }
25       div.under{
26         position: absolute;
27         left: 100px;
28         top: 100px;
29         height: 200px;
30         width: 200px;
31         background-color: #0000FF;
32         z-index: 1;
33       }
34     </style>
35   </head>
36   <body>
37     <div class="top"> </div>
38     <div class="middle"> </div>
39     <div class="under"> </div>
40   </body>
41 </html>
```

疊層程式碼結果

- 紅色色塊的
z-index=3
- 黃色色塊的
z-index=2
- 藍色色塊的
z-index=1
- 由此可知紅色
色塊在最上方，
藍色色塊在最
下方。



參考資料

- <http://www.w3schools.com/default.asp>
- <http://king971119.blogspot.tw/2010/08/css.html>
- <http://www.wibibi.com/info.php?tid=156>
- <http://www.webtech.tw/info.php?tid=36>

TO BE CONTINUED