A

HTML5繪圖標籤與Canvas API

- A-1 HTML5 繪圖標籤
- A-2 CanvasRenderingContext2D 物件

A-1 HTML5 繪圖標籤

HTML5 的繪圖功能是在<canvas>標籤建立長方形畫布，然後使用 JavaScript 程式碼呼叫 Canvas API 來繪出 2D 圖形。

HTML5 的<canvas>標籤：AppA_1.html

HTML5 提供<canvas>標籤在網頁建立可供繪圖的長方形區域，即畫布，然後就可以使用 JavaScript 程式碼在畫布上繪圖，請注意！<canvas>標籤只是建立繪圖區域的畫布容器，一定需要使用 JavaScript 程式碼才能在畫布上繪圖或是載入圖片。

在實務上，<canvas>標籤一定要指定 id 屬性，以便 JavaScript 程式碼可以取得此 DOM 元素，如下所示：

```
<canvas id="output" style="border: solid red;"
        width="200" height="100"></canvas>
```

上述標籤碼建立 id 屬性值 output 的 HTMLCanvasElement 物件，height 是 Canvas 元素的高，預設值是 300；width 是 Canvas 元素的寬，預設值也是 300。其執行結果可以看到繪圖區域的 Canvas 元素，紅色框是使用 CSS 樣式所加上的框線，如下圖所示：

在 Canvas 元素繪圖：AppA_1a.html

HTML 的<canvas>標籤只是一張指定尺寸的畫布，我們需要使用 JavaScript 程式碼才能在此畫布上繪圖，首先取得 Canvas 元素的 DOM 物件，如下所示：

```
const canvas = document.getElementById("output");
```

```
const ctx = canvas.getContext("2d");
```

上述程式碼取得 id 屬性值為 output 的 HTMLCanvasElement 物件後，呼叫 getContext() 方法取得繪圖物件 ctx，參數 "2d" 是 2D 繪圖，可以回傳 CanvasRenderingContext2D 物件（詳見 A-2 節），這就是在 Canvas 畫布上建立的繪圖環境，所謂繪圖就是使用此物件的方法和屬性來繪出圖形。例如：替畫布填滿長方形的背景色彩，如下所示：

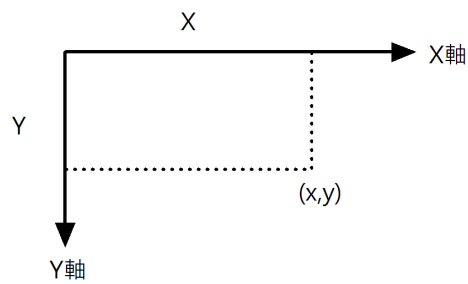
```
ctx.fillStyle = "yellow";
ctx.fillRect(0, 0, canvas.width, canvas.height);
```

上述程式碼的 fillStyle 屬性指定填滿樣式為黃色後，呼叫 fillRect()方法填滿長方形的背景色彩，即黃色的背景色彩。其執行結果可以看到紅色框線和黃色背景色彩的繪圖區域，如下圖所示：

A-2 CanvasRenderingContext2D 繪圖物件

Canvas 元素的繪圖功能是使用 CanvasRenderingContext2D 物件的屬性和方法，可以在 Canvas 畫布上繪出路徑、長方形、圓形、顯示圖檔和繪出文字內容。

基本上，Canvas 畫布是一個長方形區域，其左上角是原點，座標是(0, 0)，X 軸從左到右；Y 軸由上到下，其單位是「像素」（Pixels），如下圖所示：



上述座標系統可以使用 HTMLCanvasElement 物件的 width 和 height 屬性取得畫布的寬和高。CanvasRenderingContext2D 物件提供屬性來指定色彩、文字和線條樣式，其相關屬性的說明如下表所示：

屬性	說明
strokeStyle	指定使用空心形狀的樣式，只繪出框線，例如：長方形框線，其值是框線的 CSS 色彩值，預設值是#00000
fillStyle	指定使用填滿形狀的樣式，例如：填滿的長方形，其值就是 CSS 色彩的填滿值，預設值是#00000
lineWidth	指定線條寬度，預設值是 1.0
lineCap	指定如何繪出線條的端點，其值可以是 butt、round 或 square，預設值是 butt
font	指定目前使用的字型，其值是 CSS 的 font 屬性
textAlign	指定文字內容的對齊方式，其值可以是 start、end、left、right 或 center，預設值是 start
textBaseline	指定文字內容的基準線，其值可以是 top、hanging、middle、alphabetic、ideographic 或 bottom，預設值是 alphabetic

繪出長方形：AppA_2.html

CanvasRenderingContext2D 物件提供 strokeRect()和 fillRect() 兩種方法來繪出長方形或填滿長方形，其相關方法說明，如下表所示：

方法	說明
strokeRect(x, y, w, h)	使用 strokeStyle 屬性的樣式繪出長方形，(x,y)是左上角座標，w 是長方形的寬；h 是高
fillRect(x, y, w, h)	使用 fillStyle 屬性的樣式繪出填滿的長方形，參數和 strokeRect()方法相同
clearRect(x, y, w, h)	清除長方形區域的內容，參數和 strokeRect()方法相同

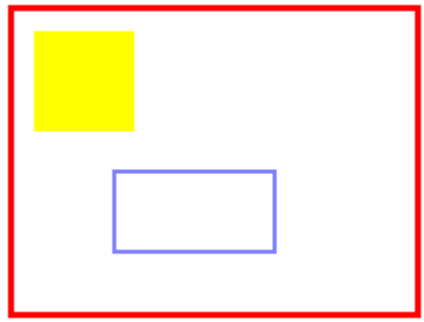
在 JavaScript 程式碼只需指定 fillStyle 屬性的填滿色彩，就可以使用 fillRect()方法繪出填滿色彩的長方形，如下所示：

```
ctx.fillStyle = "yellow";
ctx.fillRect(10, 10, 50, 50);
```

上述程式碼繪出填滿色彩的正方形，因為尺寸的寬和高都是 50。同理，繪出沒有填滿的長方形是使用 strokeStyle 屬性配合 strokeRect()方法，如下所示：

```
ctx.strokeStyle = "blue";
ctx.strokeRect(50, 80, 80, 40);
```

JavaScript 程式：AppA_2.html 在建立<canvas>標籤後，使用程式碼繪出一個填滿的黃色正方形和一個藍色框線的長方形，如下圖所示：

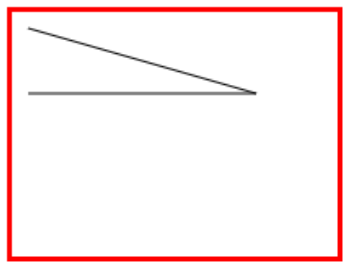


繪出直線：AppA_2a.html

在 Canvas 畫布繪出直線是使用路徑（Path）觀念，如同在白紙上手繪各種直線或形狀一般。首先呼叫 moveTo()方法移至開始繪圖的起點（下筆點），然後從此起點開始，呼叫 lineTo()方法繪出至參數座標的直線（繪至目的點），如下所示：

```
ctx.moveTo(10,10);
ctx.lineTo(150,50);
ctx.lineTo(10,50);
ctx.stroke();
```

上述程式碼可以建立 3 個端點之間的路徑，最後呼叫 stroke()方法以 strokeStyle 屬性的樣式繪出 3 個端點之間的 2 條直線，其執行結果如下圖所示：



繪出多邊形：AppA_2b.html

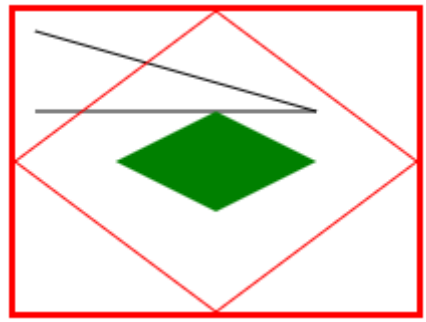
如果建立的路徑有回到起點，就是繪出形狀，例如：多邊形，在這一節的 JavaScript 程式是繼續 AppA_2a.html，因為我們在同一張畫布已經繪出二條直線，所以需要使用 beginPath()方法重新建立一條新路徑，接著就可以呼叫 moveTo()和 lineTo()方法繪出菱形，如下所示：

```
…
ctx.beginPath();
ctx.strokeStyle = "red";
ctx.moveTo(100, 0);
ctx.lineTo(0, 75);
ctx.lineTo(100, 150);
ctx.lineTo(200, 75);
ctx.closePath();
ctx.stroke();
```

上述程式碼建立菱形的路徑，closePath()方法是將最後一個端點連接至起點來圍成形狀，最後呼叫 stroke()方法繪出菱形。然後是繪出填滿的菱形，如下所示：

```
ctx.beginPath();
ctx.fillStyle = "green";
ctx.moveTo(100, 50);
ctx.lineTo(50, 75);
ctx.lineTo(100, 100);
ctx.lineTo(150, 75);
ctx.closePath();
ctx.fill();
```

上述程式碼最後是呼叫 fill()方法，所以是填滿菱形。其執行結果可以繪出 2 條直線（這是在 AppA_2a.html 已經繪出的直線）、一個紅色菱形和填滿的綠色菱形（在 AppA_2b.html 繪出），如下圖所示：

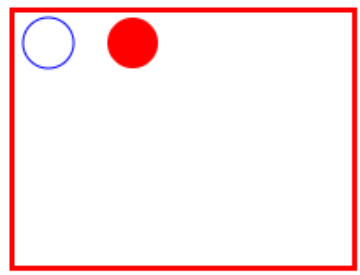


繪出圓形：AppA_2c.html

在 Canvas 畫布繪出圓形或填滿圓形是使用 arc()方法配合 stroke()和 fill()方法，如下所示：

```
ctx.beginPath();
ctx.arc(20,18,15,0,Math.PI*2,true);
ctx.closePath();
ctx.stroke();
```

上述程式碼使用 arc()方法繪出圓形，圓心座標是(20, 18)，半徑 15，孤度是 0~Math.PI*2 即一整圈，所以 stroke()方法繪出的孤形就是圓形，如果使用 fill()方法就是填滿圓形。其執行結果可以分別繪出一個圓形和一個填滿紅色的圓形，如下圖：



顯示圖片：AppA_2d.html

JavaScript 程式可以使用 Image()建構函數建立 Image 物件後，使用 drawImage()方法在 Canvas 畫布上顯示圖片，即在參數座標(x, y)的位置顯示參數 img 物件的圖片，最後 2 個參數是尺寸的寬和高，如下所示：

```
let img = new Image();
img.onload = function() {
      ctx.drawImage(this, 0, 0, 200, 150);
}
img.src = "table.png";
```

上述程式碼建立 Image 物件 img 後，註冊 onload 事件處理函數，當圖片載入後就顯示圖片內容，即呼叫 drawImage()方法繪圖圖片，最後 src 屬性值是圖檔的 URL 網址。其執行結果可以在畫布顯示 table.png 圖檔，如下圖所示：

繪出文字內容：AppA_2e.html

在 Canvas 畫布繪出文字內容可以使用 fillText()或 strokeText()方法，在呼叫之前，請先指定文字對齊和字型的相關樣式，如下所示：

```
ctx.font = "italic 20px 細明體";
ctx.textBaseline = "top";
ctx.textAlign = "center";
```

上述程式碼指定字型尺寸、字體、樣式和對齊方式，然後使用 fillText()或 strokeText()方法繪出文字內容，如下所示：

```
ctx.fillStyle = "red";
ctx.fillText("JavaScript 程式設計", 100, 10);
```

上述程式碼呼叫 fillText()方法，這是使用 fillStyle 屬性的樣式，可以在座標(100, 10)繪出第 1 個參數的字串內容，同理，strokeText()方法是使用 strokeStyle 屬性的樣式。其執行結果可以在 Canvas 畫布顯示 2 行文字內容，如下圖所示：