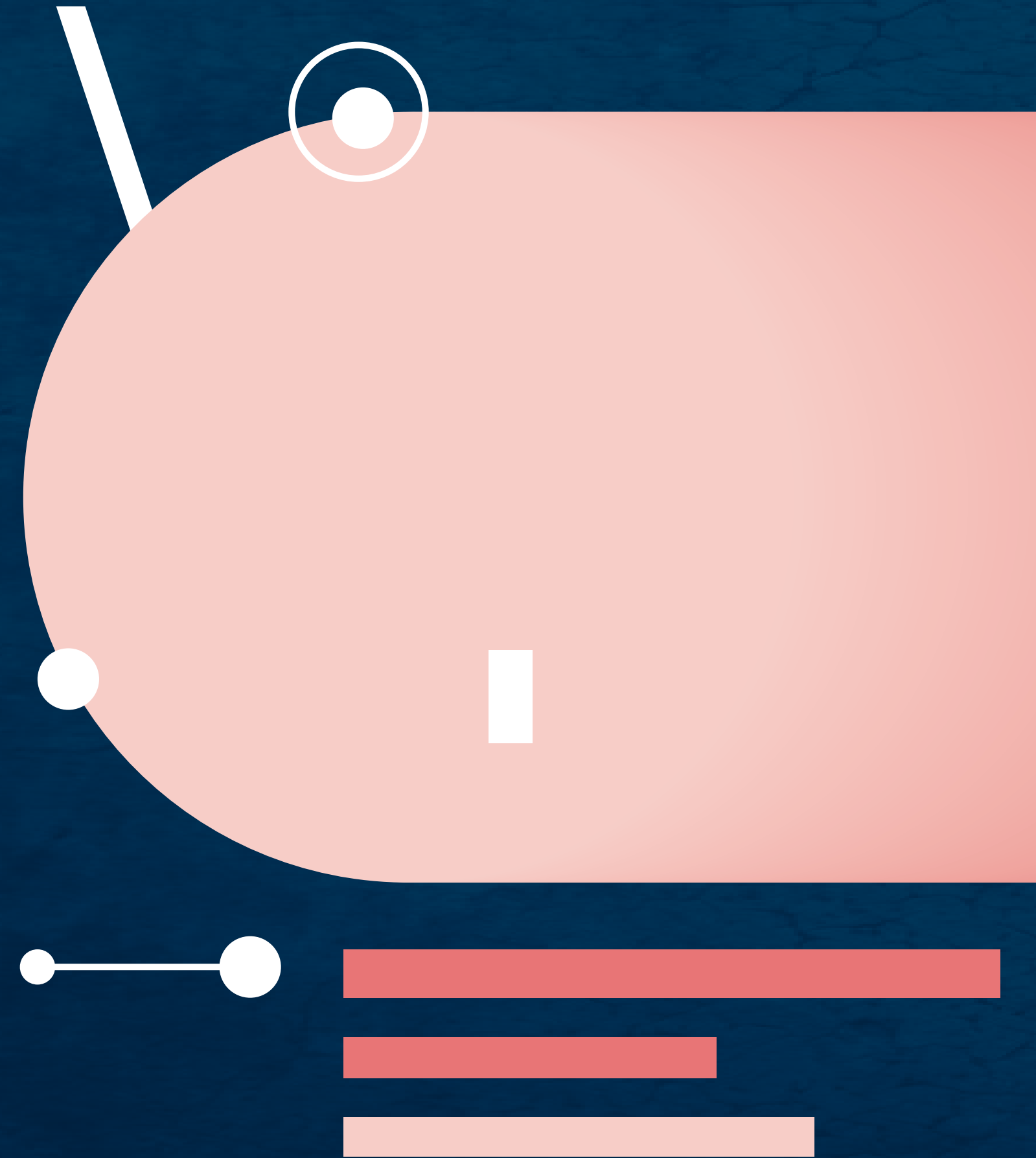


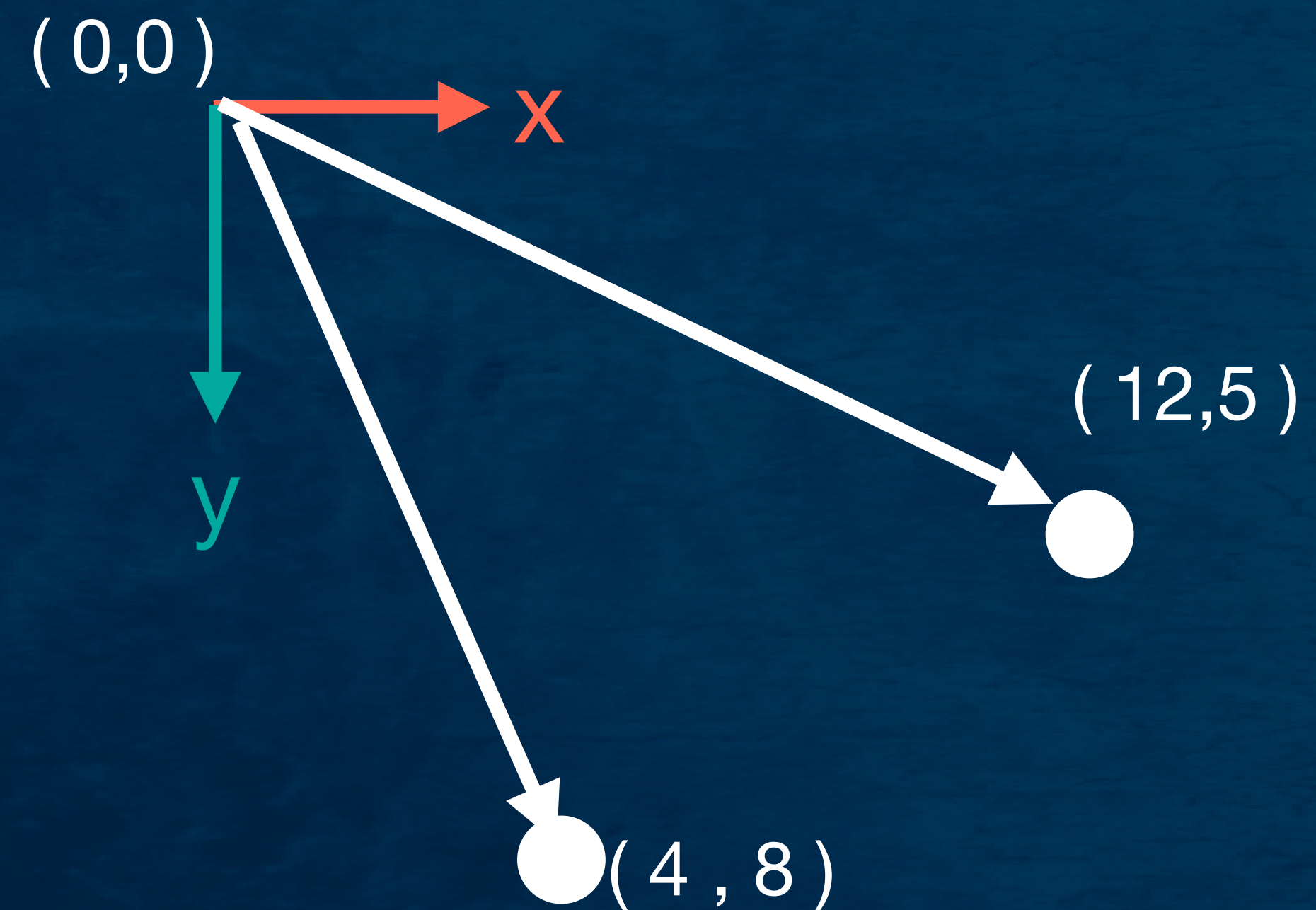
# Vector 向量的概念

Canvas與特效動畫

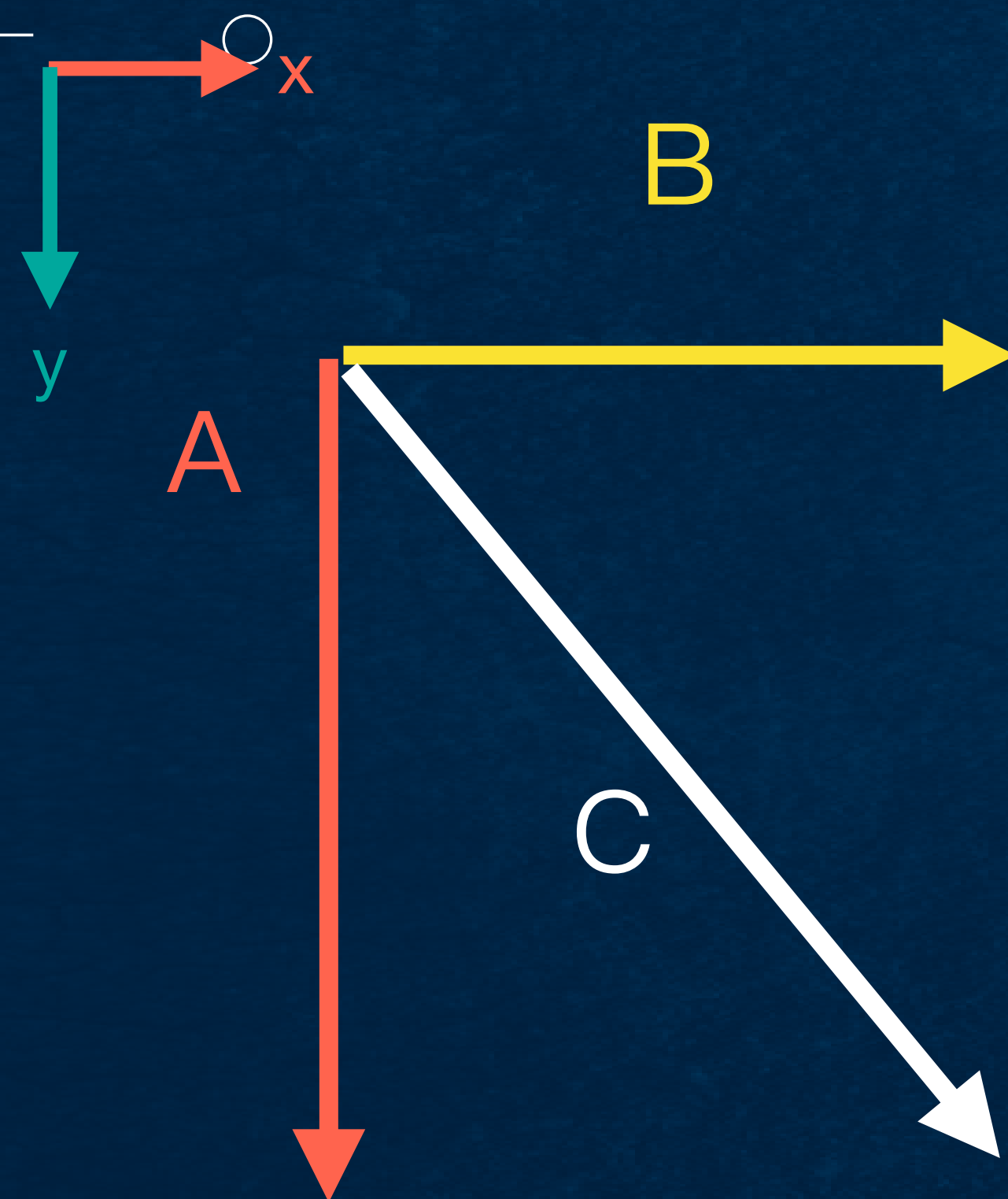


# 為什麼需要向量？

- 是有方向的「變化量」
- 注重「圖形化的幾何意義」
- 事實上，點就是「相對原點的向量」



# 向量加法



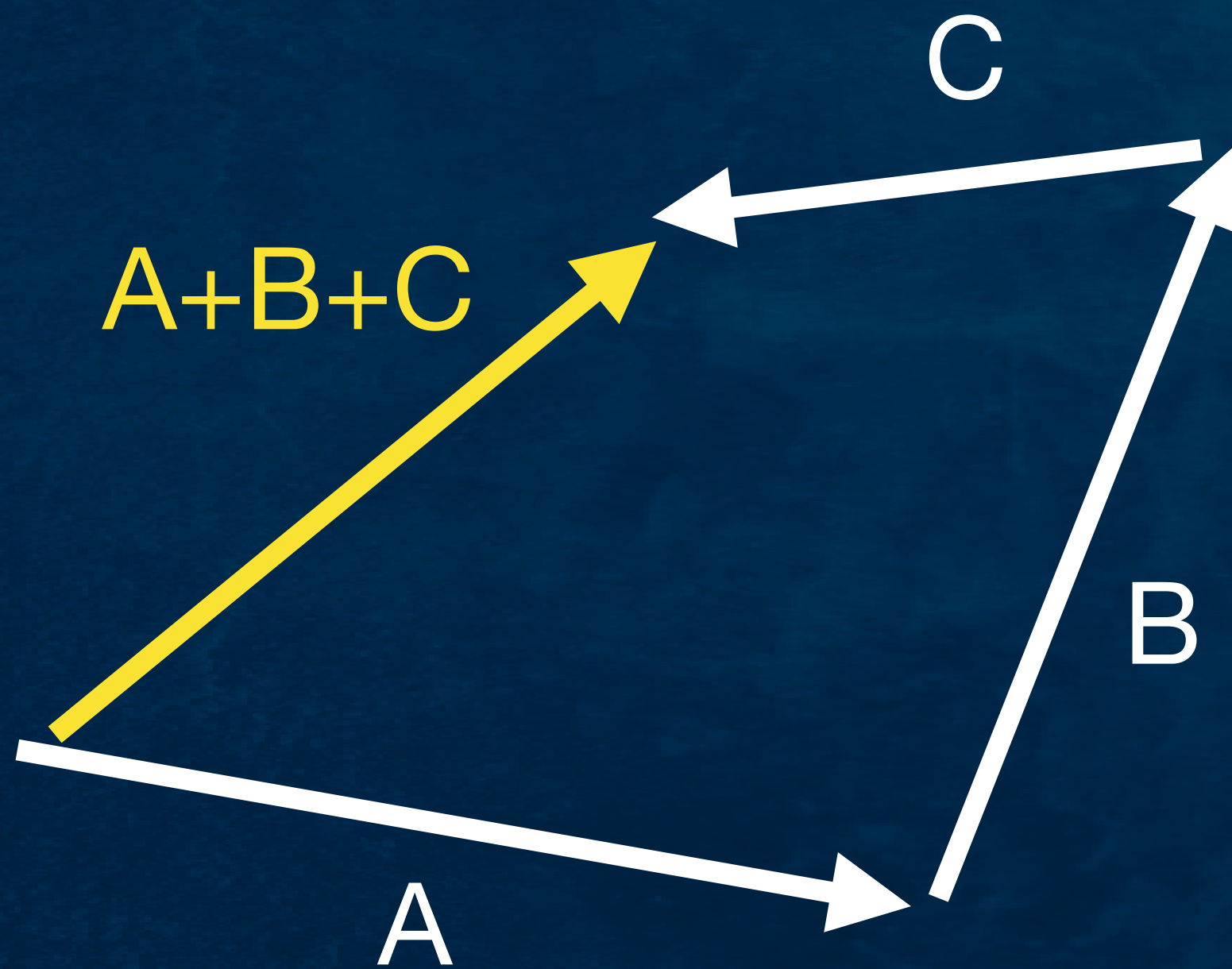
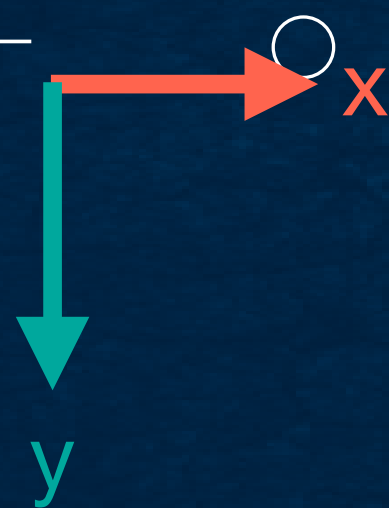
$$A = (0, 4)$$

$$B = (3, 0)$$

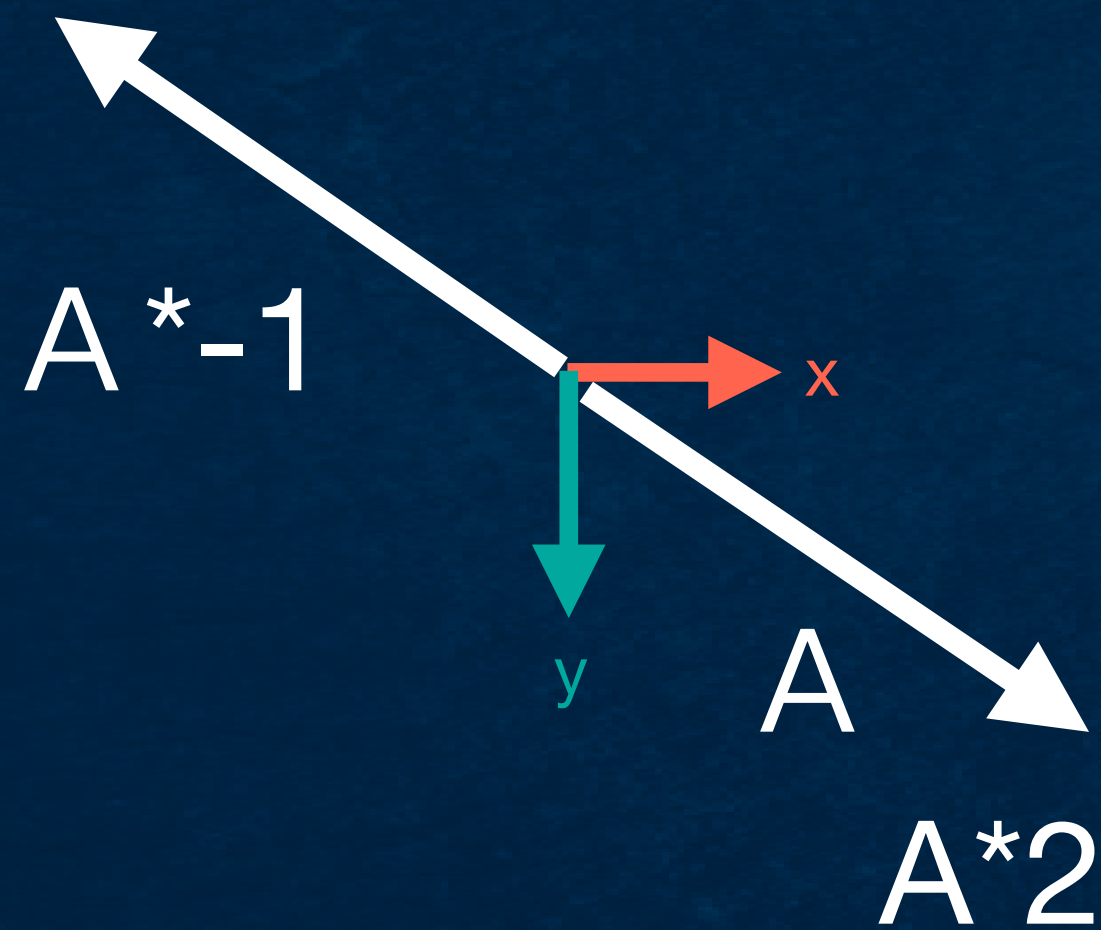
$$C = A + B \quad // \quad (3, 4)$$



# 向量加法



# 向量縮放



保持角度縮放長度

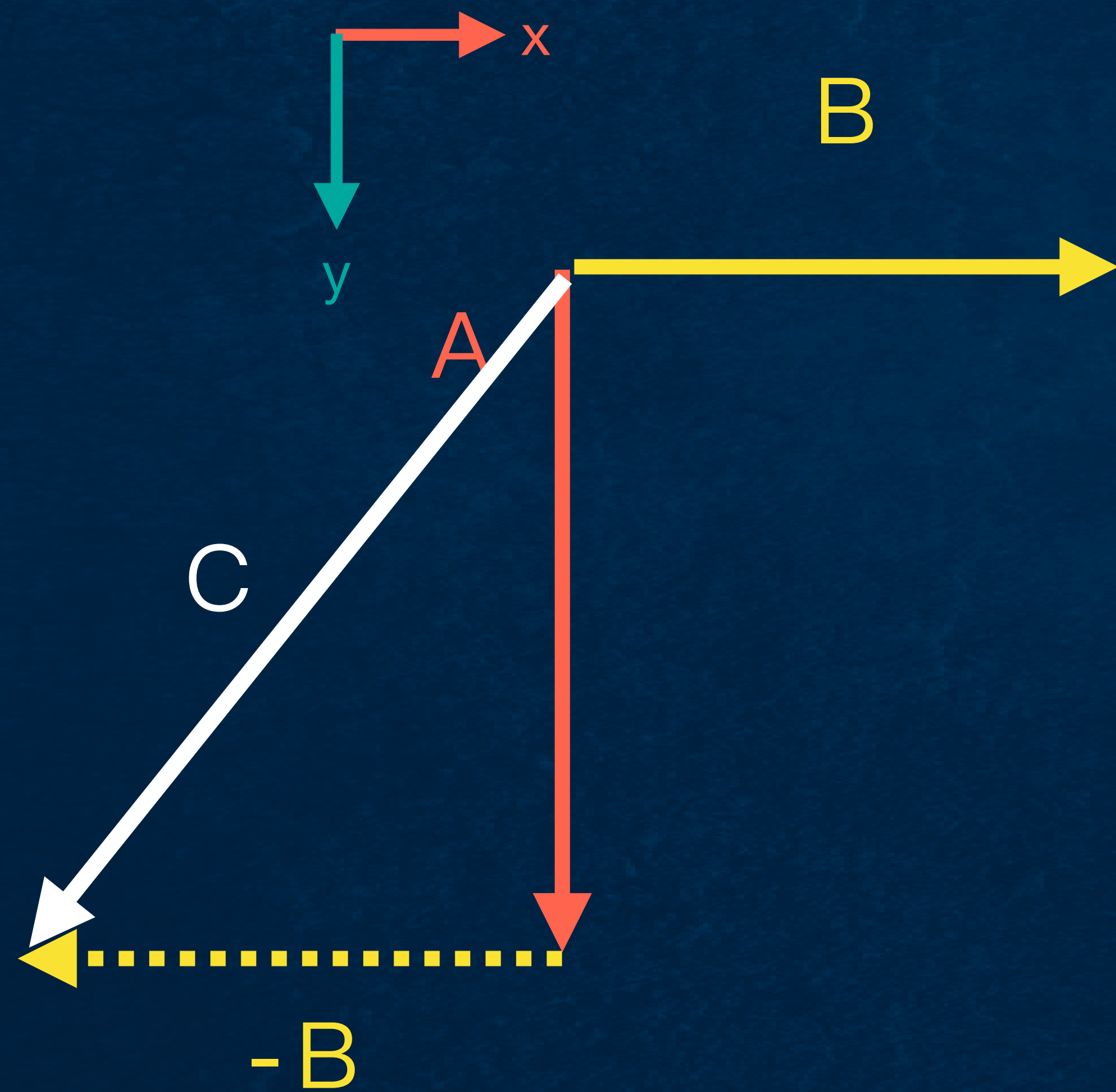
負長度為反方向

$$A = (4, 3)$$

$$A * 2 \ // \ (8, 6)$$

$$A * -1 \ // \ (-4, -3)$$

# 向量減法



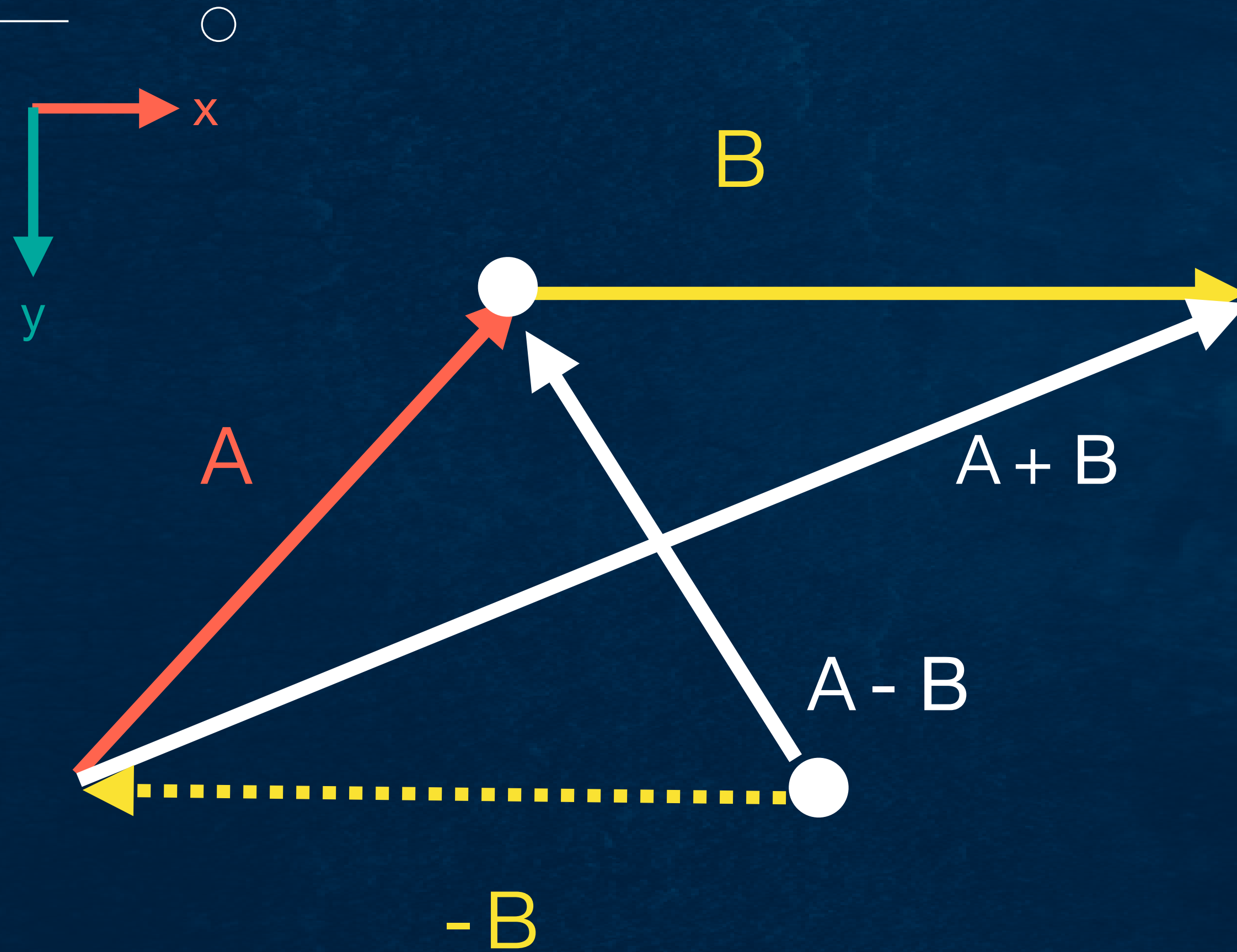
$$A = (0, 4)$$

$$B = (3, 0)$$

$$C = A - B = A + (-B) // (3, 4)$$



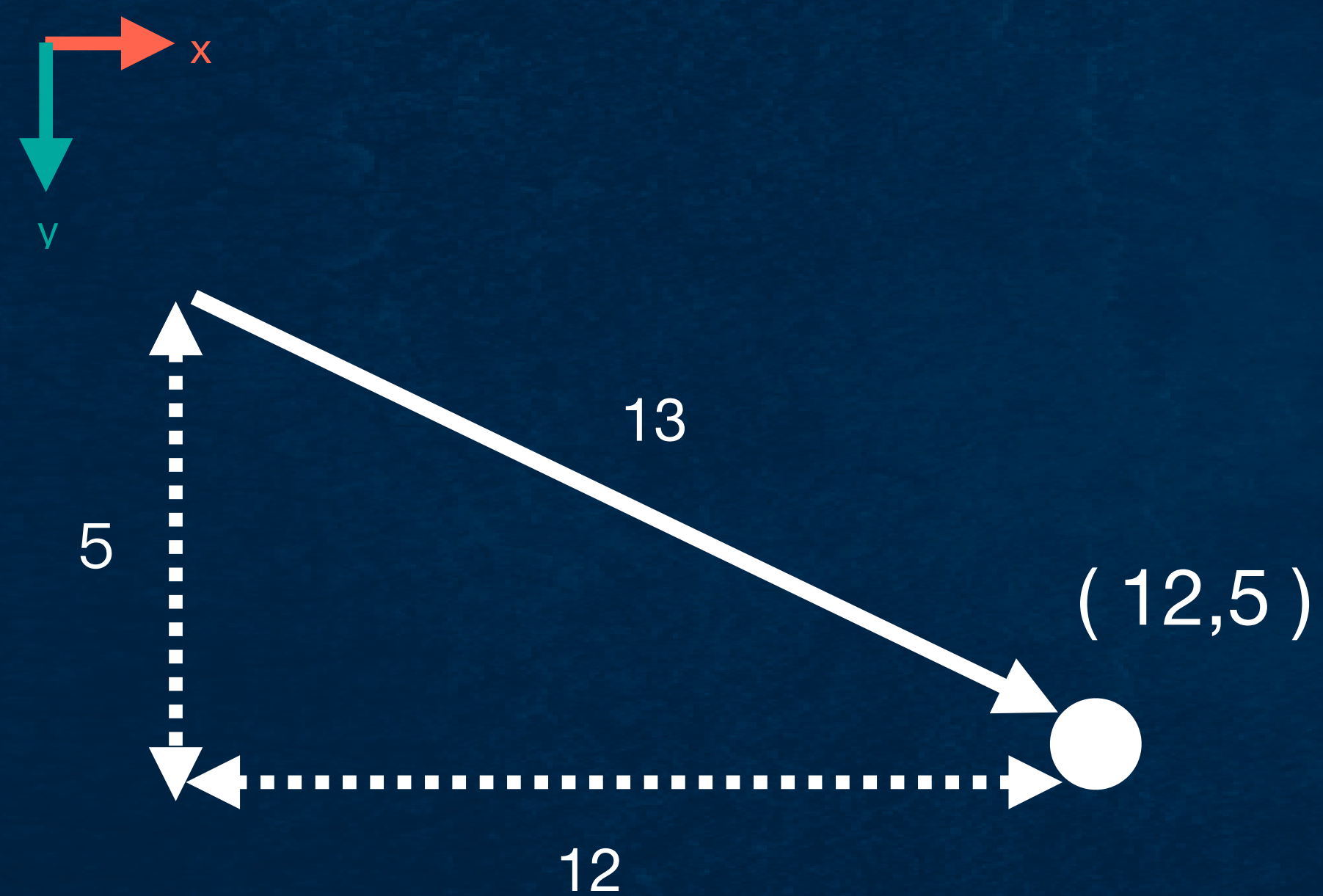
# 向量幾何



相加 - 平行四邊形的長對角線

相減 - 平行四邊形的短對角線

# 向量的長度



利用畢氏定理 - 兩股合

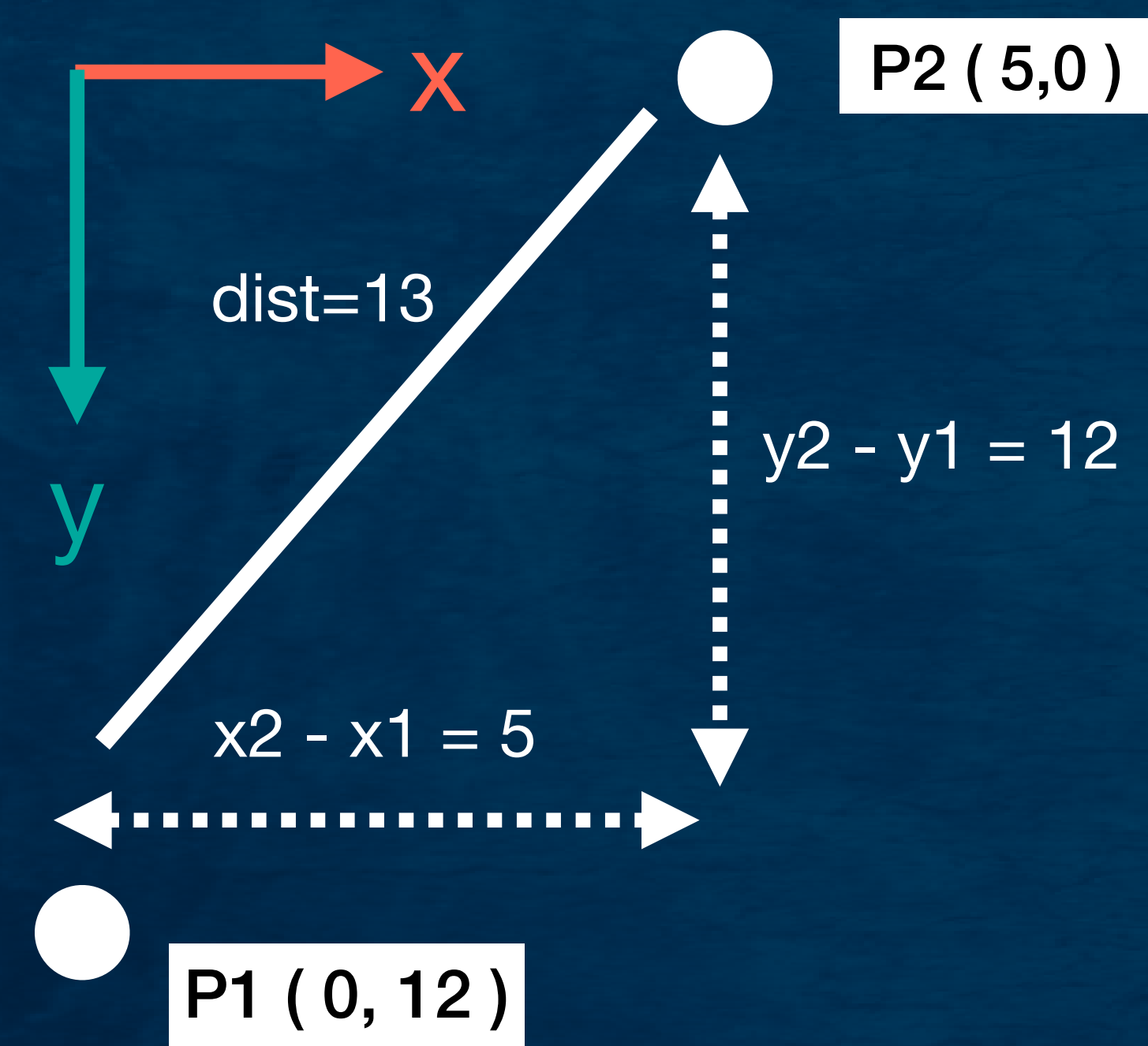
$$A = (12, 5)$$

$$|A| = 13$$



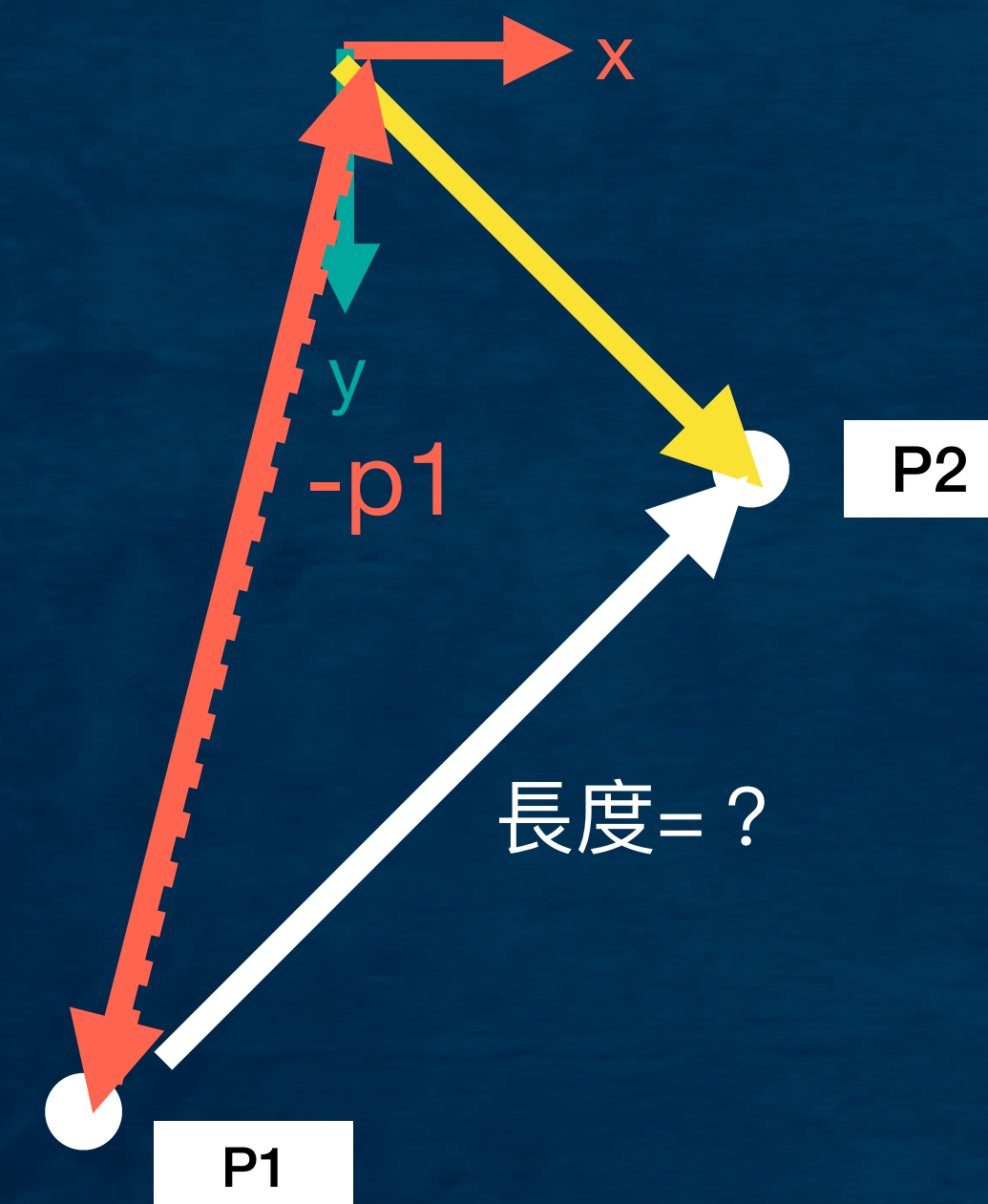
# 舉例：兩點之間距離計算

- Distance =  $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$
- 純量意義：
- 直角三角形斜邊長為 兩股的平方和開根號



# 舉例：兩點之間距離計算

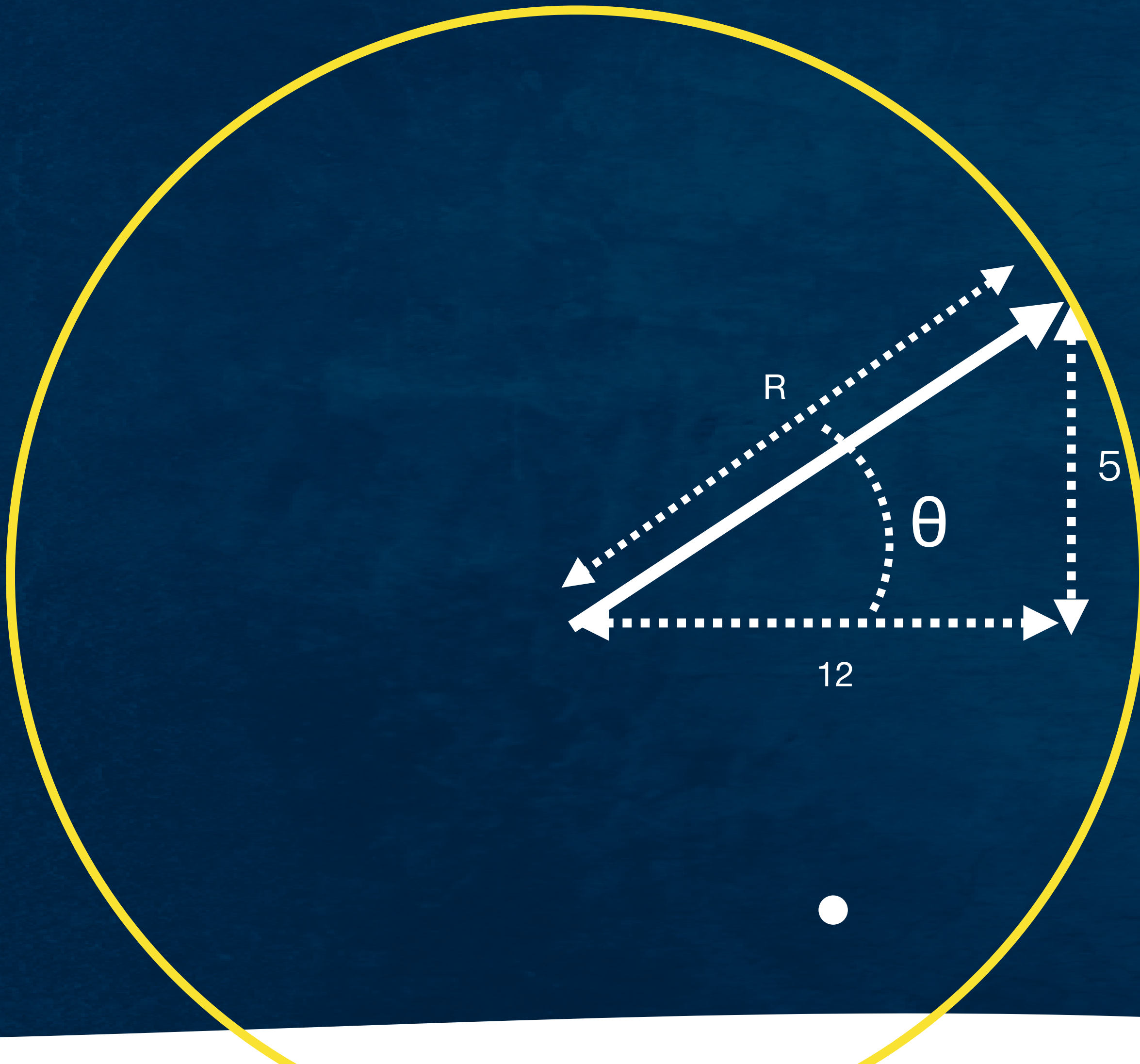
- Distance = ( P2 - P1 ) 的長度
- 向量意義：
- 把P2 加上 P1的反向，新形成向量的長度
- $P2 - P1 = P2 + (- P1)$





# 程式中的向量操作

- 向量加減法 add / sub
- 向量乘法 mul
- 向量長度 length
- 向量角度 angle
- 單位向量 unit





# 程式中的向量 - 位置更新

原本作法：

$P = \{x: 1, y: 2\}$

$V = \{x: 6, y: 6\}$

$P.x += V.x$

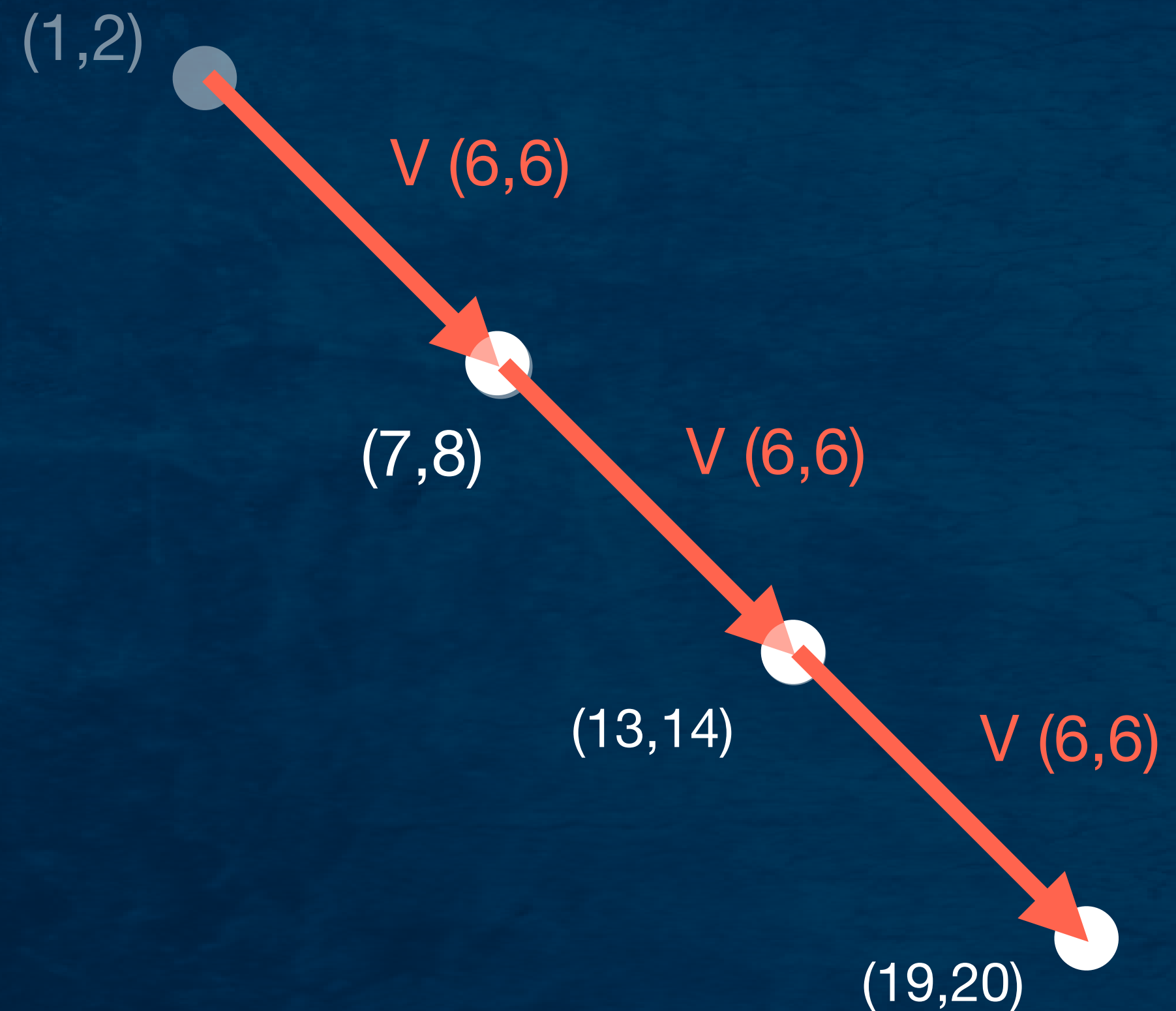
$P.y += V.y$

向量做法

$P = \text{new Vector}(1,2)$

$V = \text{new Vector}(6,6)$

$P = P.\text{add}(V)$



# 程式中的向量 - 速度更新

## 原本作法

$V = \{x: 5, y: -1\}$

$A = \{x: 0, y: 2\}$

$V.x += A.x$

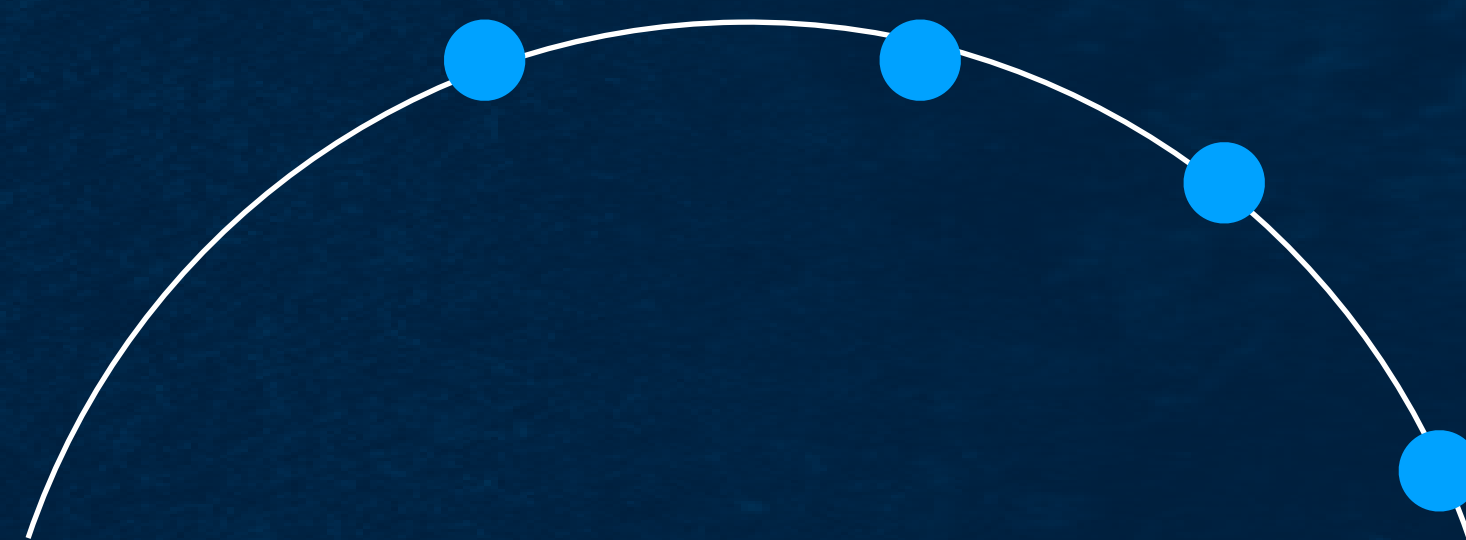
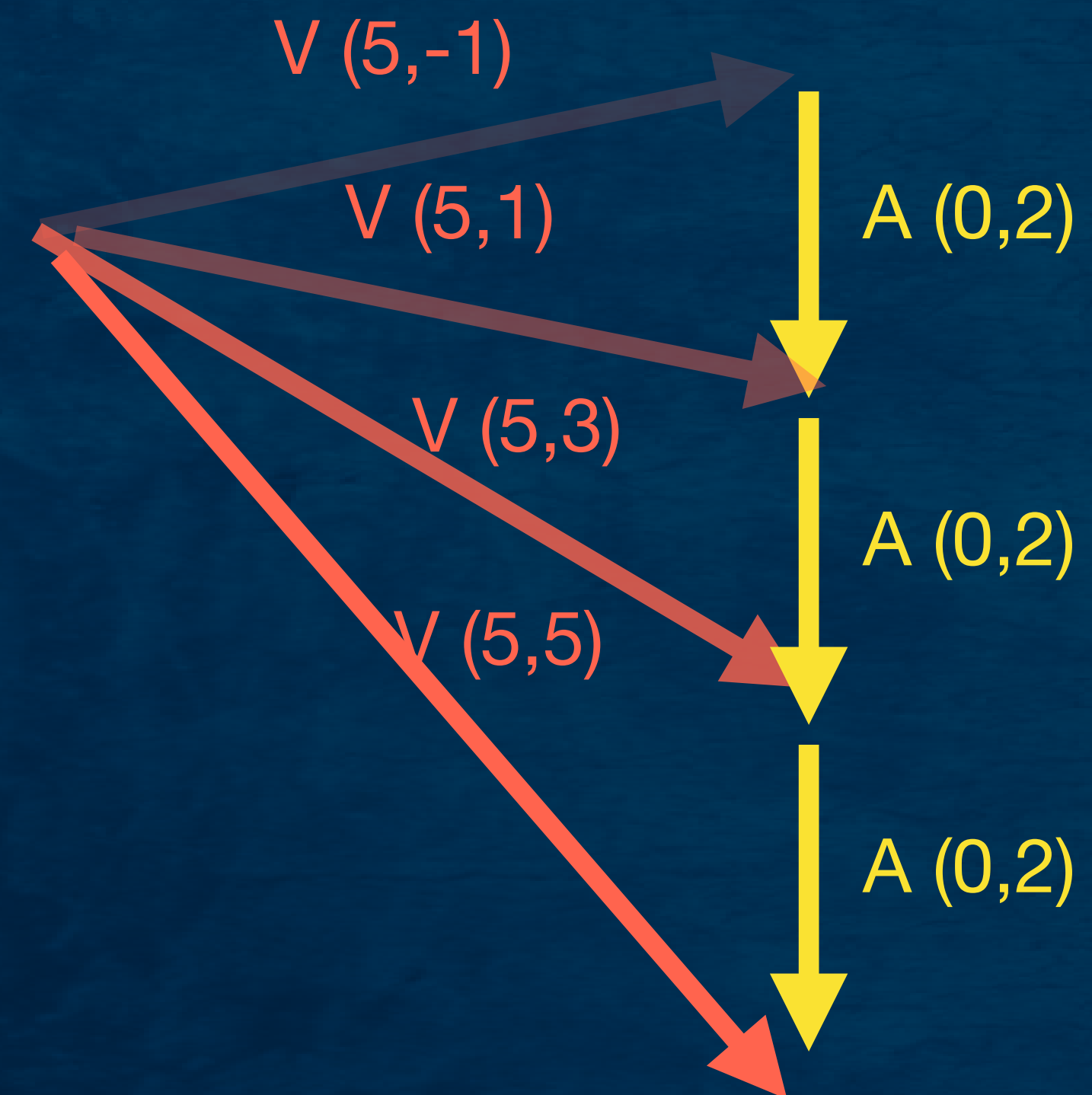
$V.y += A.y$

## 向量做法

$V = \text{new Vector}(5, -1)$

$A = \text{new Vector}(0, 2)$

$V = V.\text{add}(A)$





# 程式中的向量 - 速度衰減

## 原本作法

$V = \{x: 5, y: -1\}$

$V.x *= 0.9$

$V.y *= 0.9$

## 向量做法

$V = \text{new Vector}(5, -1)$

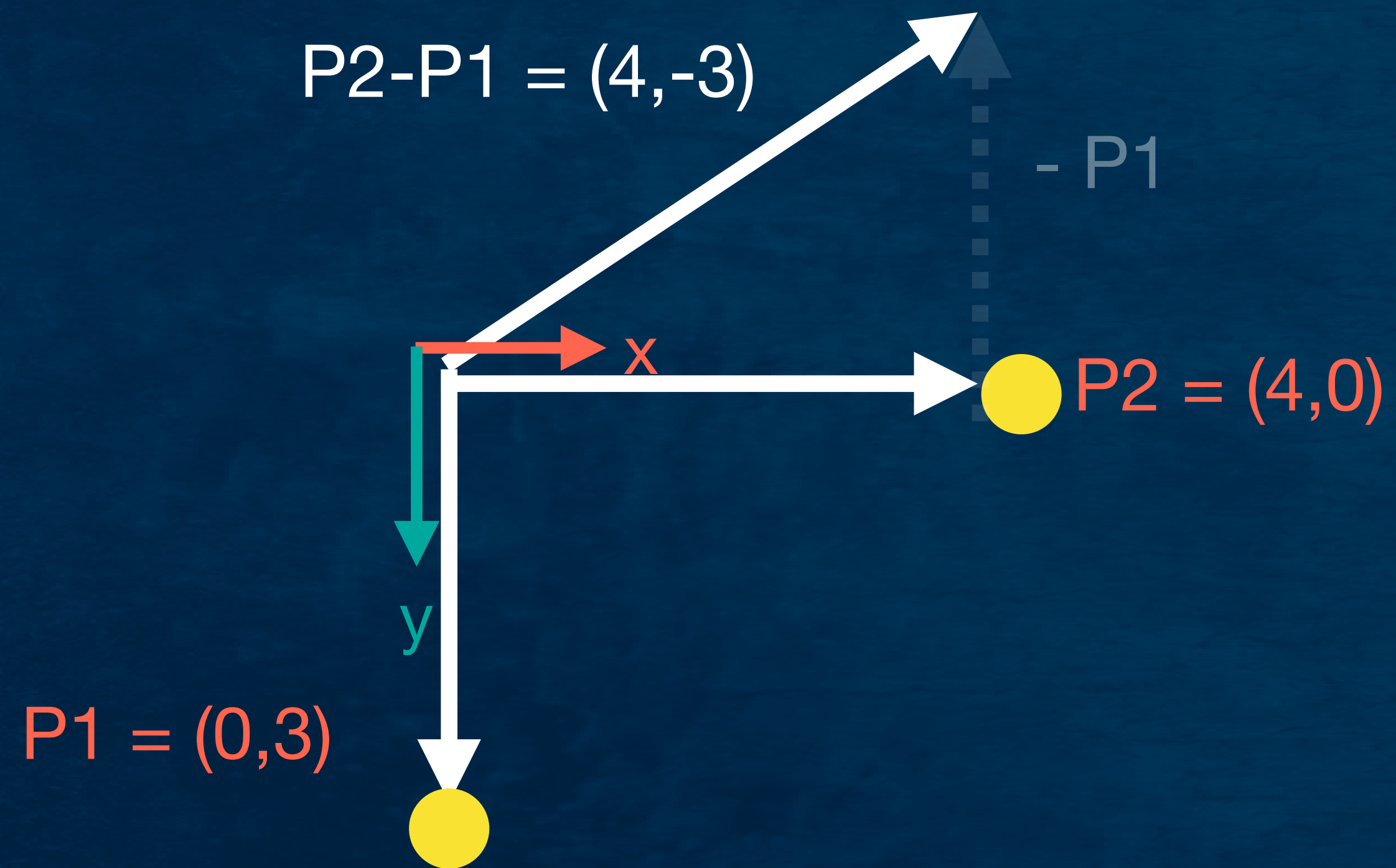
$V = V.\text{mul}(0.9)$





# 應用 - 判斷兩點之間距離

- $P1 = \text{new Vector}(0,3)$
- $P2 = \text{new Vector}(4,0)$
- $\text{Dist} = P2.\text{sub}(P1).\text{length}()$



# ● 程式中的向量- 繪製一個點到滑鼠的線

純量作法 - 兩點連線

$P = \{x: 6, y: 6\}$

$M = \{x: mx, y: my\}$

Translate( P.x , P.y )

MoveTo( 0 , 0 )

LineTo( M.x - P.x, M.y - P.y )

多了角度旋轉

讓線段可以直覺的用長度繪製

向量做法-轉到角度，畫長度

$P = \text{new Vector}(6,6)$

$M = \text{new Vector}(mx, my)$

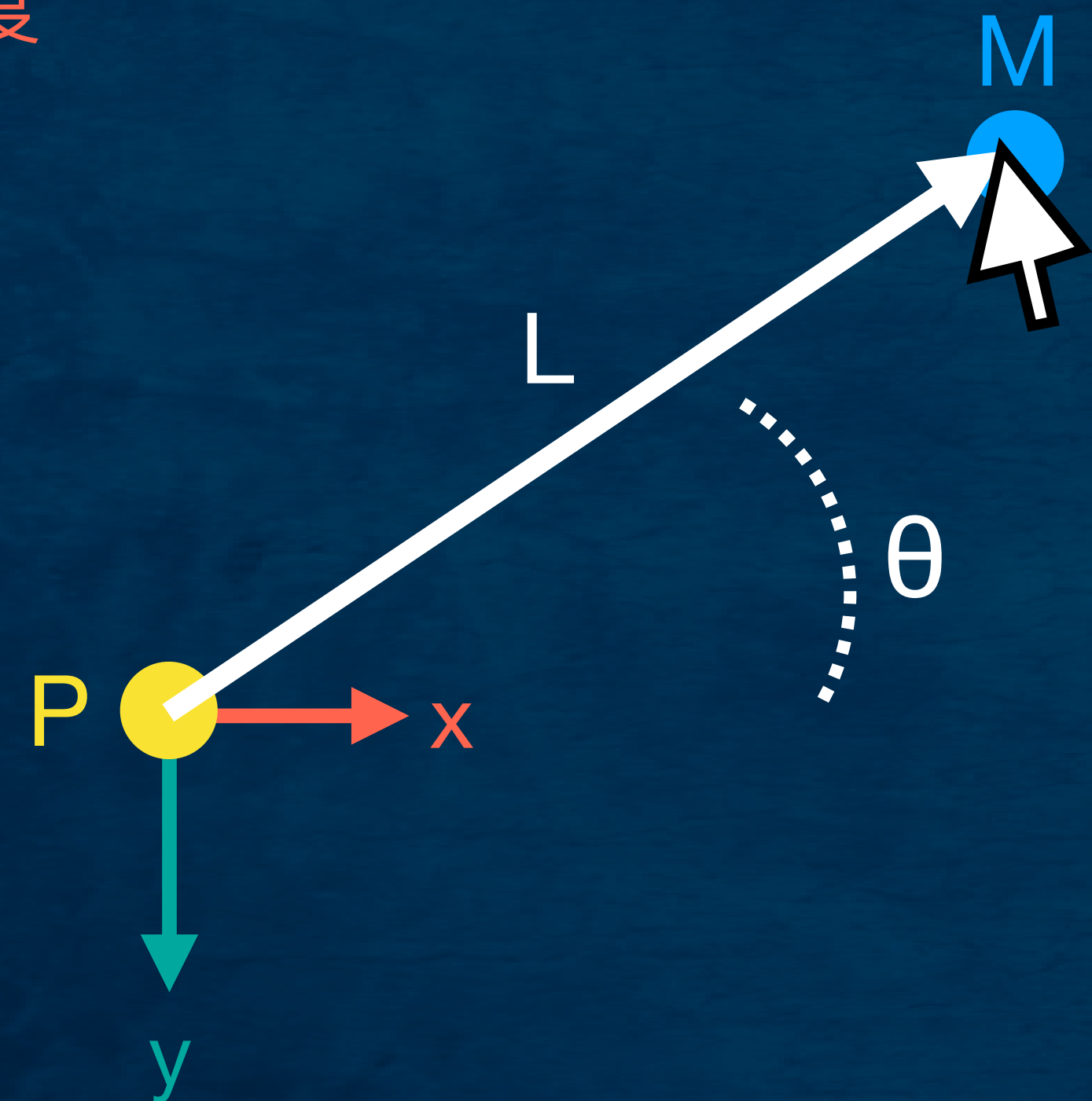
$L = P - M$

Translate( P )

Rotate ( L.angle() )

MoveTo ( 0 , 0 )

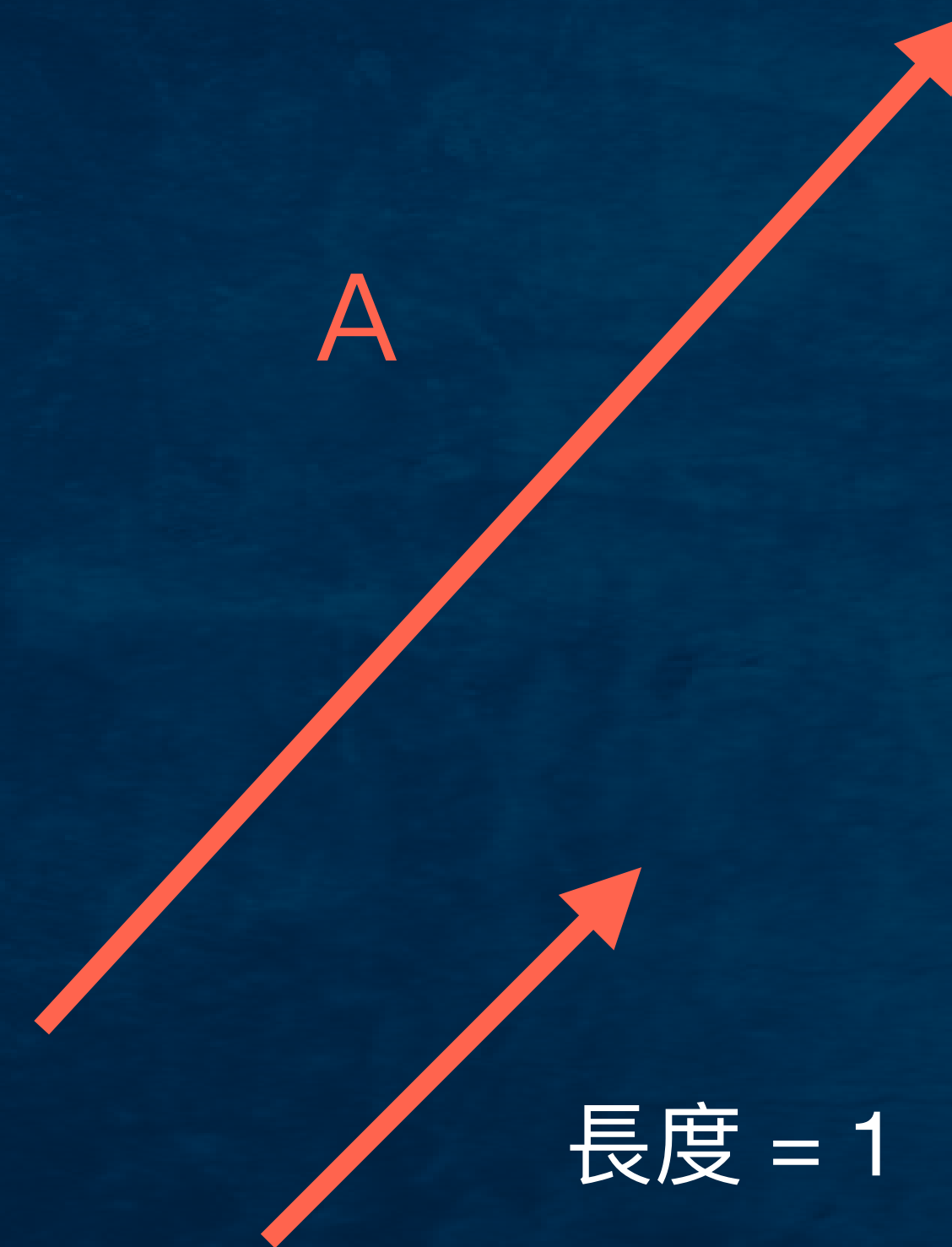
LineTo ( L.length() , 0 )





# Unit 單位向量

- 這個方向其長度為1的向量
- $\text{Unit} = V . \text{mul}( 1 / V . \text{length}() )$





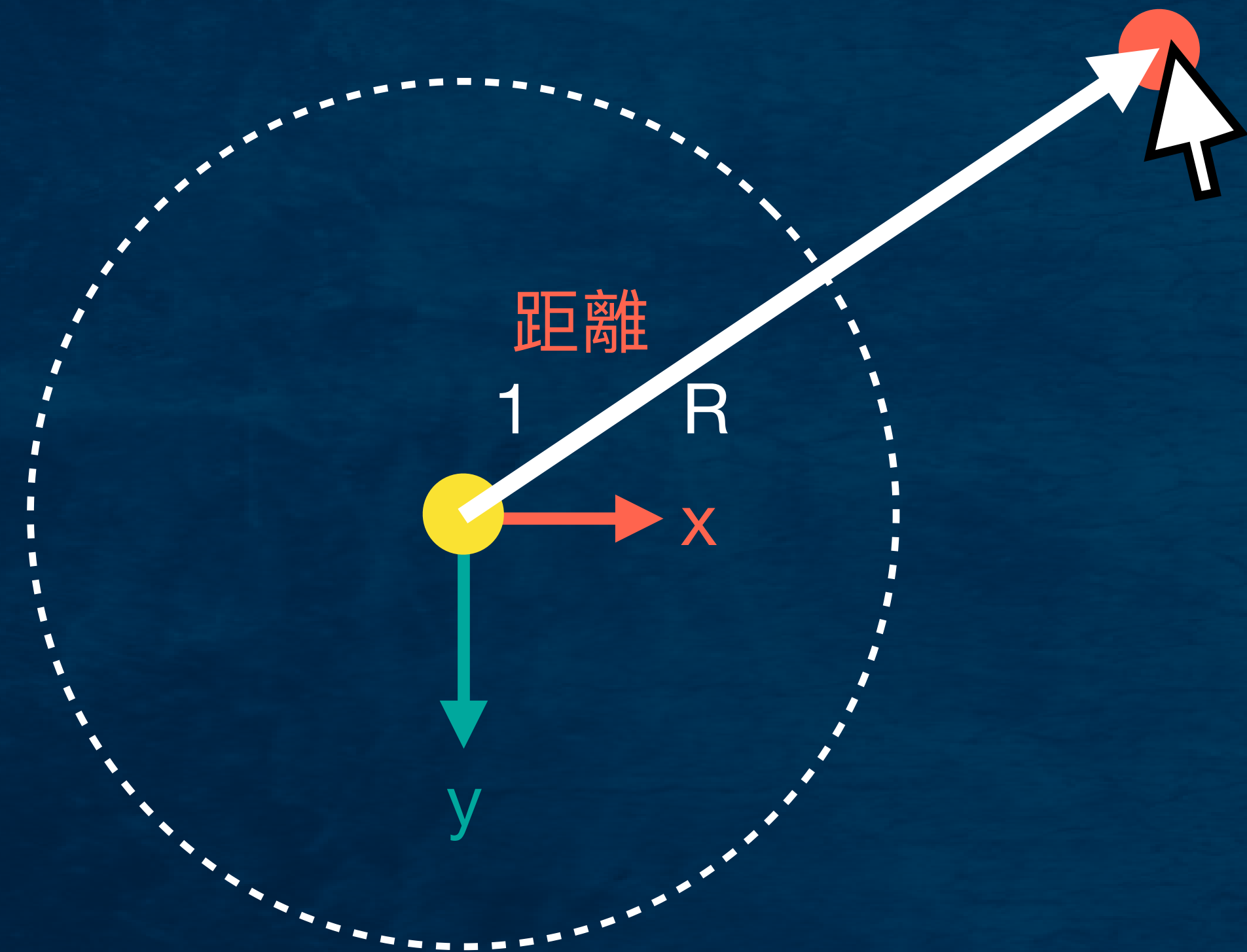
# 應用 - 朝向滑鼠畫固定長度

●  $\text{Cen} = \text{new Vector}(cx, cy)$

●  $\text{Mos} = \text{new Vector}(mx, my)$

➡  $\text{Delta} = \text{Mos.sub}(\text{Cen})$

- $\text{Point} = \text{Delta.mul}(1 / \text{Delta.length}) * R$
- $\text{Point} = \text{Delta.unit}() * R$



# 應用 - 串連與新增

- `A = new Vector(3,0)`
- `B = new Vector(4,4)`
- `C = A.sub( B ).mul( 0.6 ).length( )`

```
Vector.prototype = function ( ){  
    return this  
    return new Vector(x,y)  
}
```

使用回傳物件本身或新向量  
讓使用上可以串連  
//回傳值是向量





# 實作範例練習





課後練習：寫一個往滑鼠位置旋轉的箭頭