

# D3.JS

## 互動式資料視覺化

**Lecturer: LinJer 林哲**

evin92@gmail.com







Data-Driven Documents

# D3.js - 資料驅動的**文件**

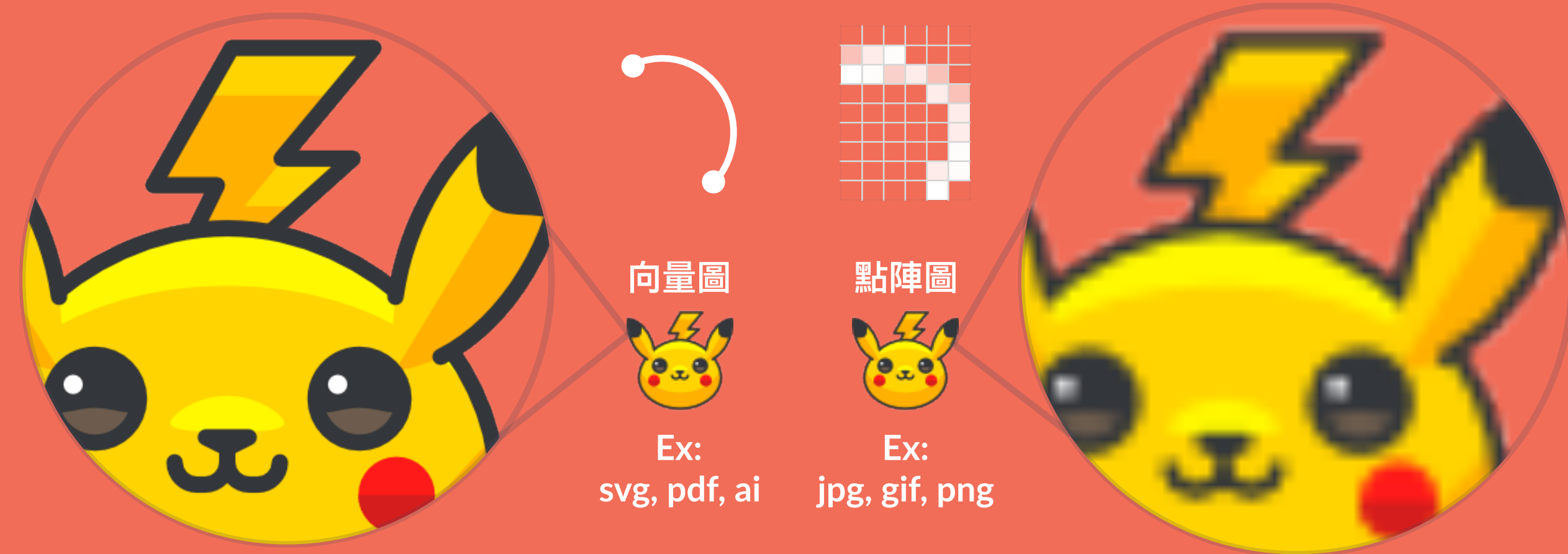
[白話]

用JS函式庫把資料與**文件中的視覺元素**  
綁定在一起

## 產生視覺元素 - SVG



**SVG:** 可縮放向量圖形  
**Scalable Vector Graphics**



SVG可顯示:

1. 向量圖形: 矩形、圓、橢圓、多邊形、直線、任意曲線
2. 內嵌圖像，包括PNG、JPEG、SVG等
3. 文字物件

## 產生視覺元素 - SVG



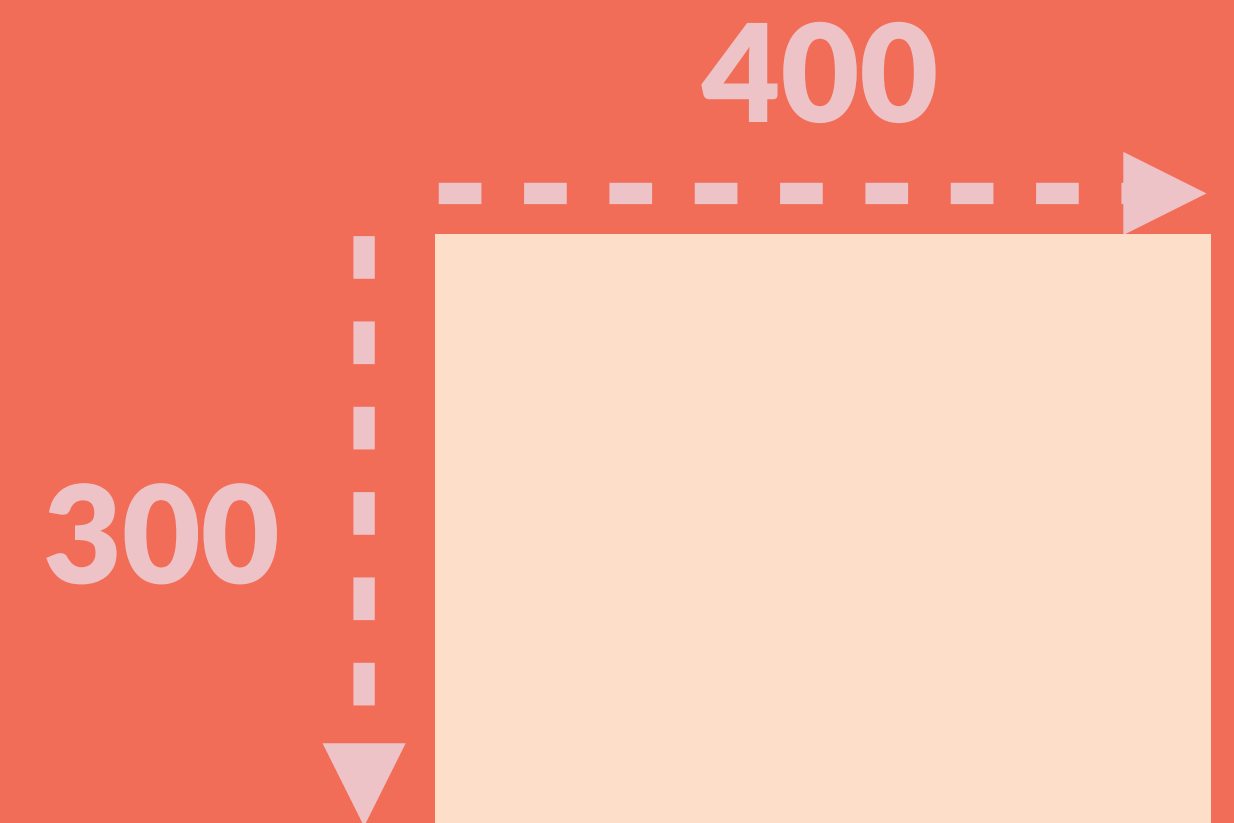
**SVG:** 可縮放向量圖形  
**Scalable Vector Graphics**

## SVG元素

標籤                      屬性

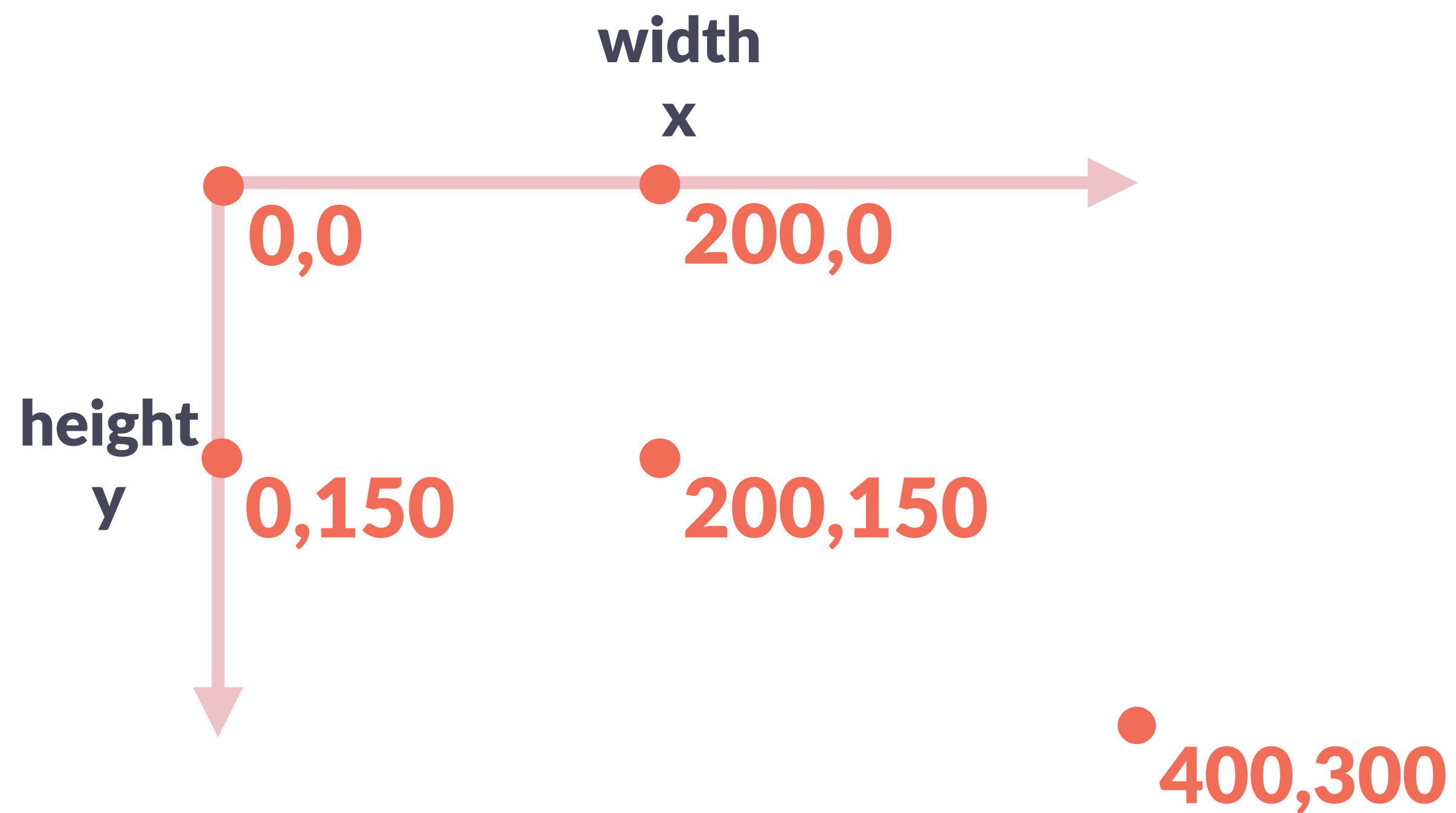
—|—

`<svg width="400" height="300">`  
`</svg>`





# SVG 的座標系統

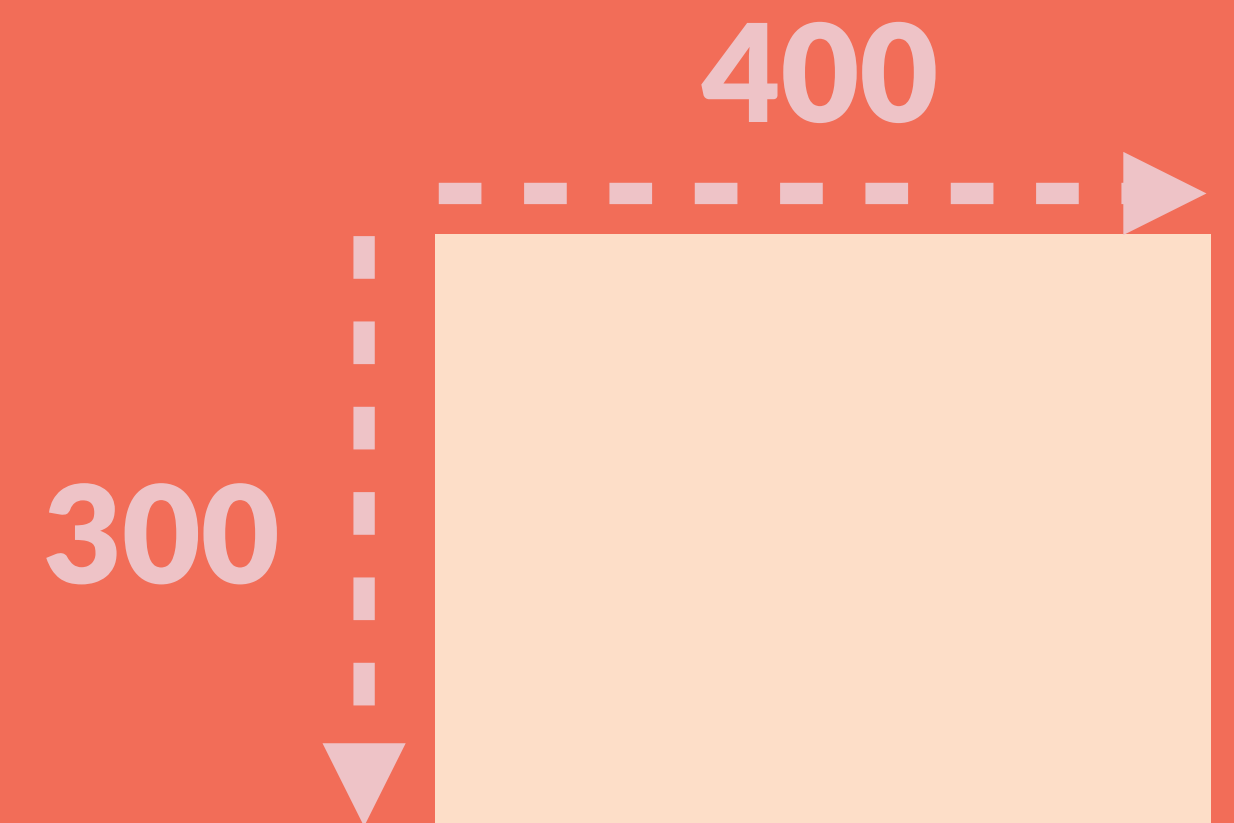


**<svg>**

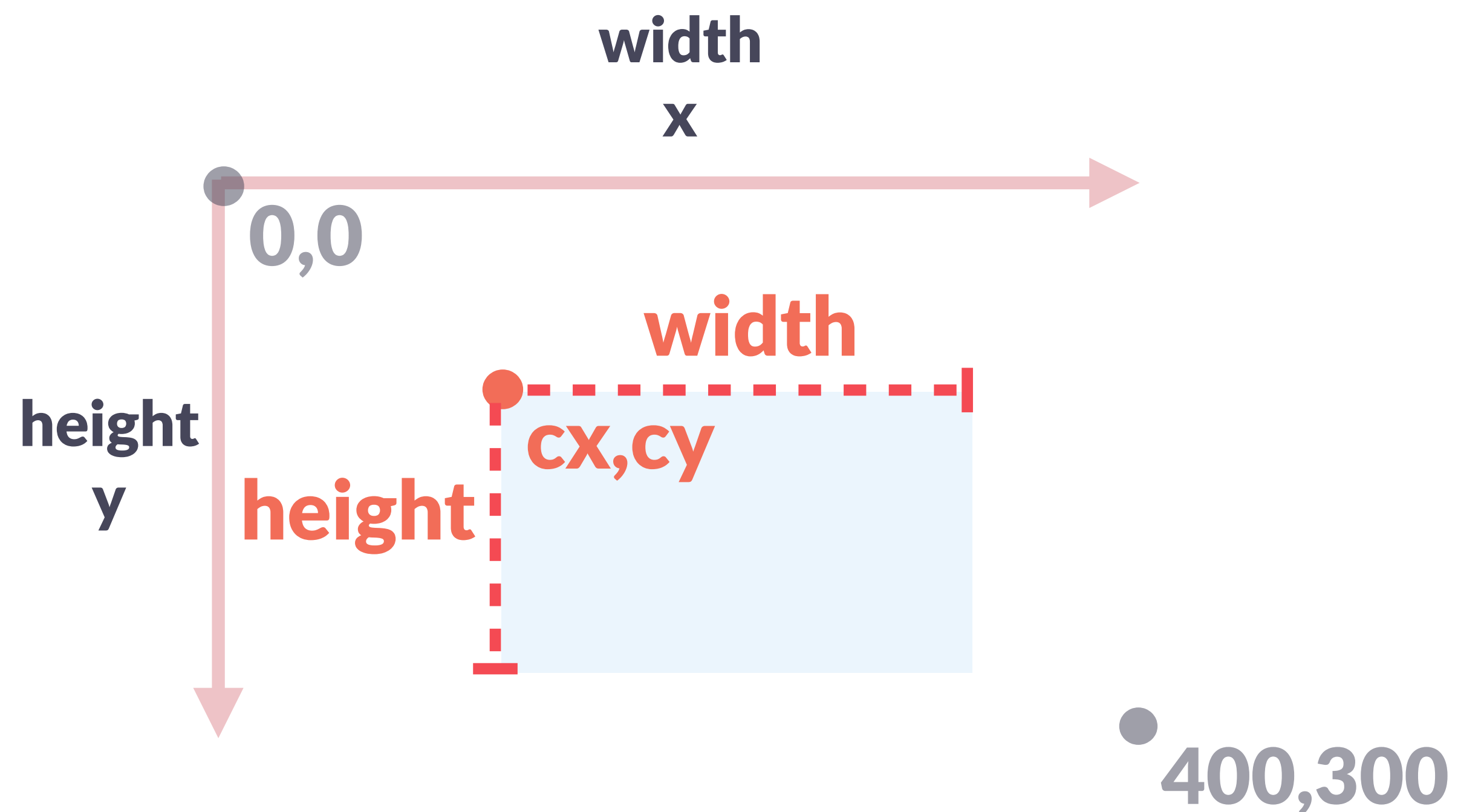
標籤

屬性

**<svg** width="400" height="300"  
**</svg>**



# 用SVG 畫矩形(RECT)

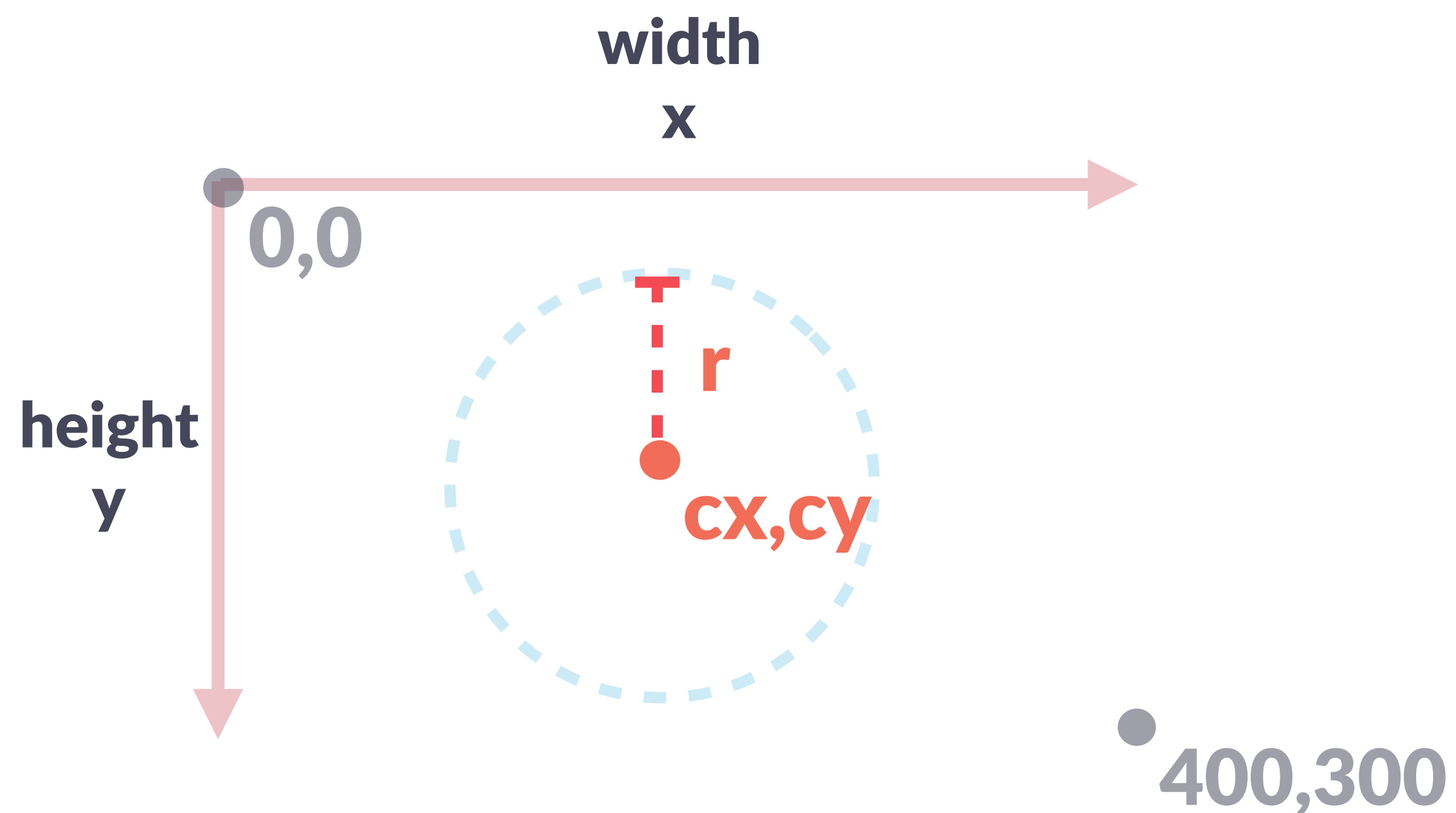


**<rect>**

```
<svg width="400" height="300">  
  <rect x="100" y="100"  
    width="150" height="50">  
  </rect>  
</svg>
```

x: (左上)起始點 x  
y: (左上)起始點 y  
width: 寬度  
height: 高度

# 用SVG 畫圓形

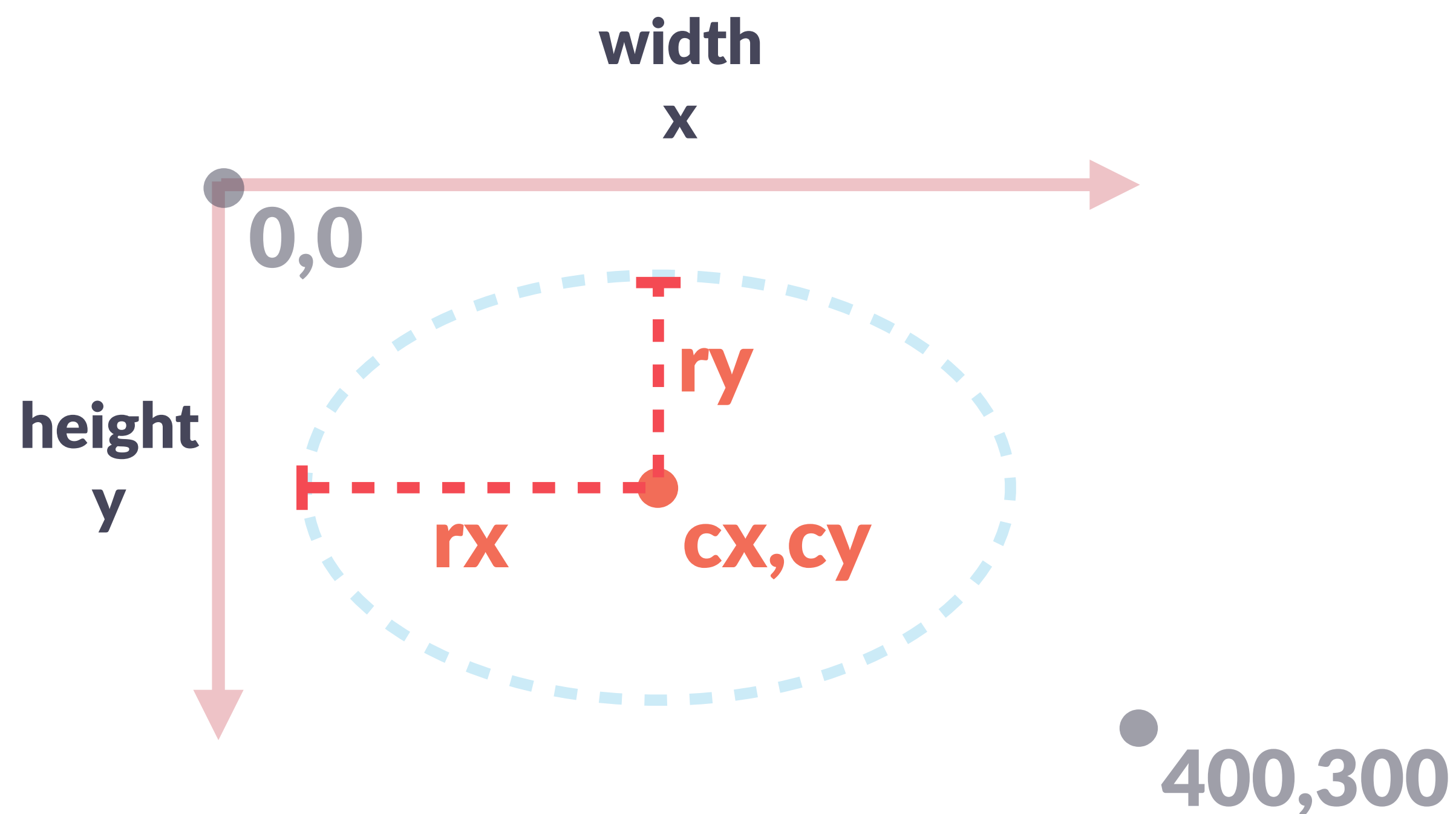


**<circle>**

```
<svg width="400" height="300">  
  <circle cx="100" cy="100"  
    r="50"></circle>  
</svg>
```

**cx: Center-X 中心點 x**  
**cy: Center-Y 中心點 y**  
**r: Radius 半徑長**

# 用SVG 畫橢圓形



## <ellipse>

```
<svg width="400" height="300">  
  <ellipse cx="100" cy="100"  
    rx="150" ry="50"></ellipse>  
</svg>
```

cx: Center-X 中心點 x  
cy: Center-Y 中心點 y  
rx: Radius-X 半徑x長  
ry: Radius-Y 半徑y長



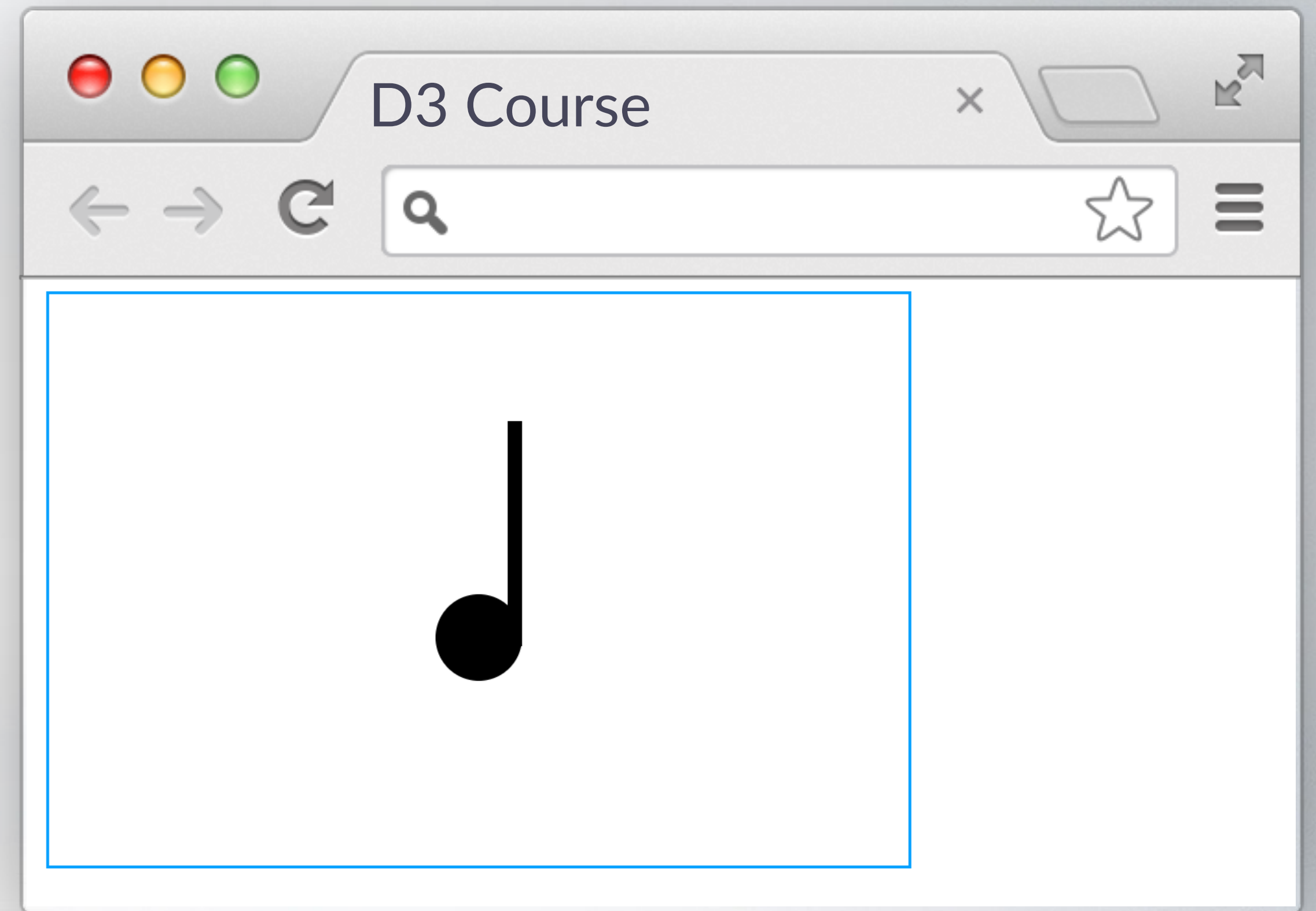
# 動手時間

-靜態SVG使用法-

## 使用SVG語法畫出來

[溫馨提示]

- 1.svg畫布寬高: 300 x 200
- 2.圓形-中心: 150, 120 半徑: 15
- 3.矩形-起始:160, 45 寬高: 5, 78



## D3語法 - 屬性器

Select

```
<body>  
  <svg width="400" height="300"></svg>  
</body>
```

Append

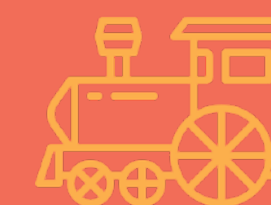
```
<body>  
  <svg width="400" height="300">  
    <circle></circle>  
  </svg>  
</body>
```

Attr

```
<body>  
  <svg width="400" height="300">  
    <circle cx="100" cy="100" r="50"></circle>  
  </svg>  
</body>
```

## 在標籤中增加屬性與值 .attr(屬性, 值)

```
d3.select("svg")  
  .append("circle")  
  .attr("cx", 100)  
  .attr("cy", 100)  
  .attr("r", 50);
```



Select

Append

Attr





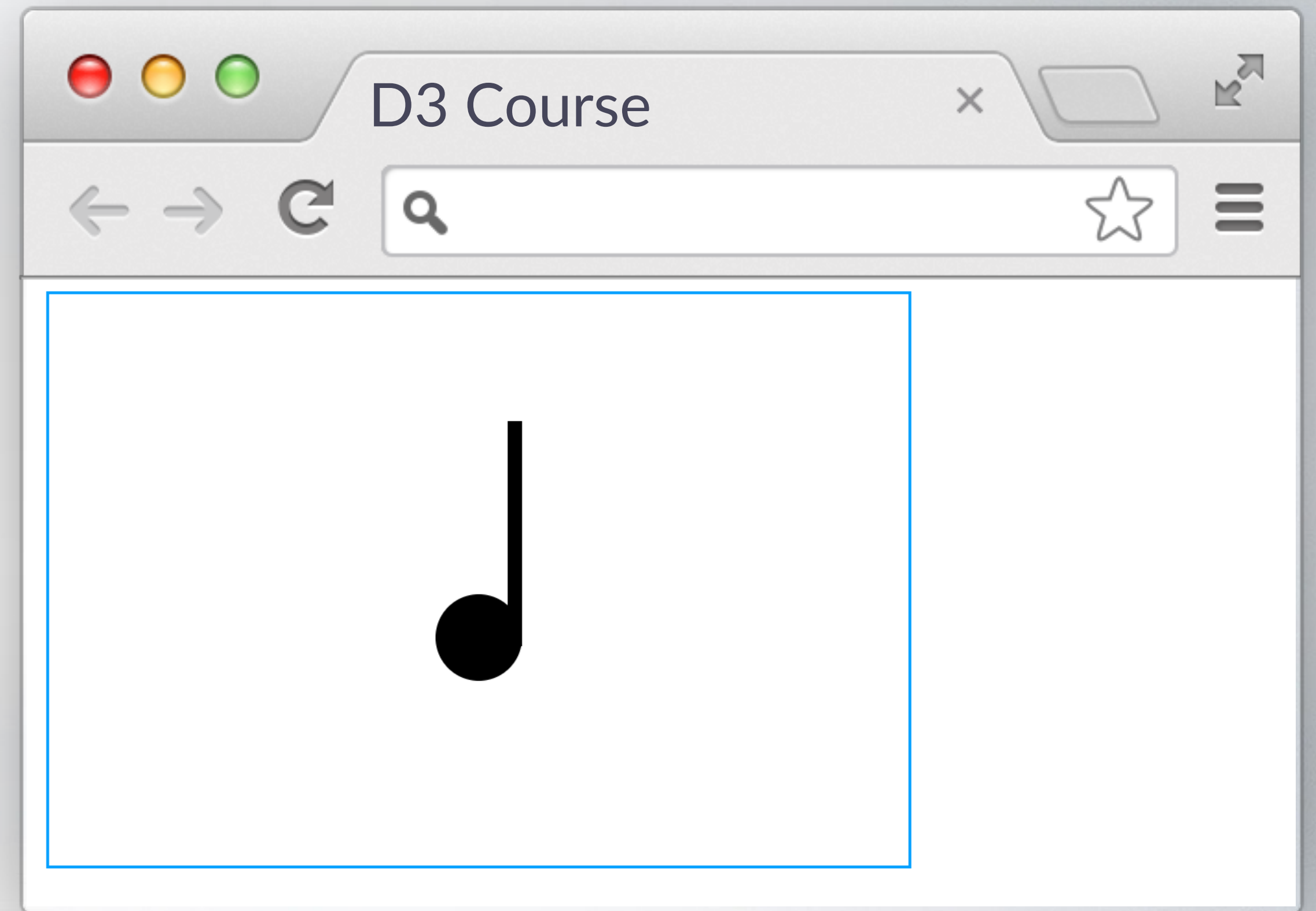
# 動手時間

-動態載入SVG使用法-

## 使用D3語法畫出來

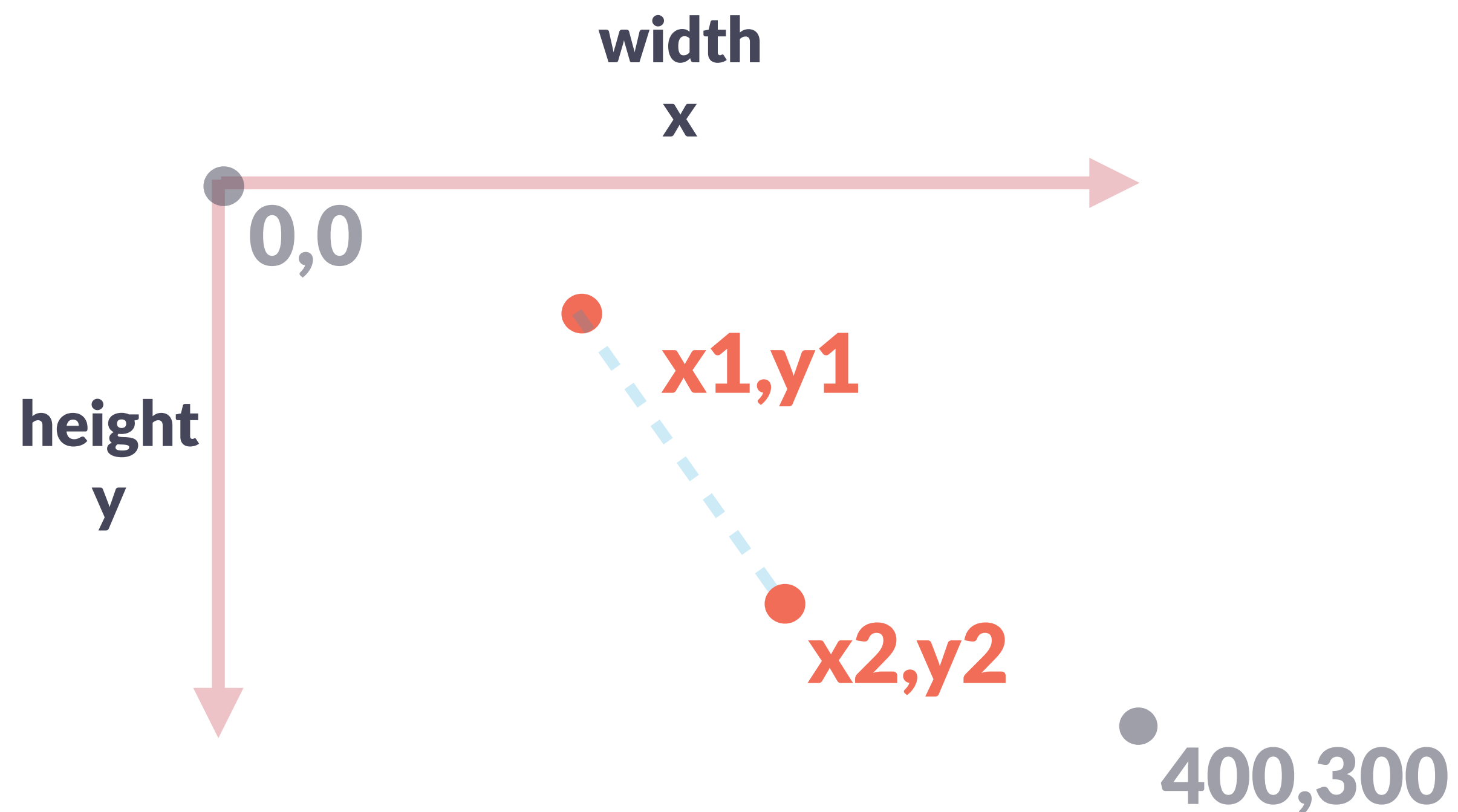
[溫馨提示]

- 1.引用2次d3(一次畫圓形，一次畫矩形)
- 2.使用屬性器 .attr()





# 用SVG 畫直線



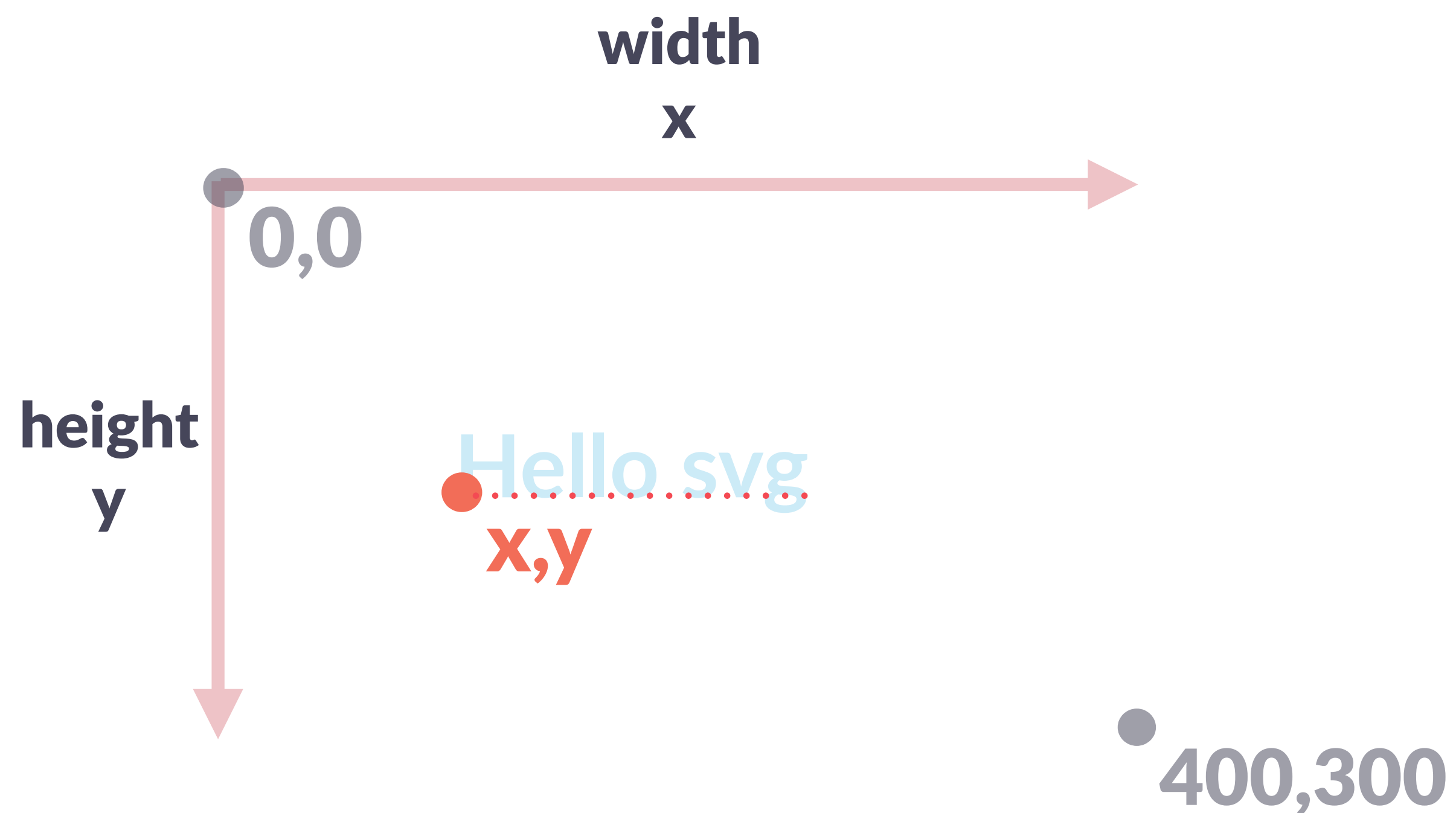
**<line>**

```
<svg width="400" height="300">  
  <line x1="50" y1="30" x2="150"  
        y2="200" stroke="black" stroke-  
        width="2"></line>  
</svg>
```

必填

$x_1, 2$ : x軸分別之兩端點  
 $y_1, 2$ : y軸分別之兩端點

# 用SVG 畫文字



**<text>**

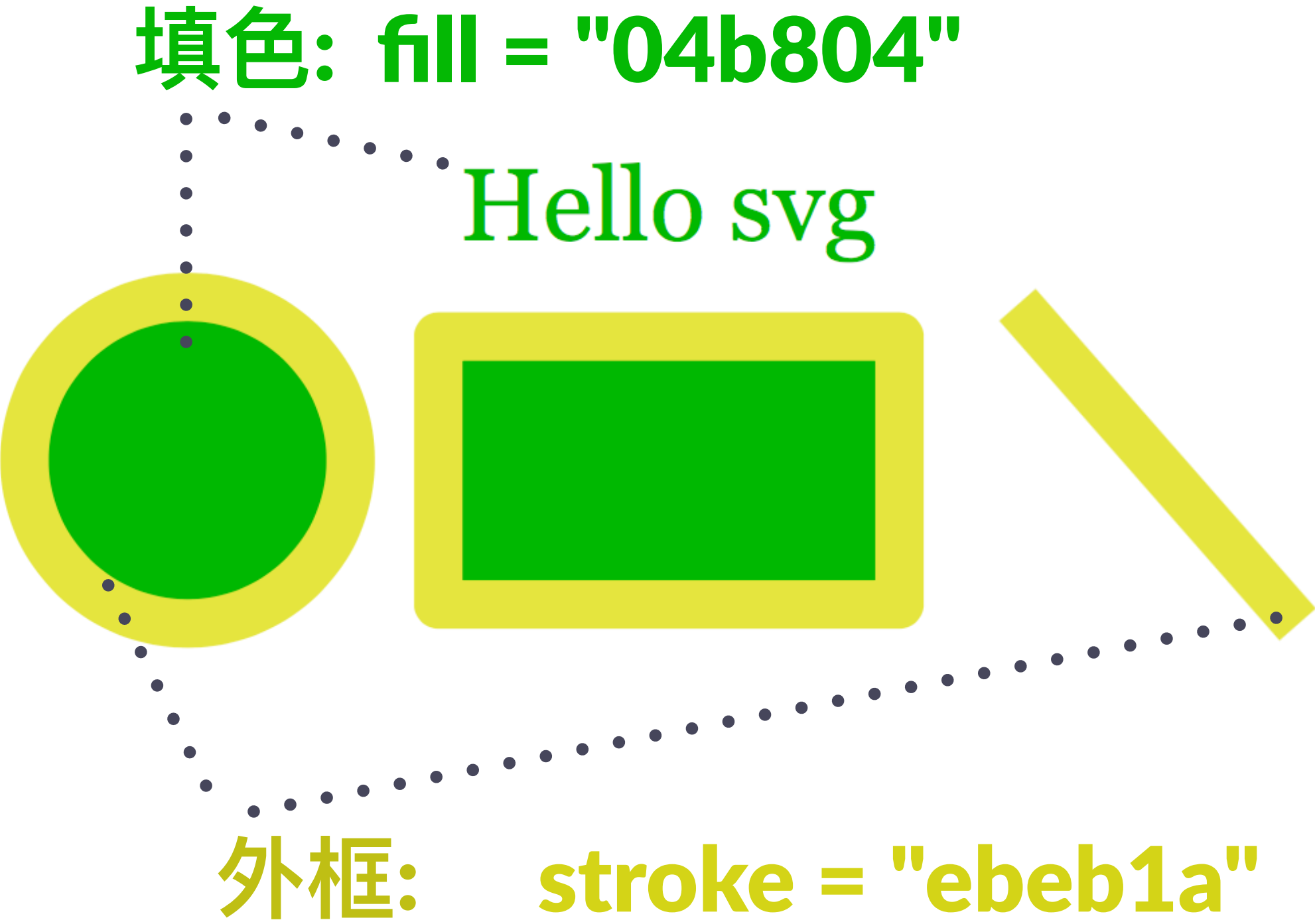
```
<svg width="400" height="300">  
  <text x="100" y="200" font-size="40" font-family="arial">  
    Hello SVG  
  </text>  
</svg>
```

要放入的文字

x: x軸位置

y: y軸位置(baseline文字基線)

# SVG 填色與外框



# fill & stroke

屬性	功能	預設值
fill	將指定顏色填入形狀或文字中	black
stroke	將指定顏色填入形狀外框中	tansparent
stroke-width	外框粗度	1
opacity	整體透明度設定介於0~1之間。 或可使用 rgba(0,0,0, 0.5)對外 框、填色分別設定	1(不透明)

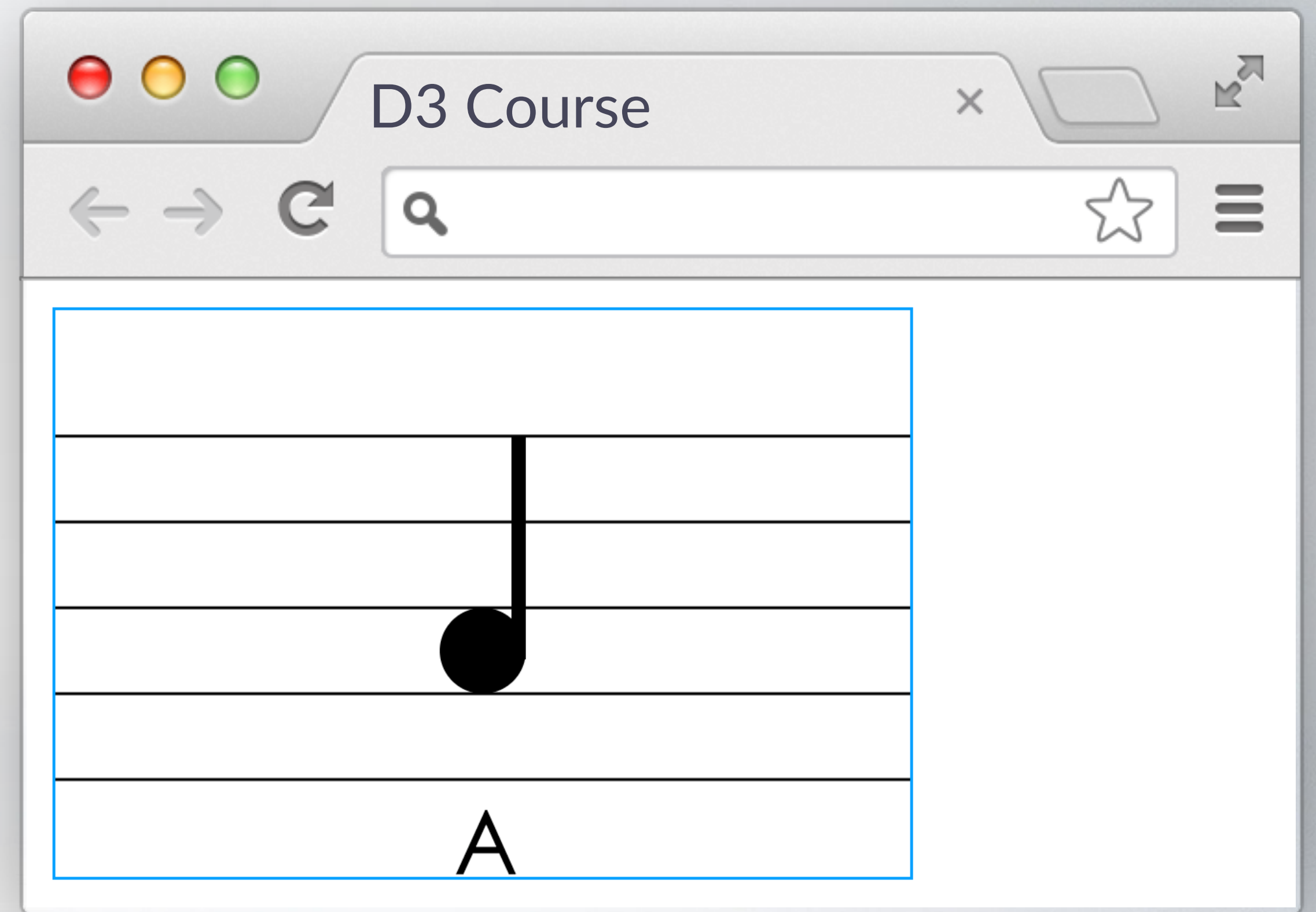




# 動手時間

-動態載入SVG使用法-

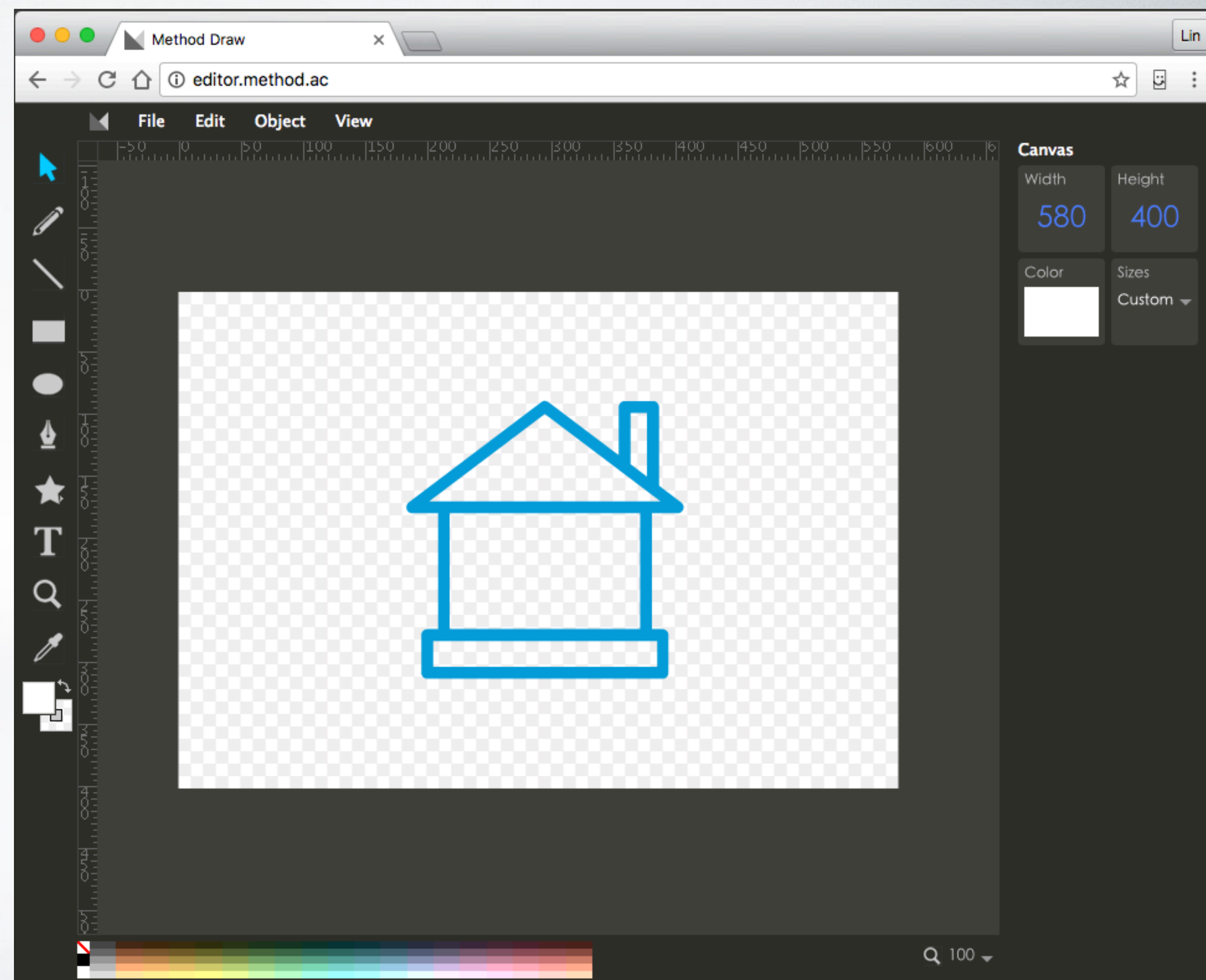
使用D3語法  
加上線譜跟字母





# 工欲善其事...

按我前往





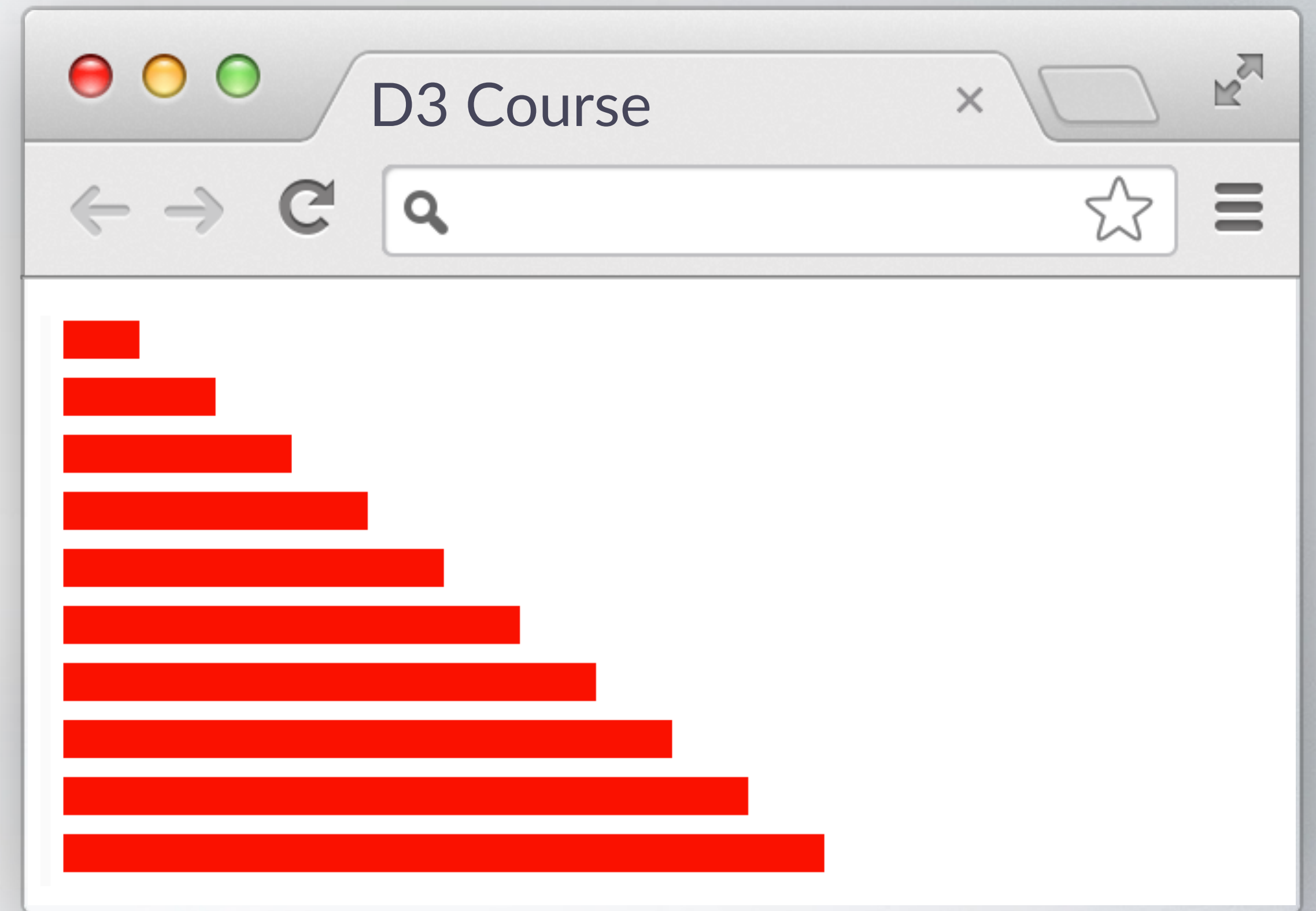
# 繼續前進

-用D3繪製數個長條-

可以怎麼做？

[提示]

1. 10條
2. 每個長條高度10px, 間距5px
3. 第一個長條寬:20px, 之後每個累加20px





-用D3繪製數個長條-

# 以目前所學到的 只能這麼做:

需要10個

```
d3.select("svg").append("rect").attr({ x: 10, y: 10, width: 20, height: 10, fill: "red" });  
d3.select("svg").append("rect").attr({ x: 10, y: 10+15*1, width: 20*2, height: 10, fill: "red" });  
d3.select("svg").append("rect").attr({ x: 10, y: 10+15*2, width: 20*3, height: 10, fill: "red" });  
d3.select("svg").append("rect").attr({ x: 10, y: 10+15*3, width: 20*4, height: 10, fill: "red" });  
d3.select("svg").append("rect").attr({ x: 10, y: 10+15*4, width: 20*5, height: 10, fill: "red" });  
  
...  
  
d3.select("svg").append("rect").attr({ x: 10, y: 10+15*9, width: 20*10, height: 10, fill: "red" });
```

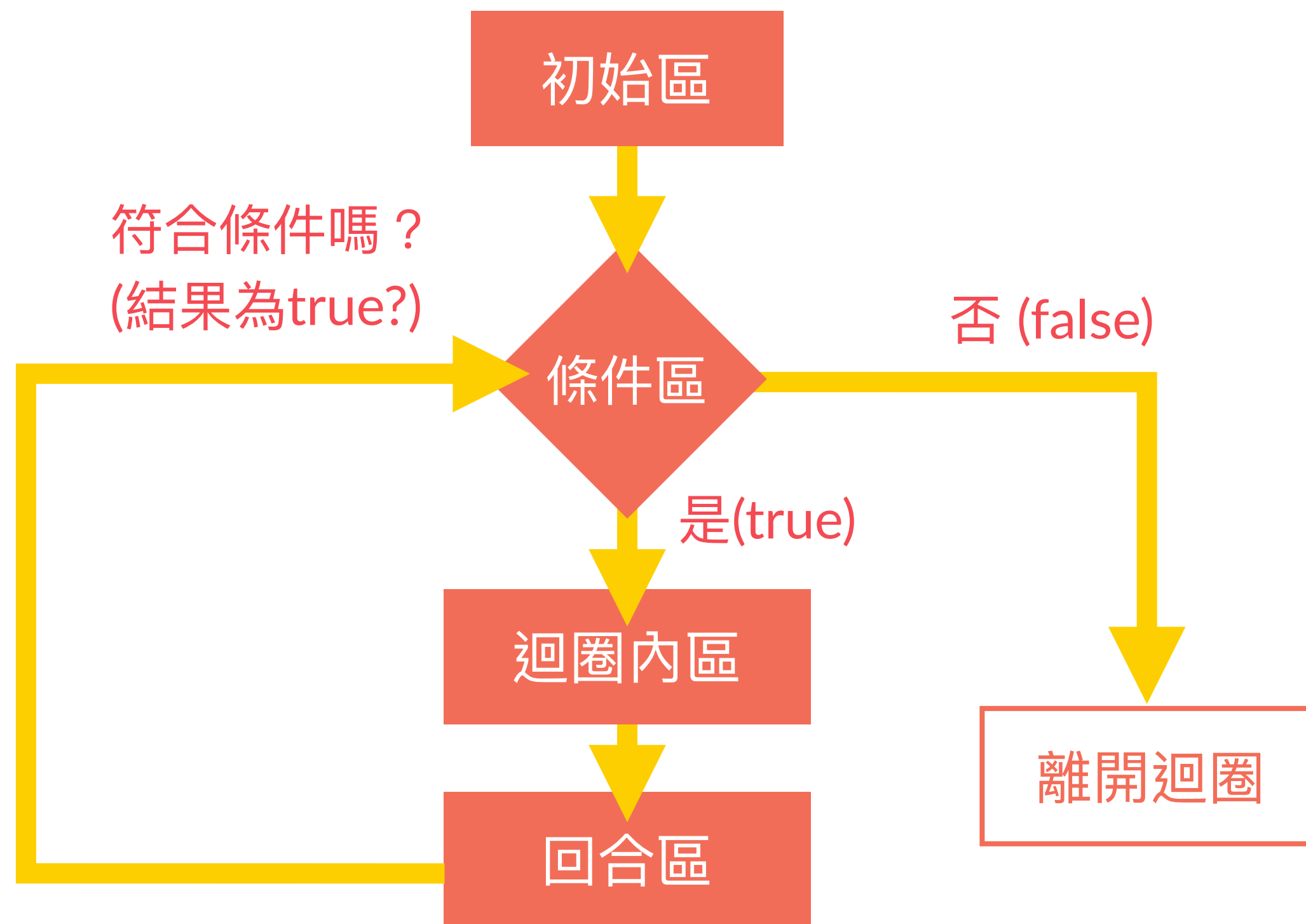
[提示]

1. 10條
2. 每個長條高度10px, 間距5px
3. 第一個長條寬:20px, 之後每個累加20px

## 當需要100000 個長條時，你是否感到害怕？

# FOR 迴圈

重複執行一定次數的程式碼



## for 語法規則

```
for ( 初始區 ; 條件區(不符合退出) ; 回合區 ) {  
    // 迴圈內區  
}
```

試著寫看看:

```
for ( var i = 0 ; i < 10 ; i=i+1 ) {  
    console.log("現在i="+i);  
}
```

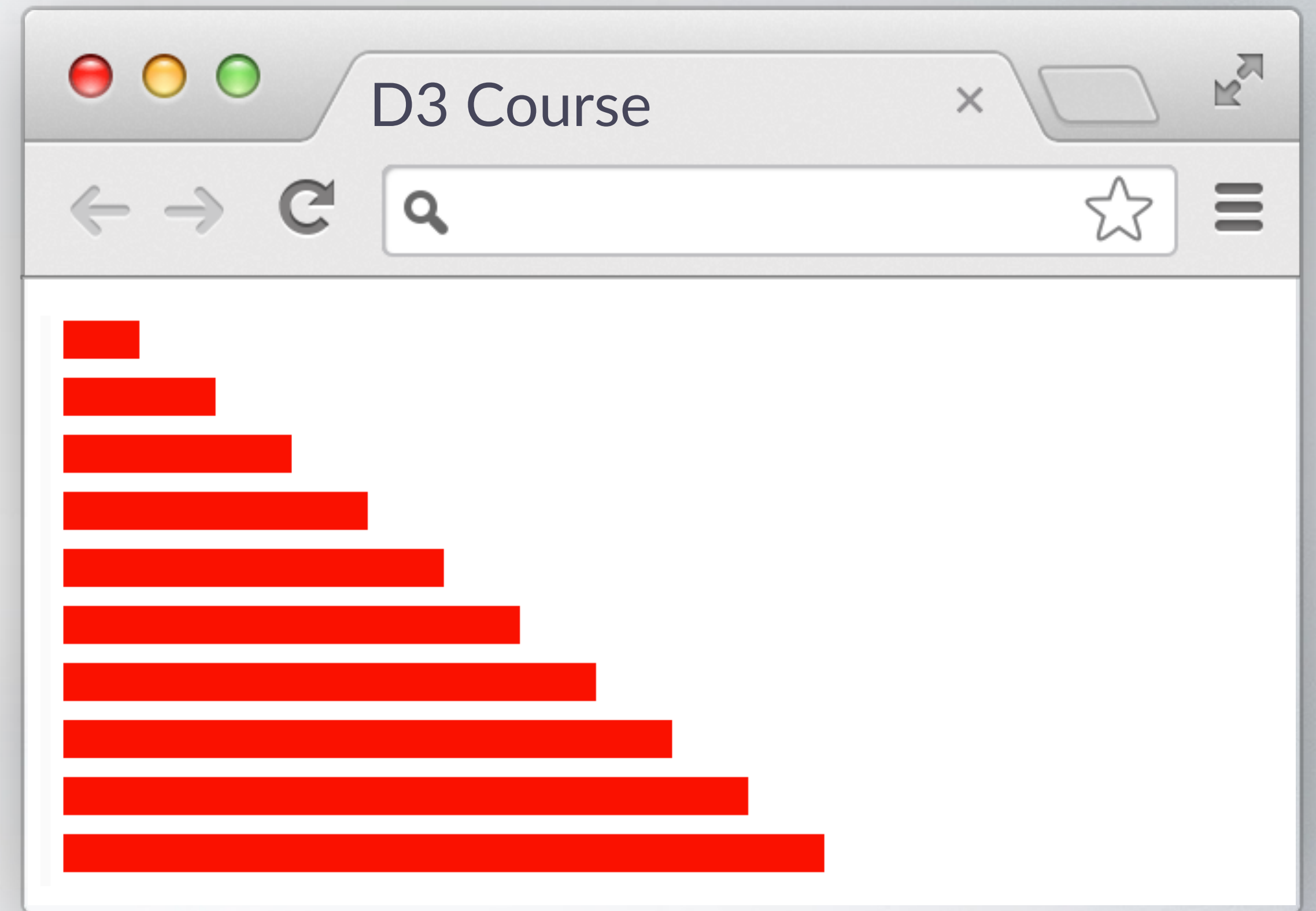


# 用迴圈解決他們

-用D3繪製數個長條-

[提示]

1. 10條
2. 每個長條高度10px, 間距5px
3. 第一個長條寬:20px, 之後每個累加20px





# -用D3繪製數個長條-

## 如何變成for迴圈

需要10個

```
d3.select("svg").append("rect").attr({ x: 10, y: 10+15*0, width: 20*1, height: 10, fill: "red" });  
d3.select("svg").append("rect").attr({ x: 10, y: 10+15*1, width: 20*2, height: 10, fill: "red" });  
d3.select("svg").append("rect").attr({ x: 10, y: 10+15*2, width: 20*3, height: 10, fill: "red" });  
d3.select("svg").append("rect").attr({ x: 10, y: 10+15*3, width: 20*4, height: 10, fill: "red" });  
d3.select("svg").append("rect").attr({ x: 10, y: 10+15*4, width: 20*5, height: 10, fill: "red" });  
  
...  
  
d3.select("svg").append("rect").attr({ x: 10, y: 10+15*9, width: 20*10, height: 10, fill: "red" });
```

[提示]

1. 10條
2. 每個長條高度10px, 間距5px
3. 第一個長條寬:20px, 之後每個累加20px



-用D3繪製數個長條-

# 有了for迴圈，現在你可以這麼做

```
for(var i=0; i<10; i++){  
  d3.select("svg").append("rect").attr({ x: 10, y: 10+15*i, width: 20*(i+1), height: 10, fill: "red" });  
}
```

## 需要100000 個長條時，你從此不怕了

[提示]

1. 10條
2. 每個長條高度10px, 間距5px
3. 第一個長條寬:20px, 之後每個累加20px





# 動手時間

用迴圈搭配亂數函式  
產生20條有數值的長條圖

[提示]

1. 20條的長度值介於20px到300px之間
2. 每個長條高度:10px，彼此間距2px
3. 亂數值距長條末端15px處，字大小: 12px



[注意]  
每次結果不盡相同  
但規則不變!



# 答案檢查



我確實寫完了才看答案



# 如何做長方形？



## CSS

```
.rect{  
    width:300px;  
    height:150px;  
    background:gold;  
}
```

## HTML

```
<div class="rect"></div>
```



# 如何做正方形？



## CSS

```
.square{  
    width:300px;  
    height:300px;  
    background:gold;  
}
```

## HTML

```
<div class="square"></div>
```



# 如何做有圓角正方形？



## CSS

```
.radius-square{  
    width:300px;  
    height:300px;  
    border-radius:8px;  
    background:gold;  
}
```

## HTML

```
<div class="radius-square"></div>
```



# 小小測驗時間

其實你會做一個圓形 (?)





# 你答對了嗎？

## CSS

```
.circle{  
    width:300px;  
    height:300px;  
    border-radius:50%;  
    background:gold;  
}
```

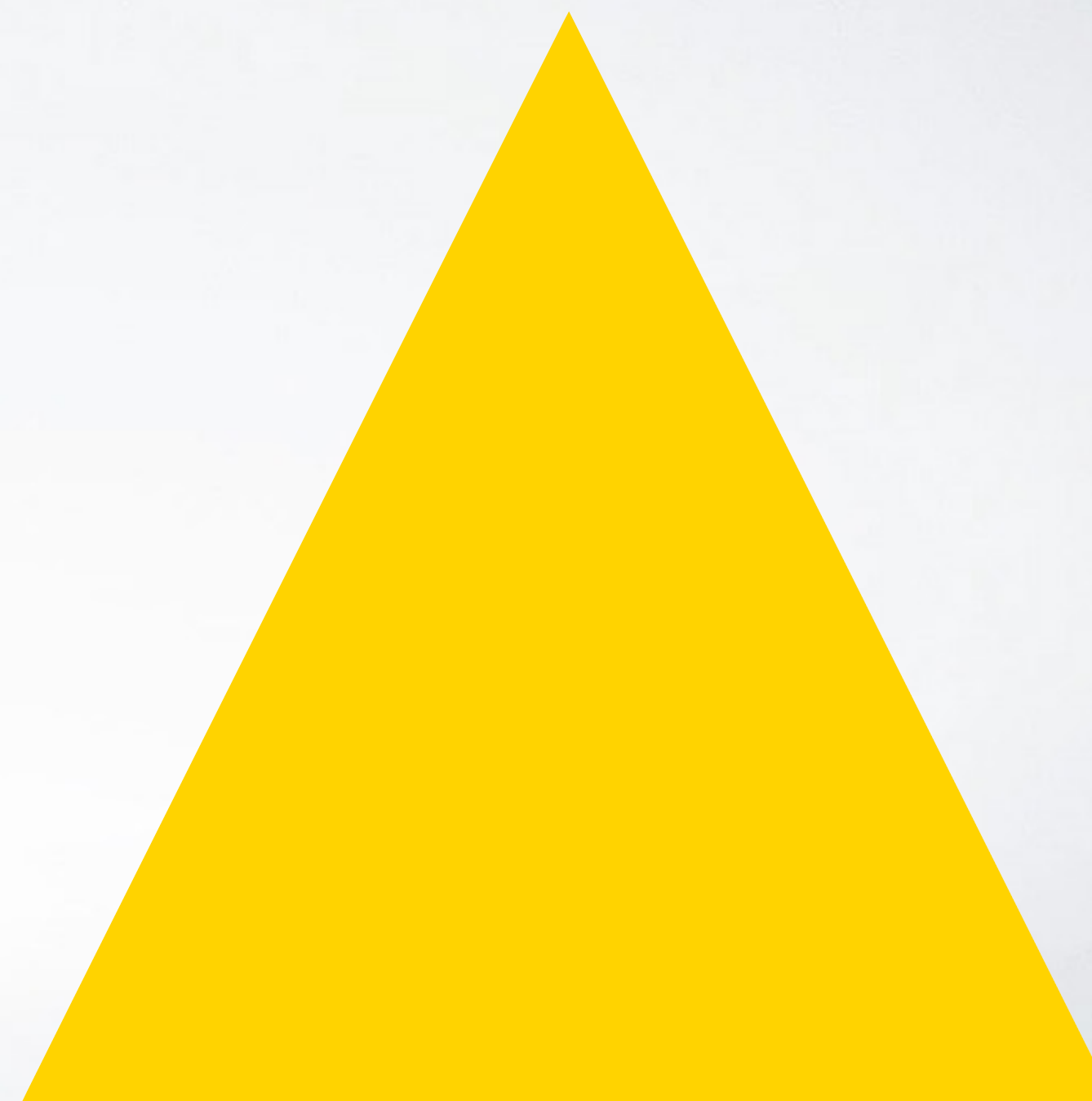
## HTML

```
<div class="circle"></div>
```



# 三角形 有辦法畫出來嗎？

其實你學過！





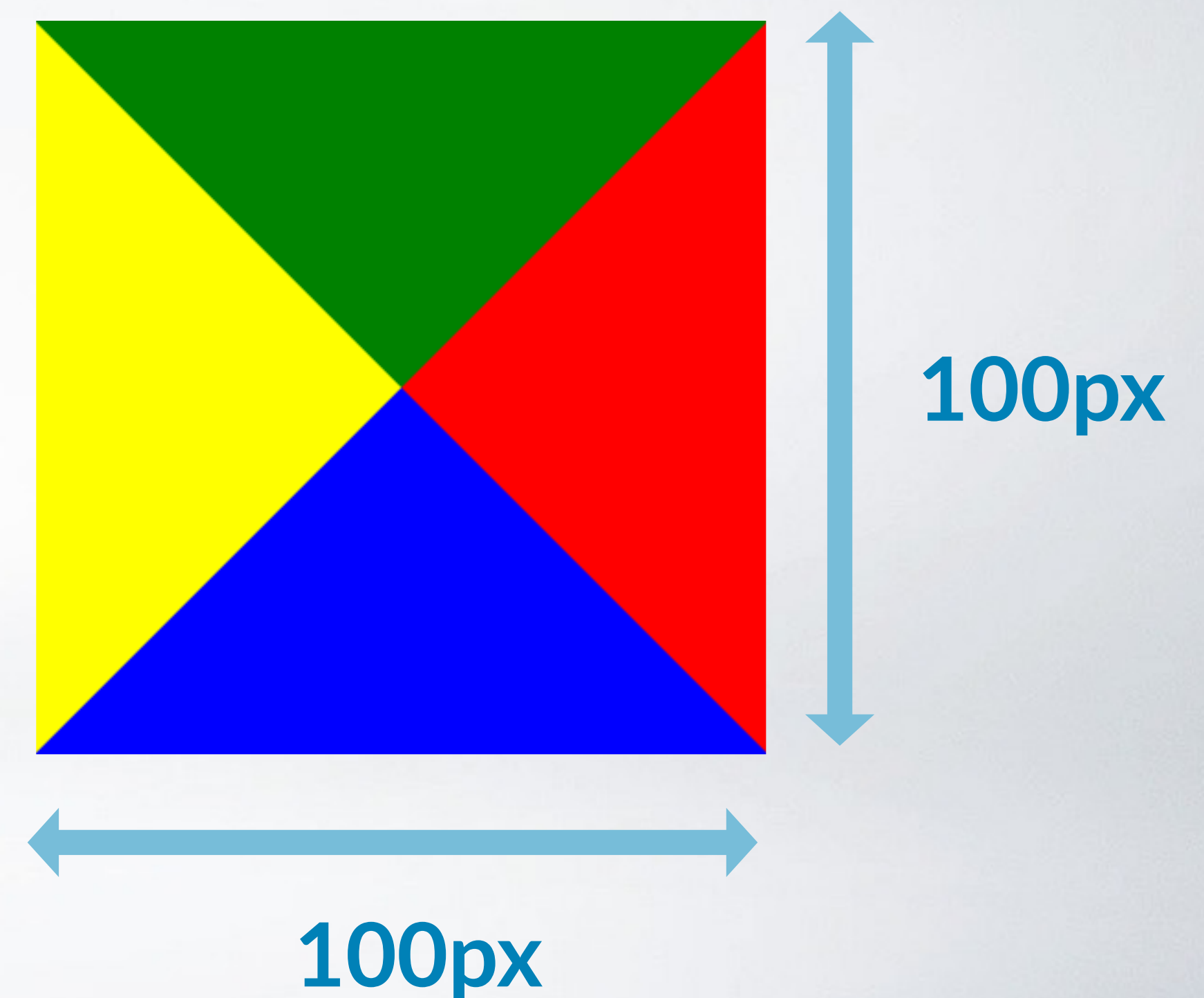
# 動手時間

先來做出這樣的圖形  
然後有看出什麼嗎？

[溫馨小提示]

1.用border 上下左右分別設定

2.content寬高都為0, padding也都為0





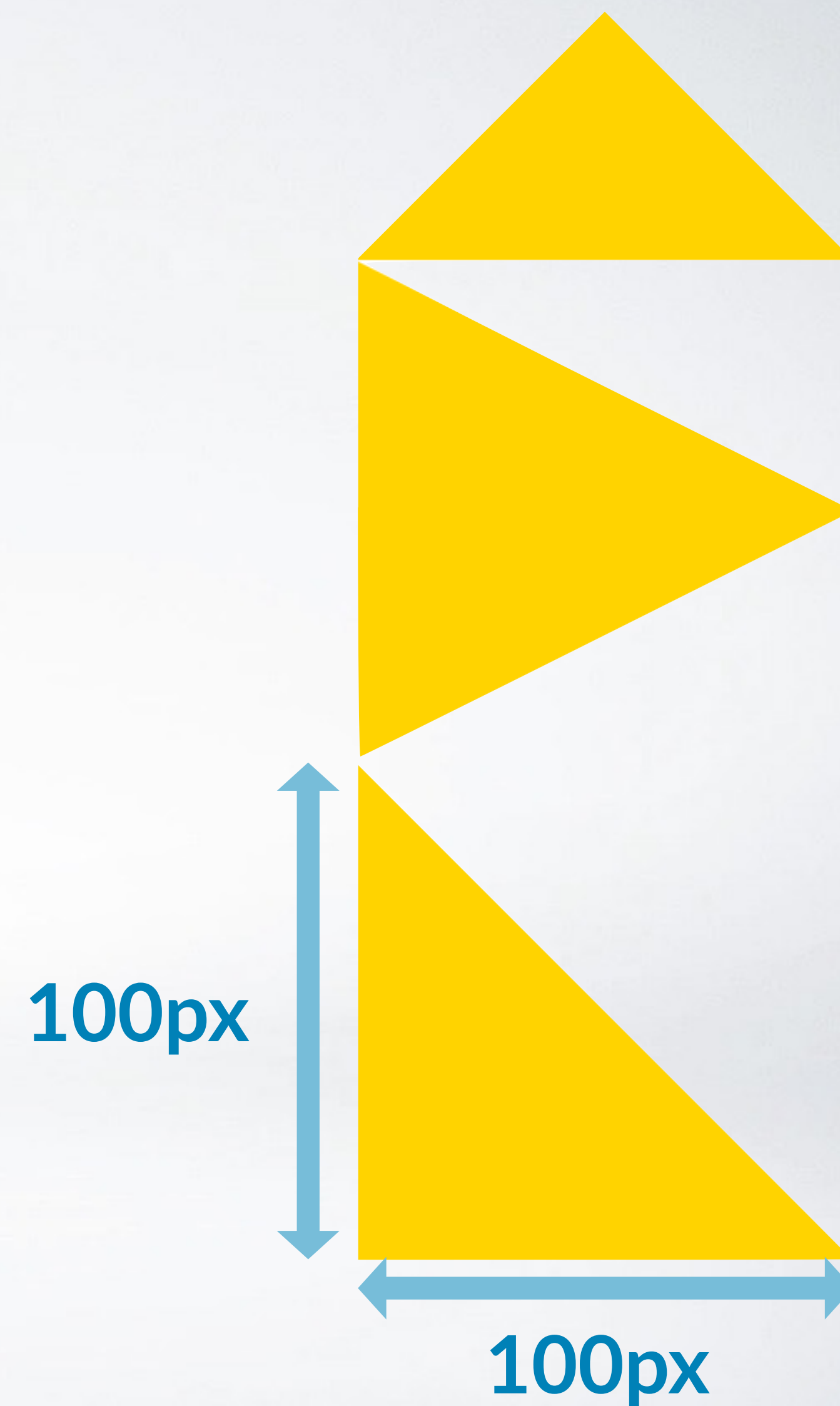
# 你答對了嗎？

```
<div style="background-color:lightblue;width:0px;height:0px;border-top:50px solid green;border-bottom: 50px solid blue;border-right: 50px solid red;border-left: 50px solid yellow;"></div>
```



# 三種三角形練習

練習: 等腰、正、直角三角形





# 你答對了嗎？

```
<div style="width:0px;  
height:0px;  
border-bottom: 50px solid gold;  
border-right: 50px solid transparent;  
border-left: 50px solid transparent;"></div>
```

```
<div style="width:0px;  
height:0px;  
border-top: 50px solid transparent;  
border-left: 100px solid gold;  
border-bottom: 50px solid transparent;"></div>
```

```
<div style="width:0px;  
height:0px;  
border-bottom: 100px solid gold;  
border-right: 100px solid transparent;"></div>
```