

來玩寫程式 (一)

翁添雄
靜宜大學資工系

前言

- 寫程式很難嗎?很無趣?
- 你寫的程式是你的智慧財產
- 寫程式可創意哦
- 來! 輕鬆又有趣的玩寫程式吧!
 - 寫一個命令電腦繪畫的程式
 - 再寫動畫的程式
 - 再寫遊戲程式

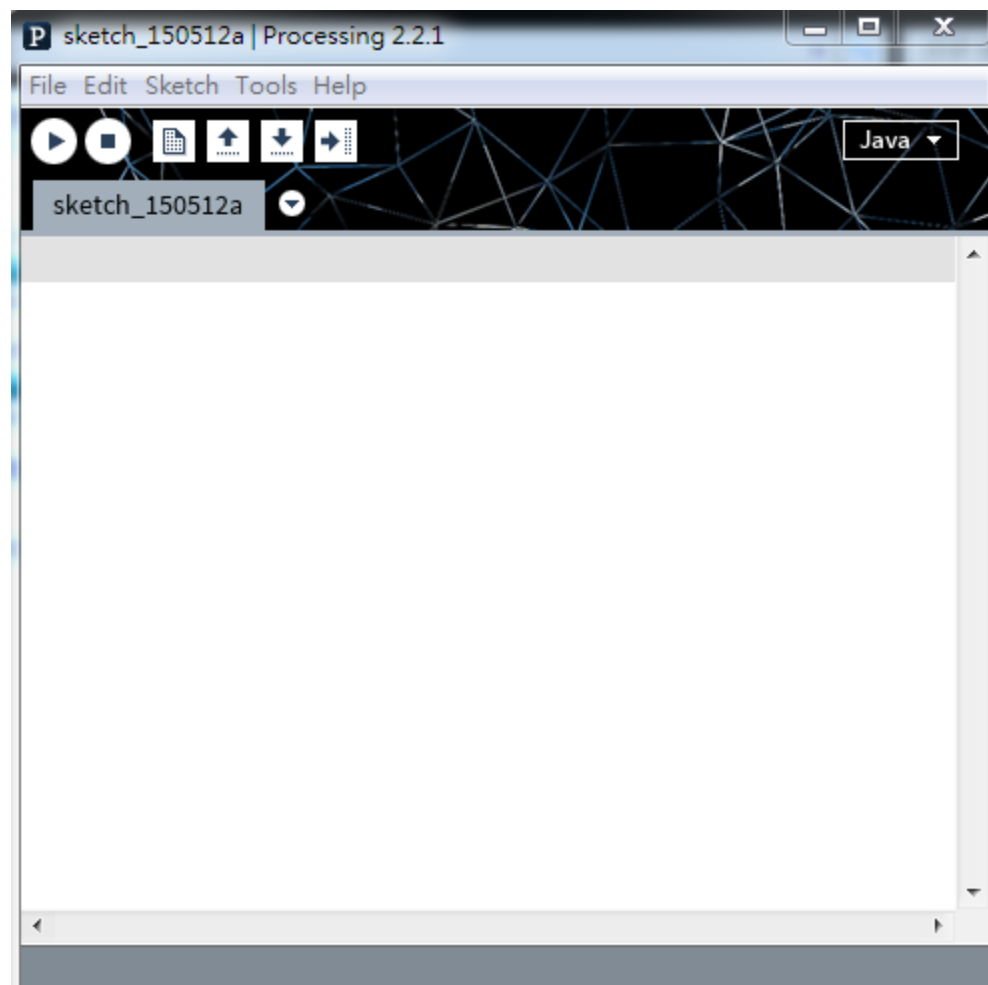
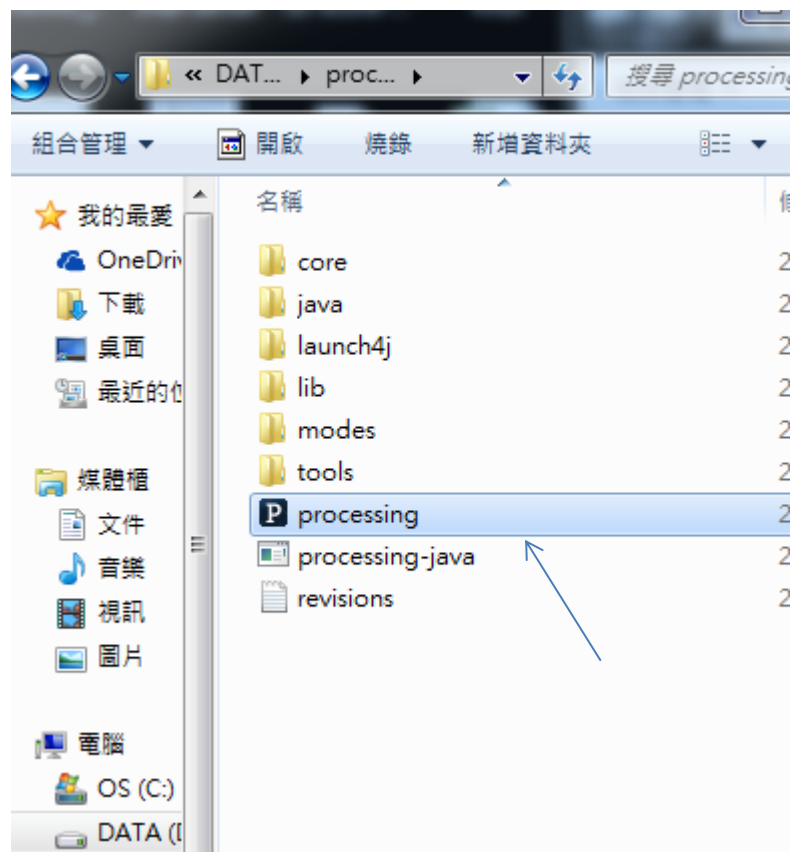
Processing程式語言

- 一種可視化程式語言
- 由 MIT 媒體實驗室(Casey Reas 和 Benjamin Fry) 所發明
- 為數字媒體與娛樂互動設計而創建
- 可讓編程者表達圖形及數字創意
- 基於Java語言，進一步簡化了語法

安裝processing

- Google search
- Keyword(關鍵字): processing programming
- <https://processing.org>
- 要注意的是你下載的是32或64位元windows版本。先查看你的windows作業系統是32還是64位元。
- 也有Linux版本。

啟動 processing



基本繪圖指令

翁添雄

靜宜大學資工系

基本繪圖指令

- 開視窗
- 視窗座標概念，設視窗背景顏色
- 設筆畫粗細及筆畫顏色
- 畫基本形狀
- 繪圖填顏色

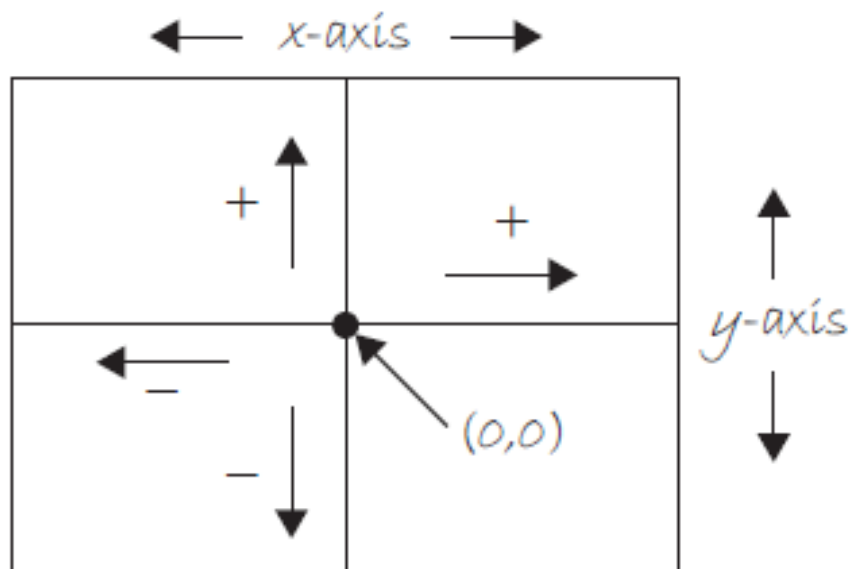
指令1: size

- 每一個指令間需要「;」分號
- 指令格式: **size(width, height);**
- 會出視窗，大小於像素點的個數
- **size(400,500);**
- 劃出寬400像素與高500像素的視窗
- 寫程式時，必須先會出視窗，在視窗內才可畫出許多的圖及文字

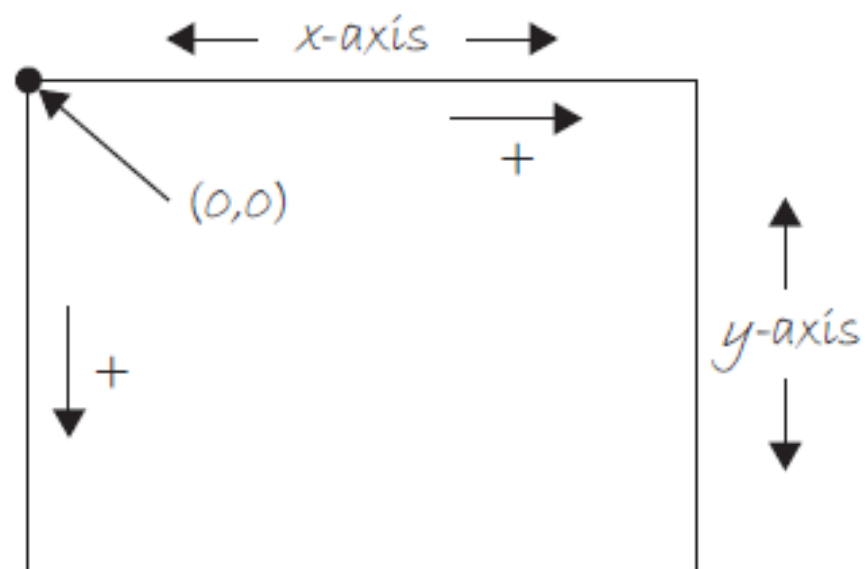
指令2: background

- 設背景顏色
- 格式: **background(color);** 或
- **background(red, green, blue);**
- color值為0至255(灰階) , 0(黑色)至255(白色)
- red值為0至255
- green值為0至255
- blue值為0至255

視窗座標系統



我們的座標系統



電腦的座標系統

控制筆的粗細及顏色

**指令3: stroke(color); 或
stroke(red,green,blue); 或
stroke(red,green,blue,alpha);**

- Red,green,和blue值為0至255
- Color: 0至255 (灰階—黑色至白色)
- Alpha為透明度，直0-255
- 繪圖前下的指令

指令4: noStroke();

- 不畫出筆
- 如畫圓圈時，不畫出筆線，但可填顏色

控制筆的粗細

- 指令5: **strokeWeight(n);**
 - n為整數
 - 繪圖前下的指令

控制筆的平滑度

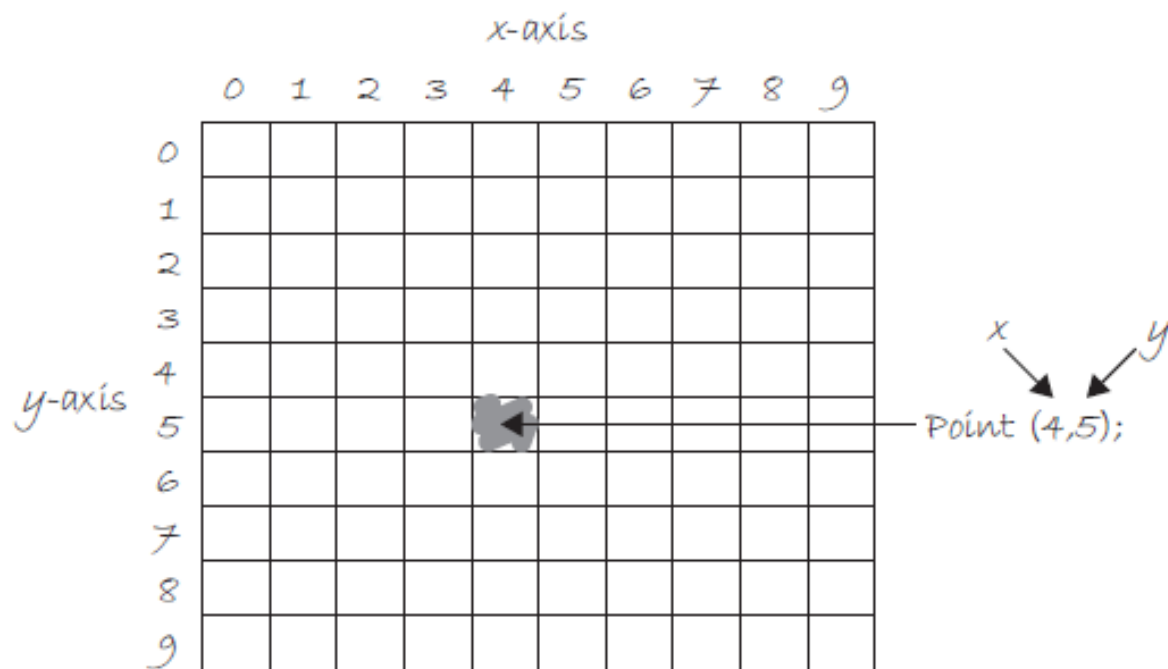
- **指令6: smooth();**
 - 設筆畫平滑，尤其是畫圓圈時為明顯
 - 無參數
 - 繪圖前下的指令
- **指令7: noSmooth();**
 - 設筆不平滑
 - 無參數
 - 繪圖前下的指令

控制填顏色

- 指令8: **fill(color);** 或 **fill(red,green, blue);**或
fill(red,green,blue,alpha);
 - 填顏色，灰階—黑色至白色
 - Color值為0至255
 - alpha為透明度，直0-255 //在說明動畫時，會應用
 - 繪圖前下的指令
- 指令9: **noFill();**
 - 不填顏色
 - 無參數
 - 繪圖前下的指令

指令10: point

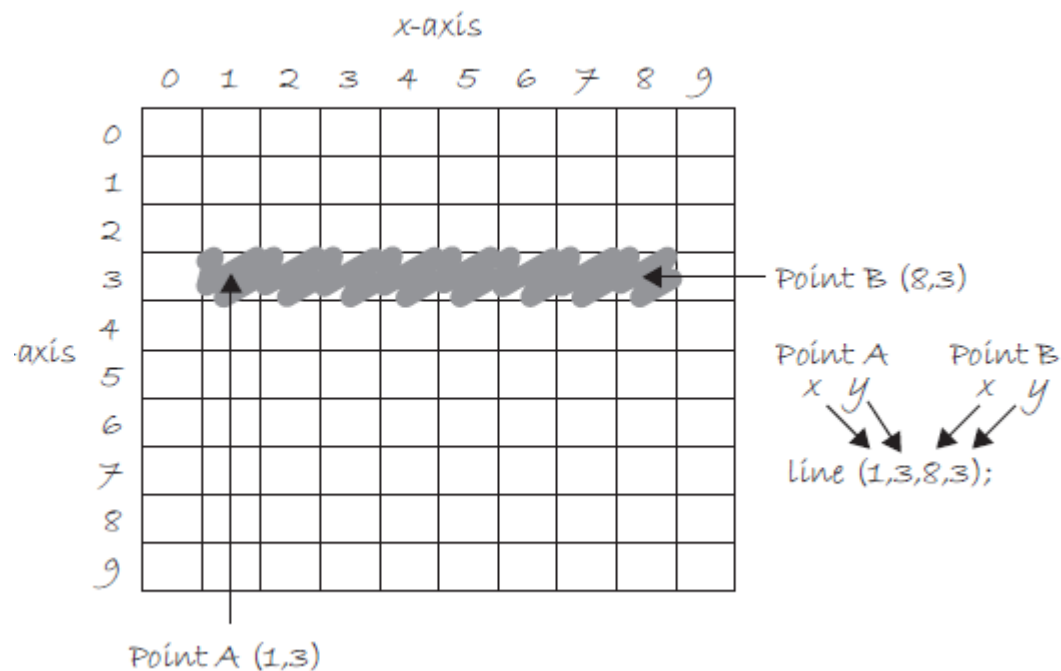
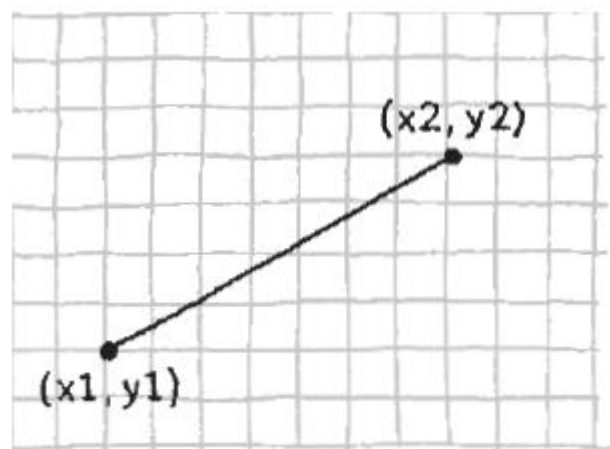
- 格式: **point(x,y);**
- 或在x,y座標畫一個點



指令11: line

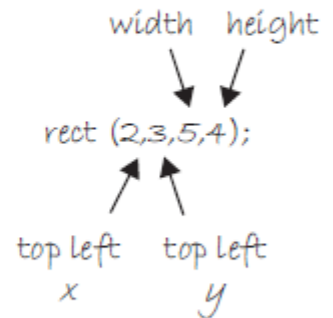
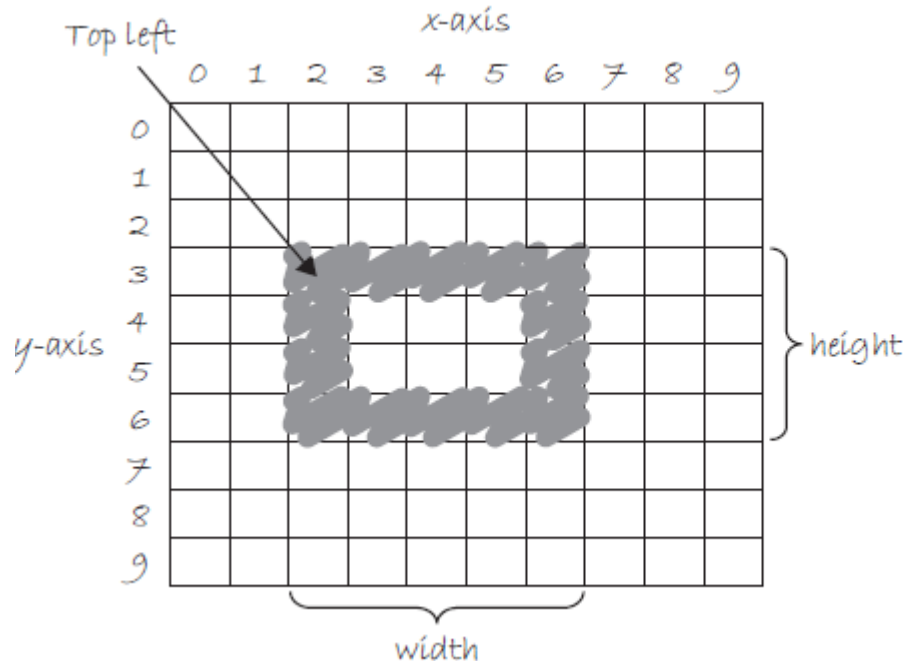
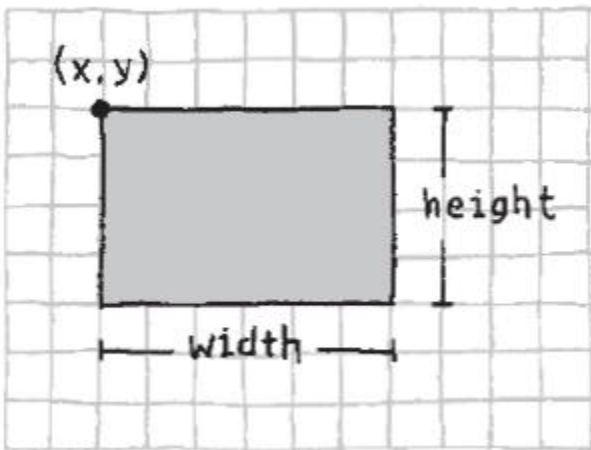
- 格式: **line(x1,y1,x2,y2);**
- 或在兩個點座標畫一條線

如line(1,3,8,3);



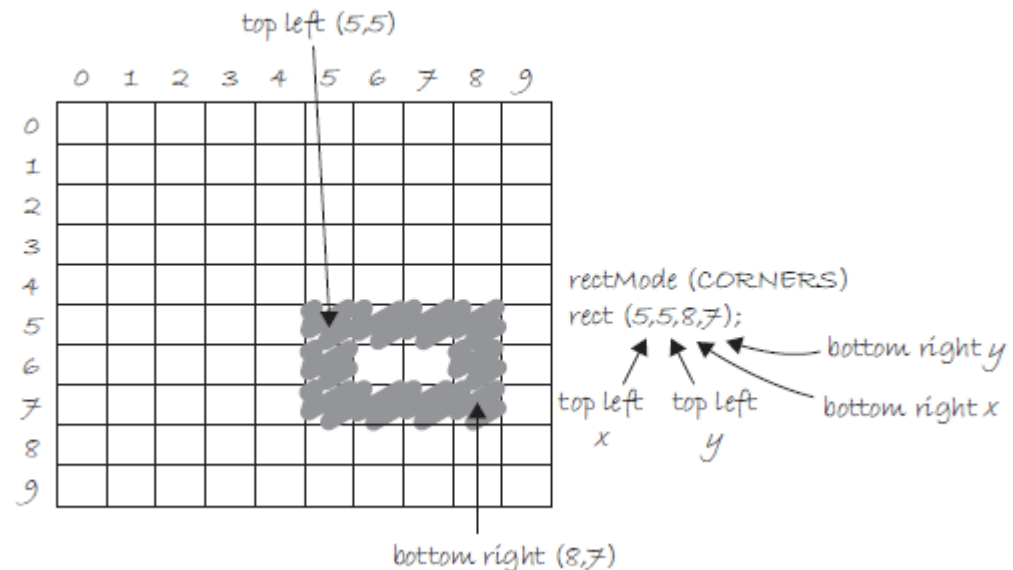
指令12: rect

- 格式: **rect(x,y,width,height);**
- 或在x,y座標按寬和高所述畫長方形
- 內定是畫在左上角(CORNER)
- **rectMode(CORNER);**
- **rect(2,3,5,4);**



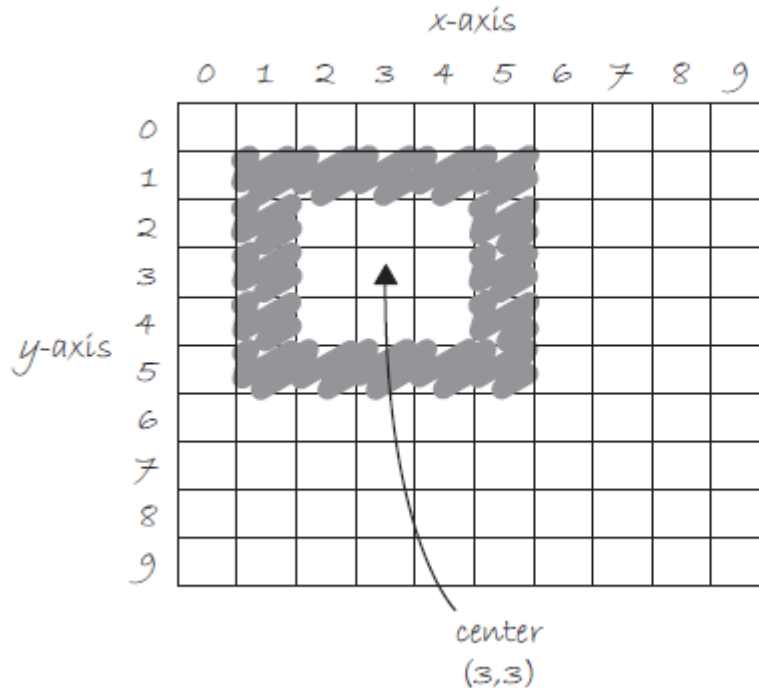
指令12: rect

- 格式: **rect(x,y,x2,y2);**
- 或在 兩個點畫長方形
- **rect(5,5,8,7);**



指令12: rect

- `rectMode(CENTER);`
- `rect(3,3,5,5);`
- 內定是畫在中心點(CENTER)

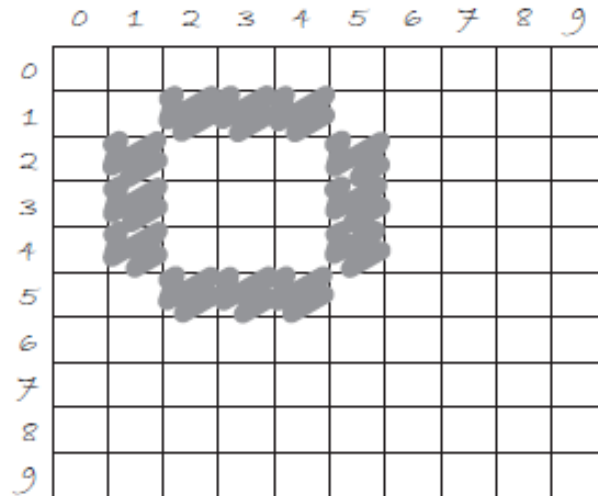
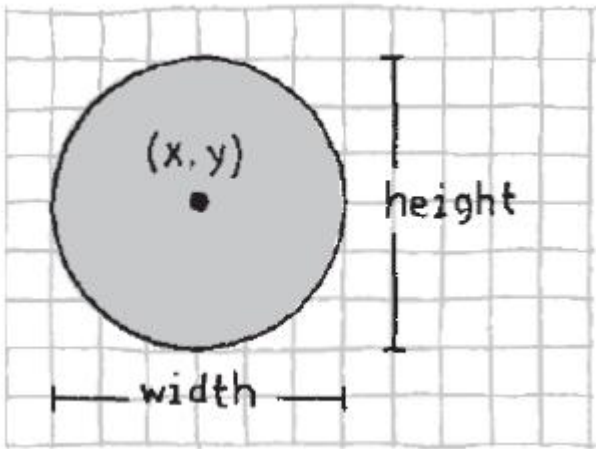


`rectMode(CENTER);`
`rect(3,3,5,5);`

center x center y height width

指令13: ellipse

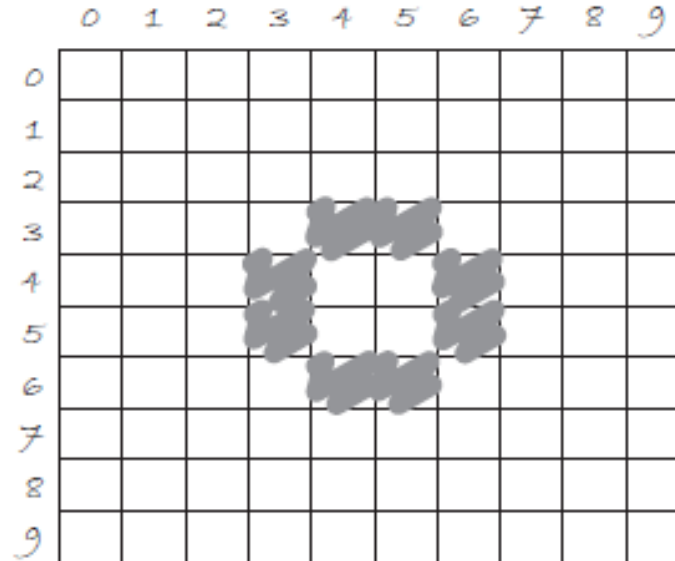
- 格式: **ellipse(x,y,width,height);**
- **ellipseMode(CENTER);**
- **ellipse(3,3,5,5);**
- 內定是畫在中心點(CENTER)



```
ellipseMode(CENTER);  
ellipse(3,3,5,5);
```

指令13: ellipse

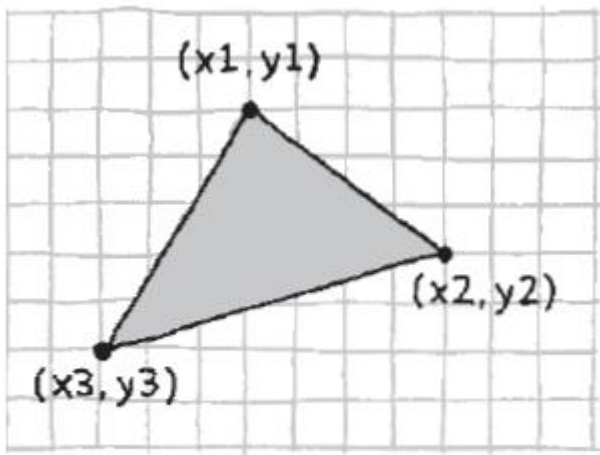
- **ellipseMode(CORNER);**
- **ellipse(3,3,4,4);**
- 可改畫在左上角(CORNER)



`ellipseMode(CORNER);`
`ellipse(3,3,4,4);`

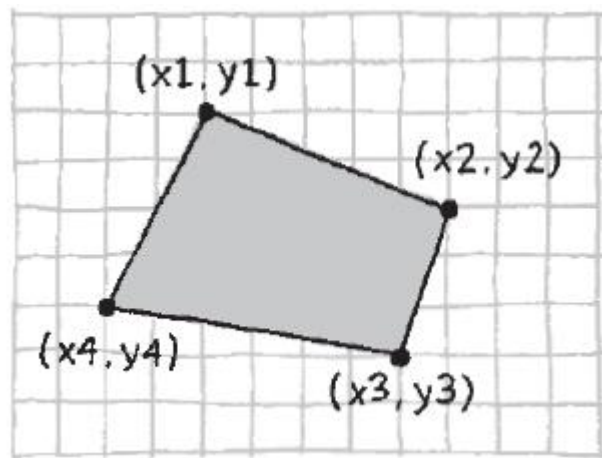
指令14: triangle

- 格式: **triangle(x1,y1,x2,y2,x3,y3);**



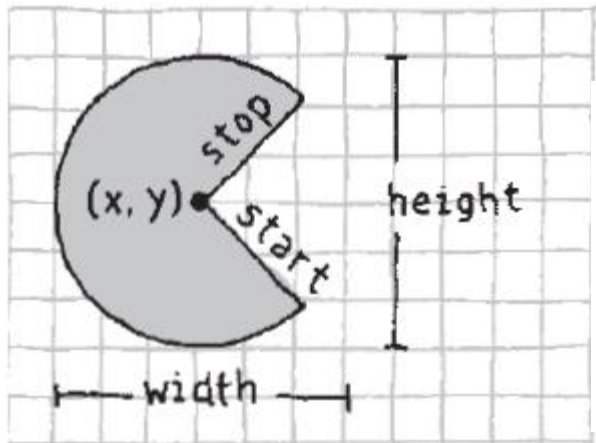
指令15: quad

- 格式: **quad(x1,y1,x2,y2,x3,y3,x4,y4);**



指令16: arc

- 格式: **arc(x,y,width,height,start,stop);**
- start和stop為角度0至360



```
arc(90, 60, 80, 80, 0, HALF_PI);
```

```
arc(190, 60, 80, 80, 0, PI+HALF_PI);
```

```
arc(290, 60, 80, 80, PI, TWO_PI+HALF_PI);
```

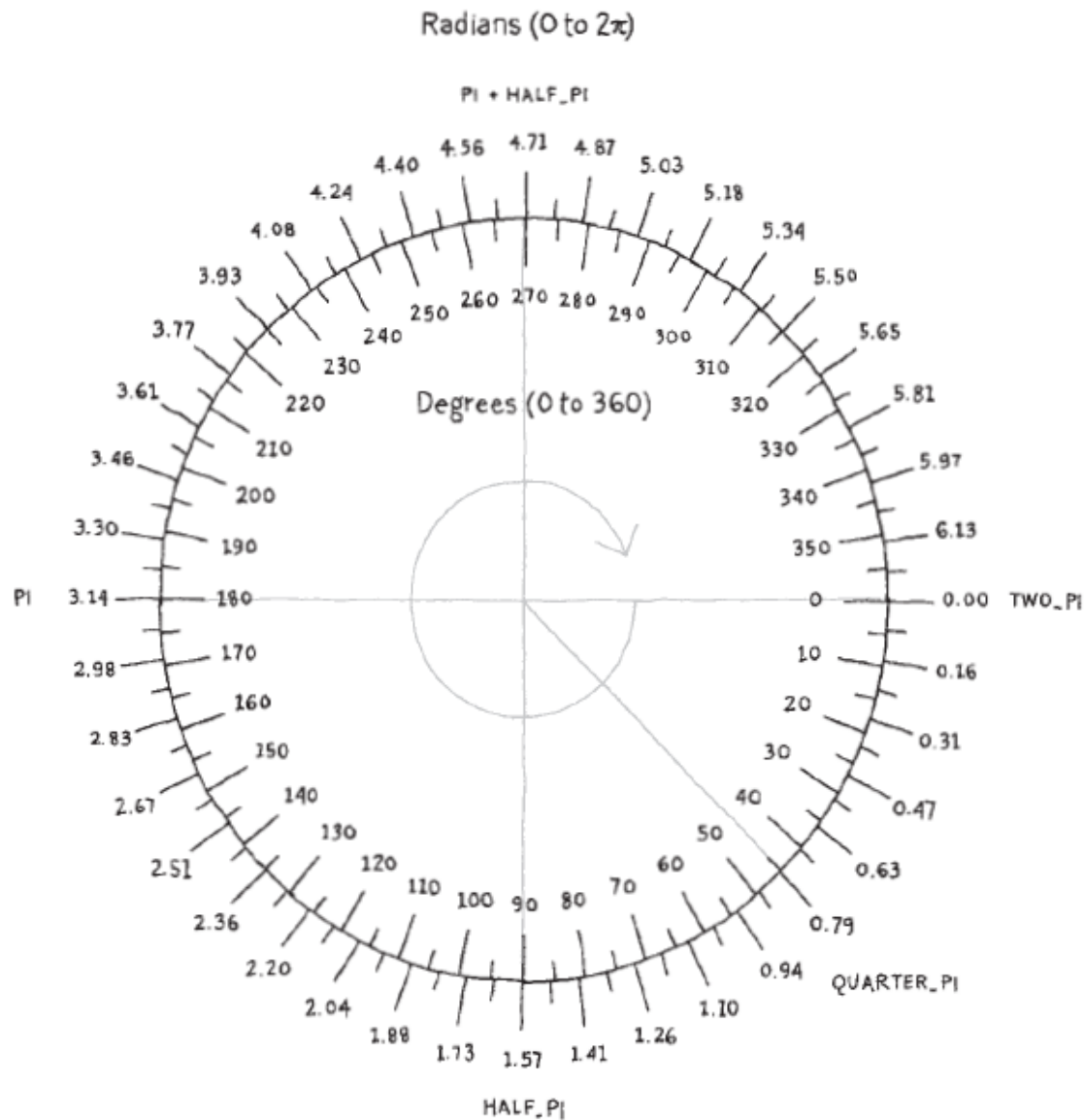
```
arc(390, 60, 80, 80, QUARTER_PI, PI+QUARTER_PI);
```


指令16: arc

```
arc(90, 60, 80, 80, 0, radians(90));  
arc(190, 60, 80, 80, 0, radians(270));  
arc(290, 60, 80, 80, radians(180), radians(450));  
arc(390, 60, 80, 80, radians(45), radians(225));
```



Radians(弧度)



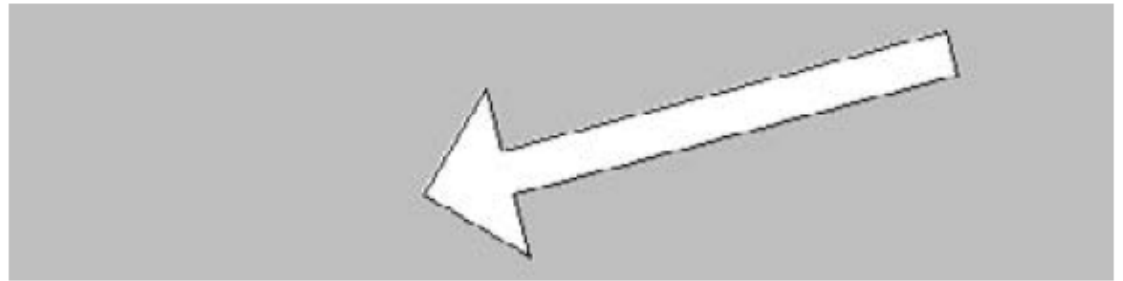
定形狀

- 指令**17**: **beginShape()**; 定義形狀開始
vertex(x,y);

.....

endShape(); 定義形狀結束

內於多點連接



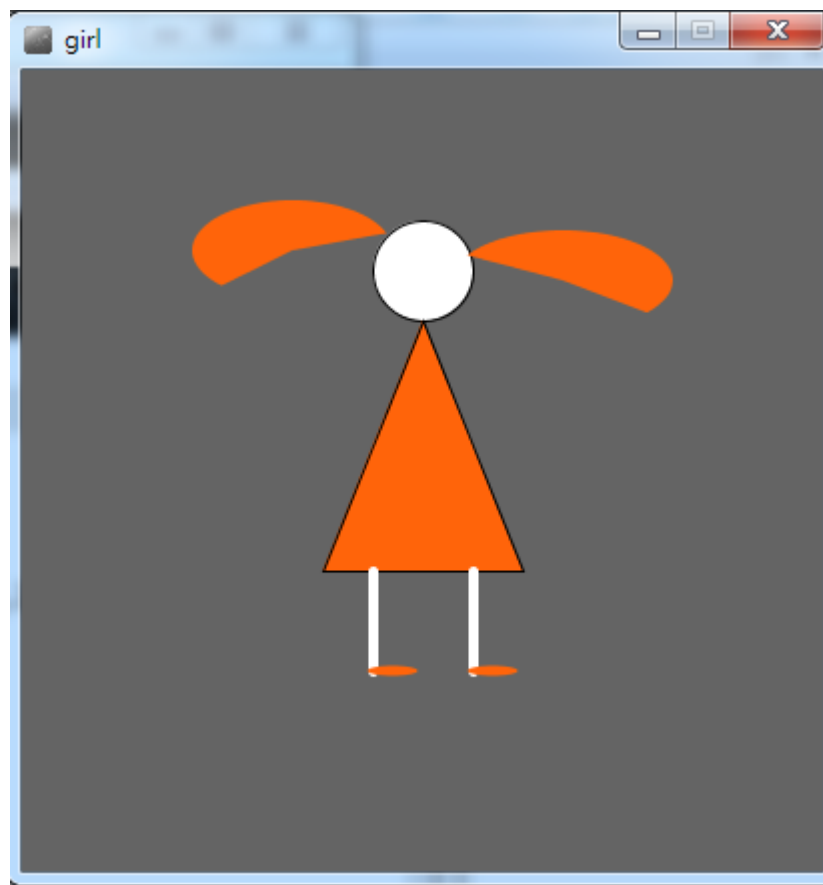
```
size(480, 120);  
beginShape();  
vertex(180, 82);  
vertex(207, 36);  
vertex(214, 63);  
vertex(407, 11);  
vertex(412, 30);  
vertex(219, 82);  
vertex(226, 109);  
endShape(CLOSE);
```

程式註解

- `//` 程式註解
- `//` 後面圍註解，不會當指令執行
- 常用來說明程式的算法

程式範例1

- 用指令畫出girl如下圖:

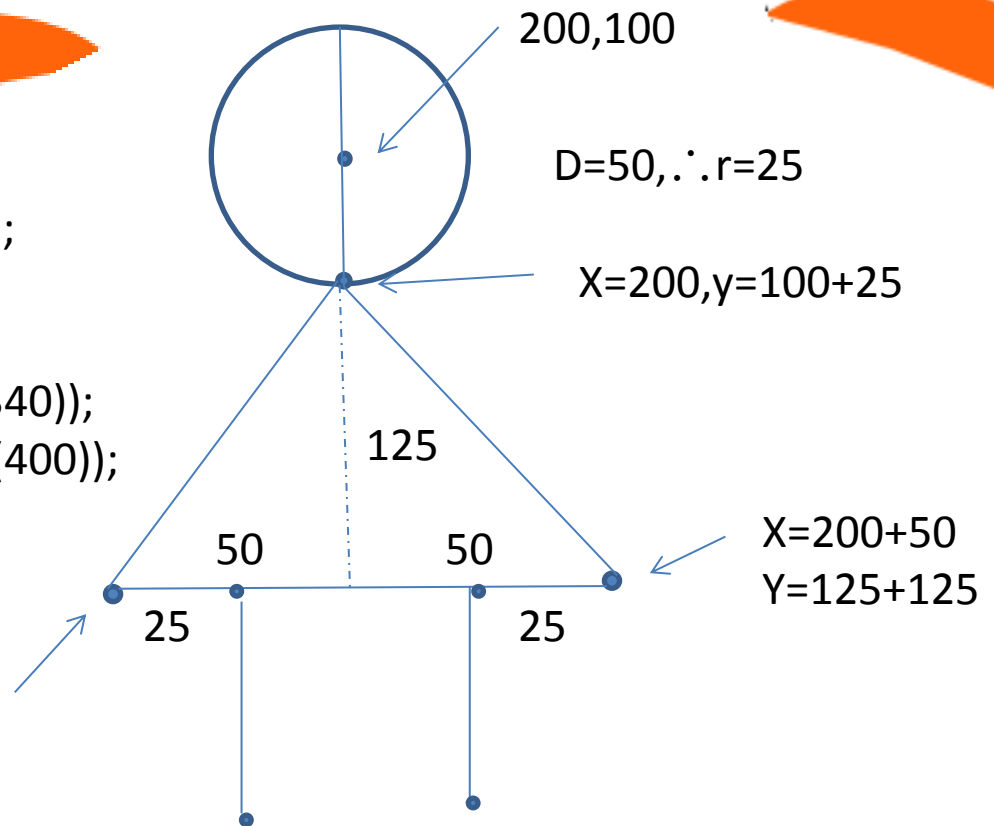


程式範例1

要預先計畫:

```
ellipse(200,100,50,50);  
triangle(200,125,200-50,250,200+50,250);  
line(175,250,175,300);  
line(225,250,225,300);  
arc(135,90,100,50,radians(135), radians(340));  
arc(270,105,110,50,radians(210), radians(400));  
arc(x,y,width,height, start, stop);
```

$X=200-50$
 $Y=125+125$

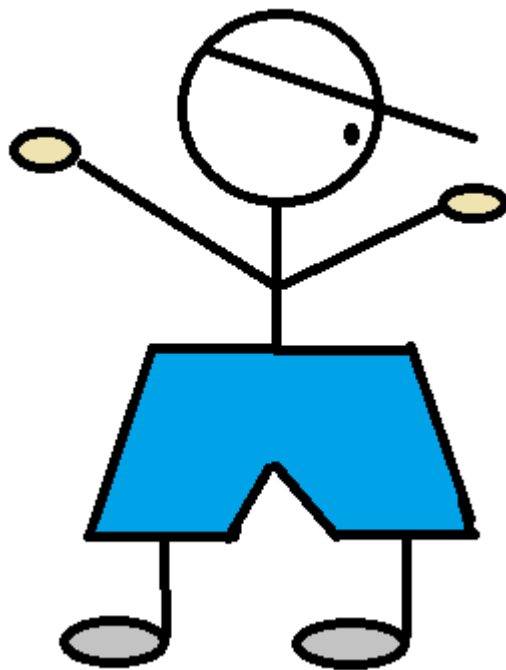


程式範例1

```
size(400,400);  
background(100);  
fill(255);  
ellipse(200,100,50,50);  
fill(255,100,10);  
triangle(200,125,200-50,250,200+50,250);  
stroke(255);  
strokeWeight(5);  
  
line(175,250,175,300);  
line(225,250,225,300);  
noStroke();  
ellipse(185,300,25,5);  
ellipse(235,300,25,5);  
arc(135,90,100,50,radians(135), radians(340));  
arc(270,105,110,50,radians(210), radians(400));
```

作業1

- 劃出boy



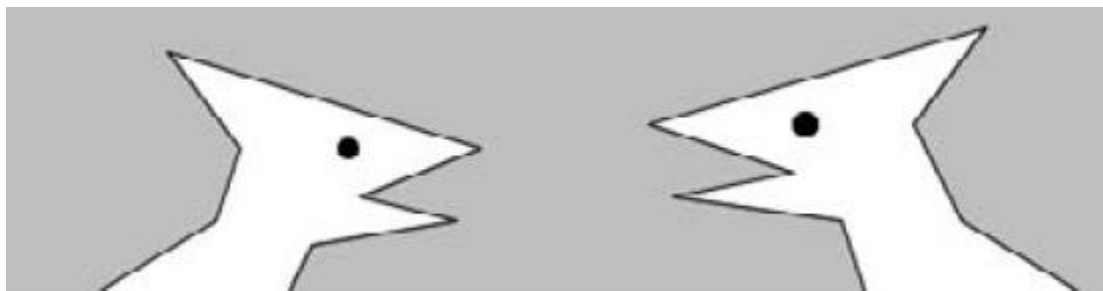
作業2

- 請寫一processing程式繪如下的圖案:



作業3

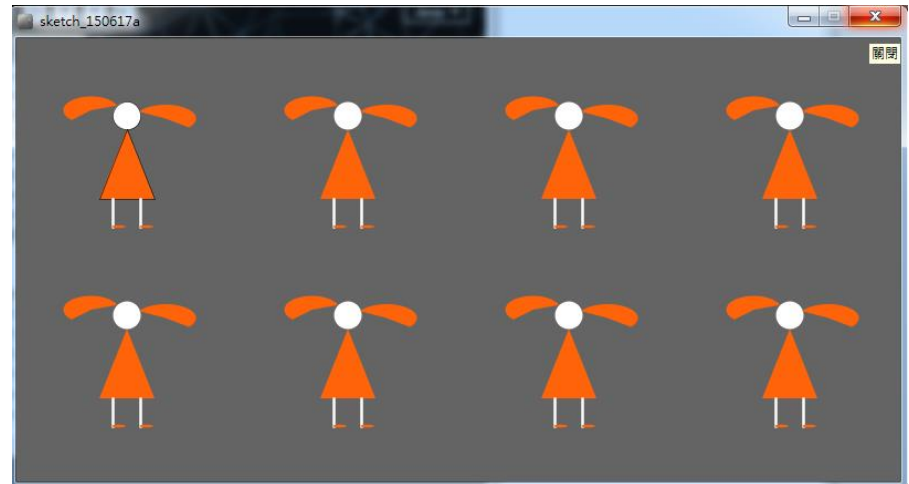
- 請寫一processing程式繪如下的圖案:



用副程式畫多個girl

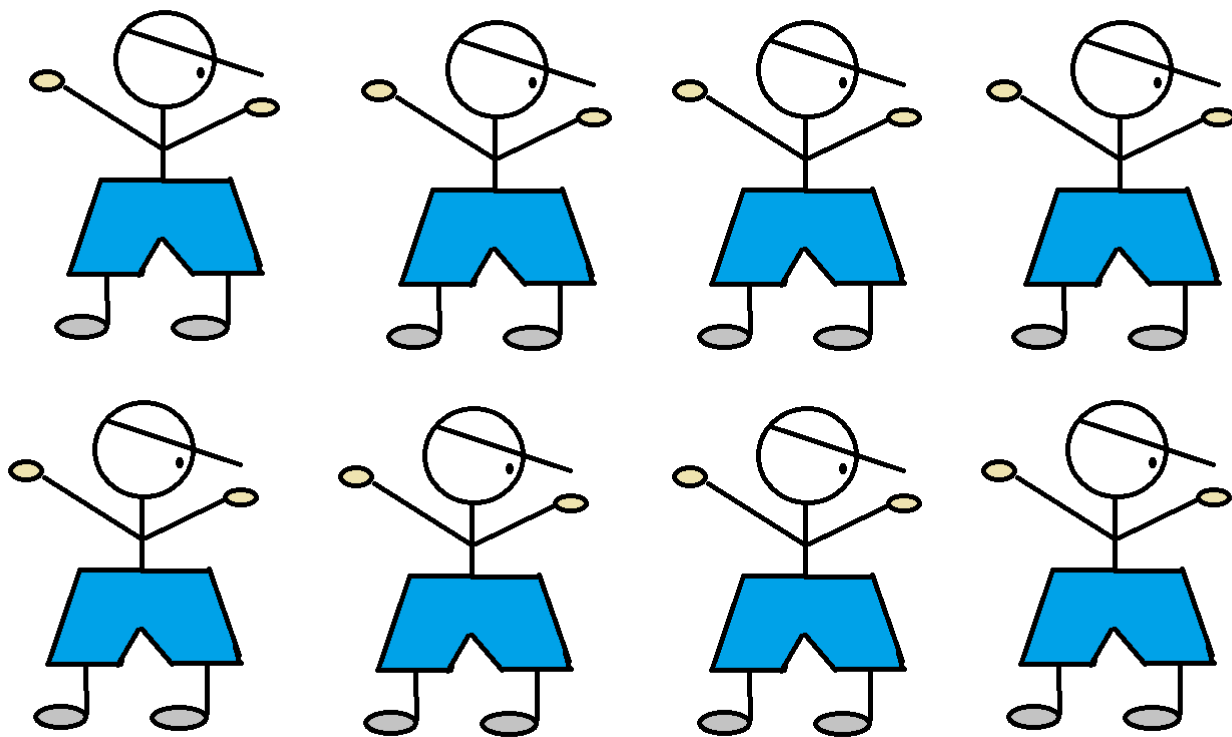
```
void setup()
{
  size(800,400);
  background(100);
  scale(0.5);    // 縮小0.5
  girl(00,40);   // 畫縮小0.5 的girl
  girl(400,40);
  girl(800,40);
  girl(1200,40);
  girl(00,400);
  girl(400,400);
  girl(800,400);
  girl(1200,400);
}

void girl(int x, int y)
{
  fill(255);
  ellipse(x+200,y+100,50,50);
  fill(255,100,10);
  triangle(x+200,y+125,x+200-50,y+250,x+200+50,y+250);
  stroke(255);
  strokeWeight(5);
  line(x+175,y+250,x+175,y+300);
  line(x+225,y+250,x+225,y+300);
  noStroke();
  ellipse(x+185,y+300,25,5);
  ellipse(x+235,y+300,25,5);
  arc(x+135,y+90,100,50,radians(135), radians(340));
  arc(x+270,y+105,110,50,radians(210), radians(400));
}
```



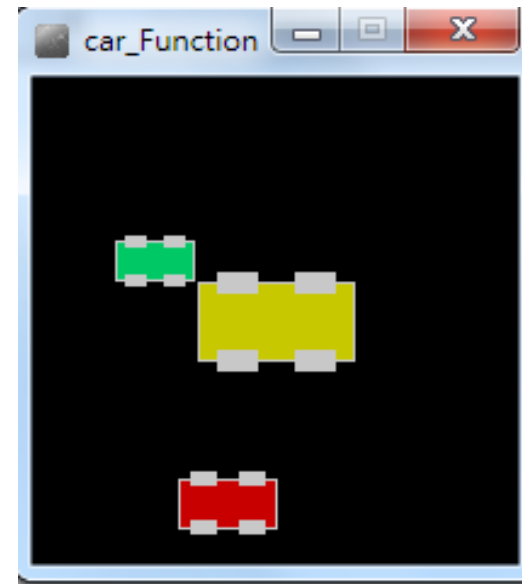
作業4

- 使用副程式，畫出下列圖：



畫車子

```
size(200,200);
background(255);
int x = 100 , y = 100;
int thesize = 64; // size
int offset = thesize/4; // position of wheels relative to car
// draw main car body (i.e. a rect)
rectMode(CENTER);
stroke(0);
fill(175);
x = 50;
y = 50;
thesize = 24;
offset = thesize/4; // position of wheels relative to car
// draw main car body (i.e. a rect)
rectMode(CENTER);
stroke(0);
fill(175);
rect(x,y,thesize,thesize/2);
// draw four wheels relative to center
fill(0);
rect(x - offset,y - offset,offset,offset/2);
rect(x + offset,y - offset,offset,offset/2);
rect(x - offset,y + offset,offset,offset/2);
rect(x + offset,y+offset,offset,offset/2);
```

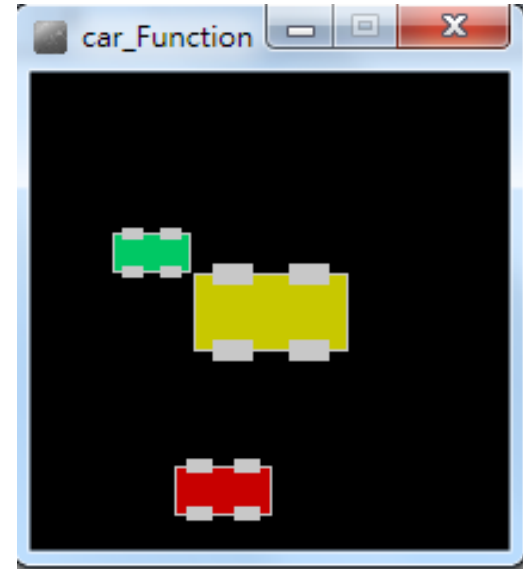


DrawCar 副程式

```
void drawCar(int x, int y, int thesize, color c)
{  int offset = thesize/4;
   // Draw main car body
   rectMode(CENTER);
   stroke(200);
   fill(c);
   rect(x,y,thesize,thesize/2);
   // Draw four wheels relative to center
   fill(200);
   rect(x - offset,y - offset,offset,offset/2);
   rect(x + offset,y - offset,offset,offset/2);
   rect(x - offset,y + offset,offset,offset/2);
   rect(x + offset,y + offset,offset,offset/2);
}

void setup()
{  size(200,200);
}

void draw()
{  background(0);
   drawCar(100,100,64,color(200,200,0));
   drawCar(50,75,32,color(0,200,100));
   drawCar(80,175,40,color(200,0,0));
}
```



Function example

指令

1. size();
2. background();
3. stroke();
4. strokeWeight();
5. smooth();
6. noSmooth();
7. fill();
8. noFill();
9. point();
10. line();
11. rectMode();
12. rect();
13. ellipseMode()
14. ellipse();
15. triangle();
16. quad();
17. arc();
18. beginShape() ; vertex(); endShape();
19. radians();
20. //
21. Subprogram(副程式)

參考

- 很棒的參考網站
 - processing.org
 - funprogramming.org
- 很棒的參考書
 - Casey Reas & Ben Fry, Getting Start with Processing. O'reilly.
 - Daniel Shiffman, Learning Processing. Morgan Kaufmann.