

# GD繪圖與圖片處理

6-1 GD繪圖

6-2 實用的圖片函式

4A355077 劉子寬

4A355087 王子凡

# 6-1 GD 繪圖

- 6-1-1 建立空圖片

在使用GD繪製線條、圖形或文字之前，必須先建立空圖片，它就像是一塊畫布一樣。我們可以使用 `imagecreate()` 或 `imagecreatetruecolor()` 函示建立空圖片，其語法如下：

```
1  <?php
2  imagecreate(int x_size,int y_size)
3  //x_size為圖片的寬度，y_size為圖片的高度
4  imagecreatetruecolor(int x_size,int y_size)
5  //x_size為圖片的寬度，y_size為圖片的高度
6
7  ?>
```

若`imagecreate()`或`imagecreatetruecolor()` 函式成功建立該圖片，就會傳回代表該圖片的圖片識別字 (image identifier)，這兩個函式的差別在於前者只支援256色，而後者可支援百萬色，若要繪製色彩單純的圖形，可以使用`imagecreate()`函式，否則就用`imagecreatetruecolor()` 函式。

下列分別建立800X600及1024X768的空圖片，前者支援256色，後者支援百萬色:

```
$im1 = imagecreate(800,600);
```

```
//單位為像素，背景色彩預設為黑色
```

```
$im2 = imagecreatetruecolor(1024,768);
```

```
//單位為像素，背景色彩預設為黑色
```

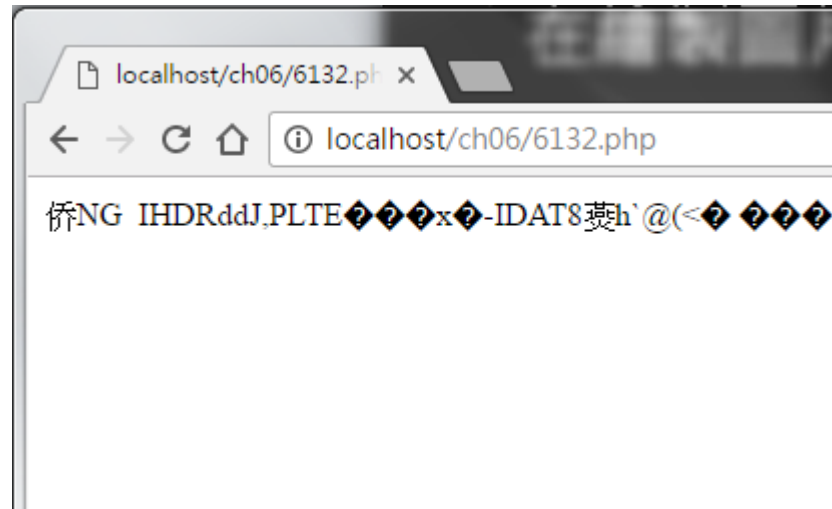
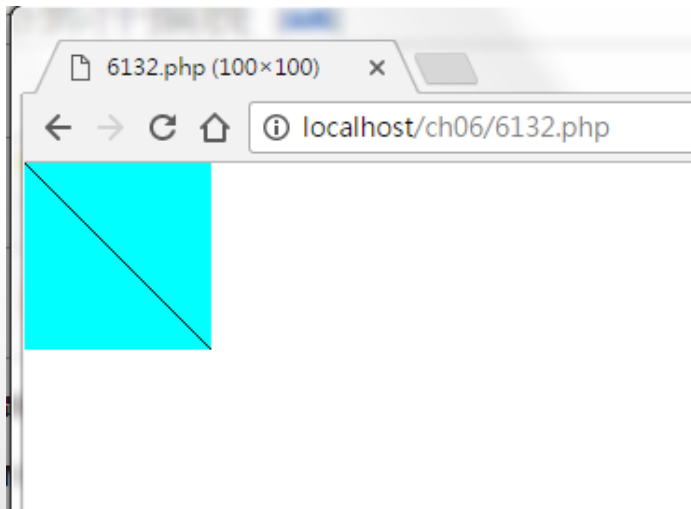
## 在繪製圖片之前，要先說明如何輸出圖片

繪製的圖片皆儲存在電腦的記憶體裡，PHP提供了下列幾個函式，可以將圖片傳送至用戶端瀏覽器或儲存在伺服器裡。函式會將圖片傳送至用戶端瀏覽器或儲存在伺服器，參數filename不寫的話，表示將圖片輸出到瀏覽器，此時，我們必須指定輸出至瀏覽器的MIME類型。

- ◆ **MIME**，多用途網際網路郵件擴展(Multipurpose Internet Mail Extensions)是一個網際網路標準，它擴展了電子郵件標準，使其能夠支援一些非英語字符消息和二進位文件，圖像，聲音等非文字消息。

在這邊我們透過MIME告知瀏覽器，輸出的圖片式何種格式，來避免瀏覽器無法解析。

下面是有無告知的差異



## GIF格式

```
imagegif(resource image[,string filename]);
```

下面為PHP函式

```
header("content-type:image/gif");
```

```
imagegif($im);
```

若要儲存在伺服器，則可以寫成:

```
imagegif($im," image/simple.gif" );
```

## PNG格式

```
imagepng(resource image[,string filename]);
```

下面為PHP函式

```
header("content-type:image/png");
```

```
imagepng($im);
```

若要儲存在伺服器，則可以寫成:

```
imagepng($im," image/simple.png" );
```

由上面可以知道GIF及PNG兩者之間並無太大差異，只是輸出的格式不同而已。

## JPG格式

```
imagejpeg(resource image[,string filename[,int quality]]);
```

JPG格式跟上面兩個格式有個不同的地方就是他能決定輸出圖片的品質，參數 `quality` 用來設定圖片品質，為0~100的數字，預設值為75，數字越大，品質就越好，檔案也就越大。

下面的敘述是將圖片傳送至瀏覽器，品質為90:

```
header("content-type:image/jpeg");
```

```
imagejpeg( $im , null , 90 );
```

若選擇儲存在伺服器，且使用預設品質(75)，則可以寫成:

```
imagejpeg($im , "images/simple.jpg");
```

若選擇儲存在伺服器，且使用最高圖片品質，則可以寫成:

```
imagejpeg($im , "images/simple.jpg " , 100 );
```

## 釋放記憶體

圖片輸出完畢之後，我們必須使用 `imagedestroy()` 來釋放圖片占用的記憶體空間，其函式如下:

```
imagedestroy(resource image);
```

- 6-1-2 指派色彩

建立空圖片後，還要用 `imagecolorallocate()` 函式指派該圖片可以使用的色彩，才能進一步繪製線條、圖片及文字，其語法如下：

```
imagecolorallocate(resource image , int red , int green , int blue)
```

參數 `image` 為欲指派色彩的圖片，參數 `red` , `green` , `blue` 為紅、綠、藍三色，必須介於 0~255 之間，其中 0 表示缺少該顏色，255 表示最多該顏色，故

`imagecolorallocate($im,0,0,0);` 表示指派黑色

`imagecolorallocate($im,255,255,255);` 表示指派白色

下面是敘述建立一個200X200、支援百萬色的空圖片，並指定該圖可使用白、紅兩色:

```
<? php
```

```
$im = imagecreatetruecolor=(200,200);
```

```
$white = imagecolorallocate($im,255,255,255);
```

```
$red = imagecolorallocate($im,255,0,0);
```

```
? >
```

- 使用imagecreatetruecolor()建立一個200X200、支援百萬色的空圖片，**\$im**就是指該圖片。
- 使用imagecolorallocate()指派紅白兩色給空圖片使用，變數**\$white**、**\$red** 分別代表白色及紅色。
- 請注意，若是用imagecreate()建立圖片，那第一個分配的色彩(白色)會自動成為背景顏色，以此例來說，背景會是黑色，因為是使用imagecreatetruecolor()函式來建立該圖。



- 6-1-3 繪製線條、圖形與文字

- 繪製橢圓形

我們可以使用imageellipse()來繪製橢圓形，語法如下， 他會以色彩 *color*在畫布image的座標(cx,cy)處繪製寬度為w，高度為h的橢圓形，若寬度及高度相同，則為圓形

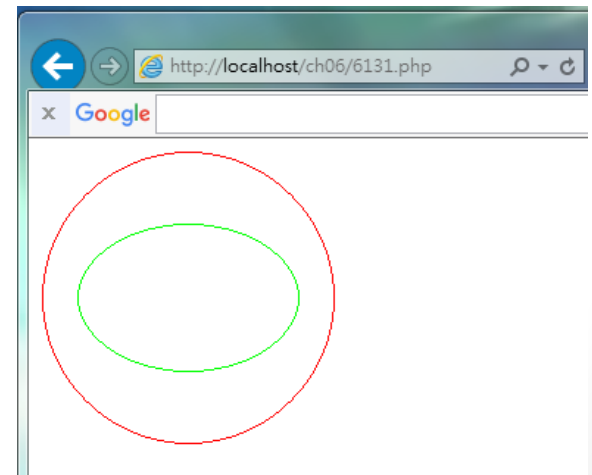
```
imageellipse(resource image , int cx, int cy, int w ,int h ,int color);
```

下面是繪製兩個橢圓形:

```
<? php
header("Content-type:image/png");
$im = imagecreate(200,200);
$background = imagecolorallocate($im,255,255,255);
$red = imagecolorallocate($im,255,0,0);
$green = imagecolorallocate($im,0,255,0);

imageellipse($im,100,100,199,199,$red);
imageellipse($im,100,100,150,100,$green);

imagepng($im);
imagedestroy($im);
? >
```



## 延伸-填滿色彩

如果要讓繪製出來的圓形填滿指定的色彩，以上面為例，只需要將函式改寫成:

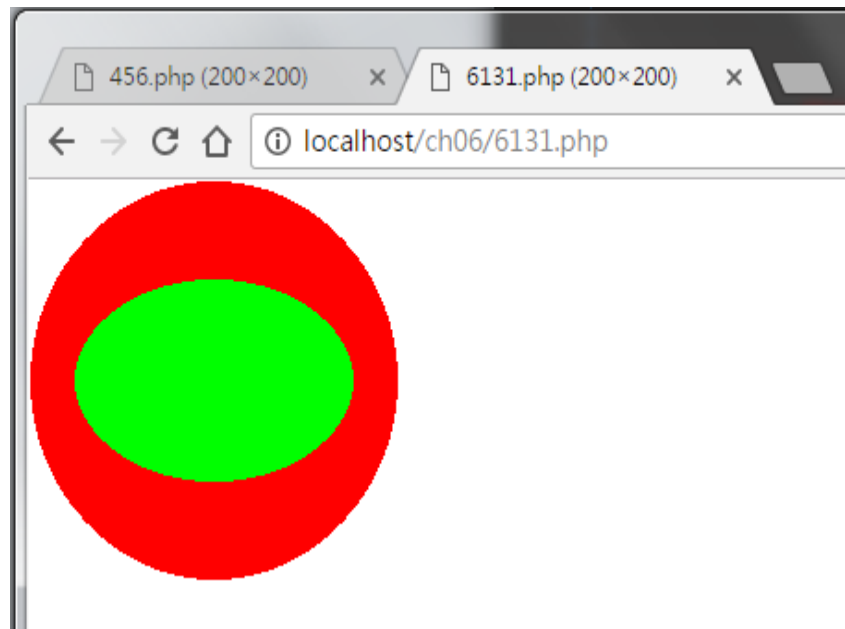
```
imagefilledellipse(resource image , int cx, int cy, int w ,int h ,int color);
```

在上面的範例加上**filled**，就會變成下面的樣子:

```
imagefilledellipse($im,100,100,199,199,$red);
```

```
imagefilledellipse($im,100,100,150,100,$green);
```

以此類推，如果要把指定的圖形填滿顏色，只需要在函式裡面加上**filled**，就能夠把圖形填滿。



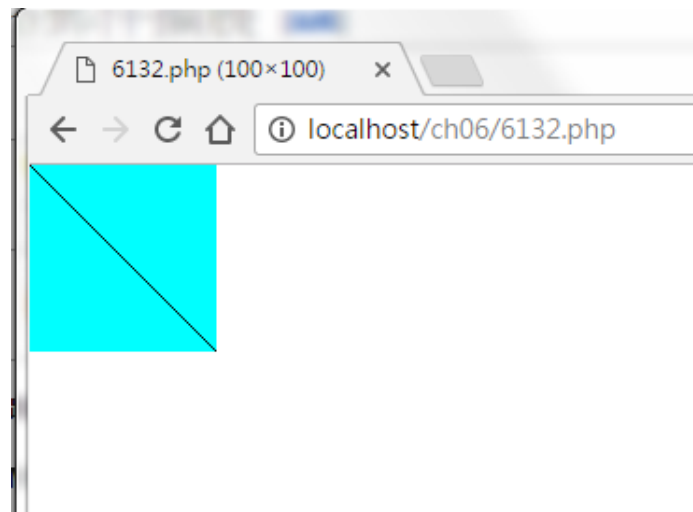
## ➤ 繪製直線

我們可以使用 `imageline()` 來繪製線條，語法如下，它會以色彩 `color` 在畫布 `image` 繪製從座標 `(x1,y1)` 到座標 `(x2,y2)` 的直線：

```
imageline(resource image,int x1,int y1,int x2,int y2,int color)
```

下面的敘述為使用黑色繪製從座標 `(0, 0)` 到座標 `(100, 100)` 的直線：

```
<? php  
$im = imagecreate(100,100);  
$background = imagecolorallocate($im,0,255,255);  
$black= imagecolorallocate($im,0,0,0);  
//繪製直線  
imageline($im,0,0,100,100,$black);  
  
header("Content-type:image/png");  
imagepng($im);  
imagedestroy($im);  
? >
```



## ➤ 繪製多邊形

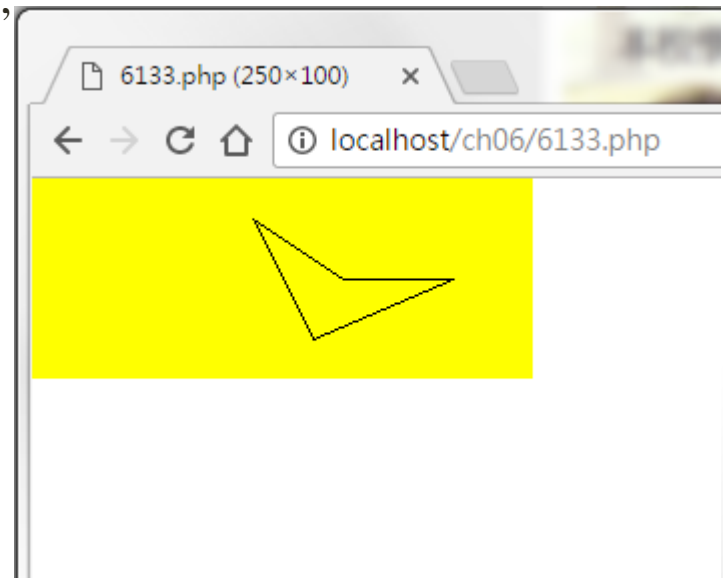
接下來用 `imagepolygon()` 來繪製多邊形，語法如下，它會以色彩 `color` 在畫布 `image` 繪製從 `point` 陣列指定之點所構成的多邊形，`num_points` 則是用來指定多邊形的端點數目：

```
imagepolygon(resource image , array point, int num_points , int color)
```

例如下面是敘述用黑色線條繪製由座標 (210,50) (140,80) (110,20) (155,50) 所構成的多邊形，由於是4個座標，所以是四邊形：

```
<?php
$point = array(210,50,140,80,110,20,155,50);
$im = imagecreate(250,100);
$background = imagecolorallocate($im,255,255,0);
$black= imagecolorallocate($im,0,0,0);
//繪製多邊形
imagepolygon($im,$point,4,$black);

header("Content-type:image/png");
imagepng($im);
imagedestroy($im);
?>
```



## ➤ 繪製矩形

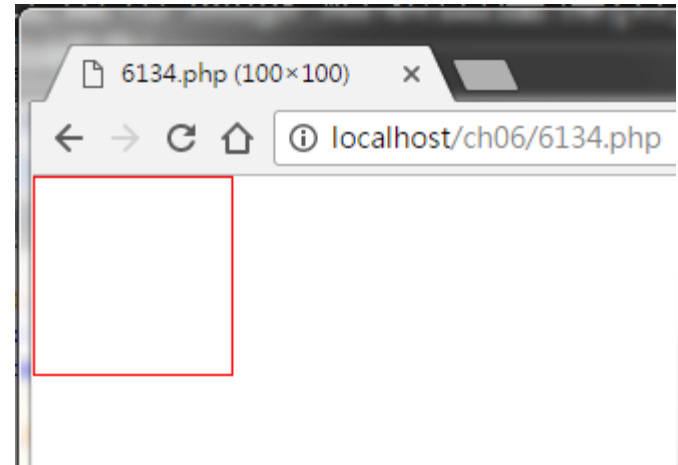
矩形的部分則用 `imagerectangle()` 來達成，它會以色彩 `color` 在畫布 `image` 繪製由座標 `(x1,y1)` 與 `(x2,y2)` 作為左上角和右下角端點所構成的矩形。

```
imagerectangle(resource image , int x1 , int y1 , int x2 , int y2 , int color)
```

下面的敘述為使用黑色繪製從座標 `(0, 0)` 到座標 `(99, 99)` 兩點所構成的矩形：

```
<? php
$im = imagecreate(100,100);
$background = imagecolorallocate($im,255,255,255);
$red= imagecolorallocate($im,255,0,0);
//繪製矩形
imagerectangle($im,0,0,99,99,$red);

header("Content-type:image/png");
imagepng($im);
imagedestroy($im);
? >
```



## ➤ 繪製弧線

弧線使用的是 `imagearc()` 來繪圖，由座標  $(cx, cy)$  為中心，繪製寬度  $w$ 、高度  $h$ 、起始角度  $s$ 、結束角度  $e$  (順時針方向) 的弧線。

```
imagearc(resource image , int cx , int cy , int w , int h , int s, int e ,int color)
```

下面的敘述為使用黑色繪製從座標  $(0, 0)$  到座標  $(99, 99)$  兩點所構成的矩形:

```
<? php
```

```
$im = imagecreate(200,200);
```

```
$background = imagecolorallocate($im,255,255,255);
```

```
$green= imagecolorallocate($im,0,255,0);
```

```
//繪製矩形
```

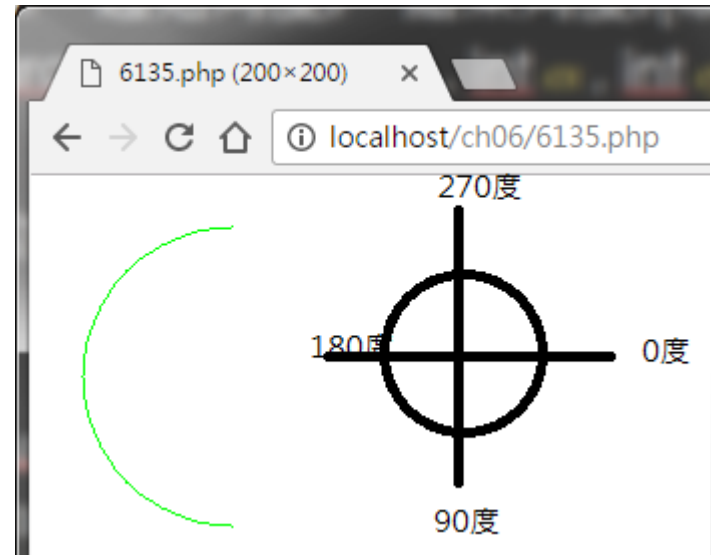
```
imagearc($im,100,100,150,150,90,270,$green);
```

```
header("Content-type:image/png");
```

```
imagepng($im);
```

```
imagedestroy($im);
```

```
? >
```



## ➤ 繪製字元

imagechar()→橫向字元

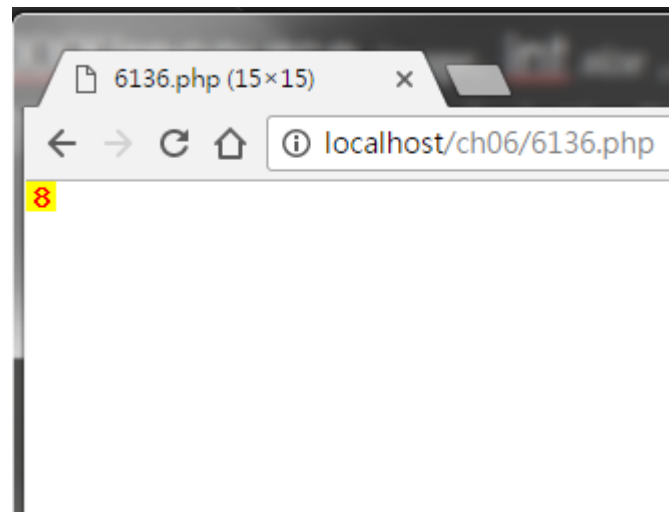
imagecharup()→直向字元

語法如下:

imageXXXX(resource image, int size , int x , int y , string c ,int color)  
x,y為起始座標，size為字元大小(1~5)，c為輸出的字元

```
<? php
$im = imagecreate(15,15);
$background = imagecolorallocate($im,255,255,0);
$red= imagecolorallocate($im,255,0,0);
//繪製橫向字元
imagechar($im,5,4,0,"8",$red);

header("Content-type:image/png");
imagepng($im);
imagedestroy($im);
? >
```



## ➤ 繪製文字

imagestring()→橫向文字

imagestringup()→直向文字

語法如下:

imageXXXX(resource image, int size , int x , int y , string s ,int color)

x,y為起始座標，size為字元大小(1~5)，s為輸出的文字。

**注意!! 此函數並不支援中文!!**

```
<? php
```

```
$im = imagecreate(400,20);
```

```
$background = imagecolorallocate($im,255,255,255);
```

```
$textcolor= imagecolorallocate($im,255,0,255);
```

```
//繪製橫向文字
```

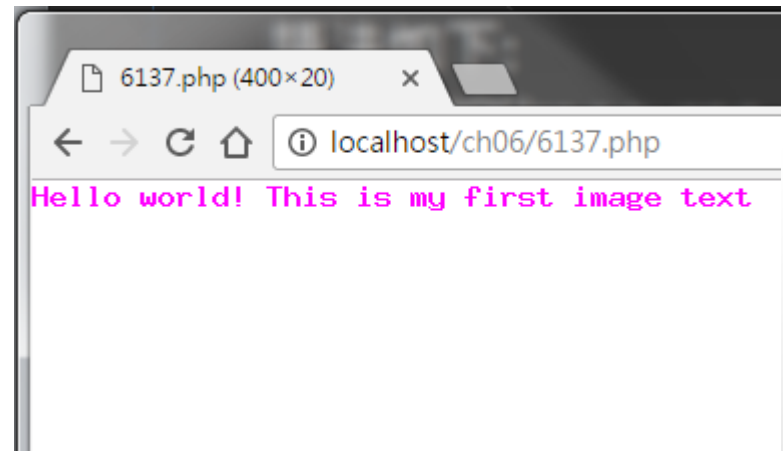
```
imagestring($im,5,0,0, "Hello world! This is my first image text",$textcolor);
```

```
header("Content-type:image/png");
```

```
imagepng($im);
```

```
imagedestroy($im);
```

```
?>
```





## ➤ 繪製 TrueType 文字

TrueType 文字跟上面不同的地方就在於它是可以輸出中文的語法如下:

```
imagefttext( resource image, float size ,float angle, int x , int y , string s ,int color, string fontfile , string text )
```

x,y 為起始座標，size 為字型大小(單位為點數)，text 為輸出的文字，angle 為文字的角度(0 表示由左向右的書寫文字)，字型檔為 fontfile 。

```
<? php
```

```
$im = imagecreate(500,28);
```

```
$background = imagecolorallocate($im,255,255,255);
```

```
$textcolor= imagecolorallocate($im,0,0,255);
```

```
$string = "哇哈哈，imagefttext() 函式可以繪出中文字喔...";
```

```
imagefttext($im,16,0,0,20,$textcolor,"simhei.ttf",$string);
```

```
header("Content-type:image/png");
```

```
imagepng($im);
```

```
imagedestroy($im);
```

```
? >
```



# 6-2實用的圖片函式

## ■6-2-1取得圖片格式

`exif_imagetype()`函式會透過檔案的第一個字原來取得圖片格式，而不是透過副檔名。參數`filename`是檔案的相對位址或絕對位址(URL)，傳回值為 `int` 型別，若指定非圖片的檔案，則會傳回`FALSE`並產生錯誤的訊息。

```
<?php
```

```
    $imageType = exif_imagetype("images/pic1.jpg");
```

```
    if (!$imageType)
```

```
        echo "這個檔案不是圖片";
```

```
    elseif ($imageType == IMAGETYPE_JPEG)
```

```
        echo "這張圖片的格式為JPG";
```

```
?>
```



## ■ 取得圖片大小與格式

getimagesize()用來取得圖片大小與格式，傳回值包含4個元素的陣列，若指定非圖片的檔案，則會傳回FALSE並產生錯誤的訊息。

- ✘ 第1個元素代表圖片寬度，單位為像素(pixel)。
- ✘ 第2個元素代表圖片高度，單位為像素(pixel)。
- ✘ 第3個元素為圖片格式，與exif\_imagetype()相同，若只取圖片格式，exif\_imagetype()較有效率。
- ✘ 第4個元素會以 width= "xxx" height= "xxx" 的形式描述圖片的寬度與高度。

```
<?php
```

```
    $size = getimagesize("images/pic1.jpg");
```

```
    echo "圖片寬度：$size[0]<br>";
```

```
    echo "圖片高度：$size[1]<br>";
```

```
    echo "圖片格式：$size[2]<br>";
```

```
    echo "圖片長寬字串：$size[3]";
```

```
?>
```



- 6-2-3 讀取外部圖片

我們可以分別使用函式讀取GIF、PNG、JPEG圖形檔

`imagecreatefromXXX(string filename)`

`filename`表示要讀取的圖片路徑及名字，可以是相對位址或絕對位址(URL)，傳回的值代表圖片的圖片識別字。

下面的例子，它會讀取圖片，然後加入文字再輸出至瀏覽器:

```
<?php
```

```
$im = imagecreatefromjpeg("images/pic1.jpg");
```

```
$textcolor = imagecolorallocate($im, 0, 0, 255);
```

```
//加入文字
```

```
imagestring($im, 20, 380, 350, "Photo By Jean", $textcolor);
```

```
//輸出圖片
```

```
header("Content-type: image/png");
```

```
imagepng($im);
```

```
//釋放影像佔用的記憶體
```

```
imagedestroy($im);
```

```
?>
```

