

第5課

向量與物件 (Vectors and Objects)

授課教師：黃啟賢

向量

- 二維向量

- 一個向量記錄了方向以及距離，它是一個把物體移動的過程。

在二維平面中

(a, b) 可以代表一個點

也可以代表一個向量

這個向量可以把物體從 $(0, 0)$ 移動到 (a, b)

所以這向量的長度（或是距離）為 $\sqrt{a^2 + b^2}$

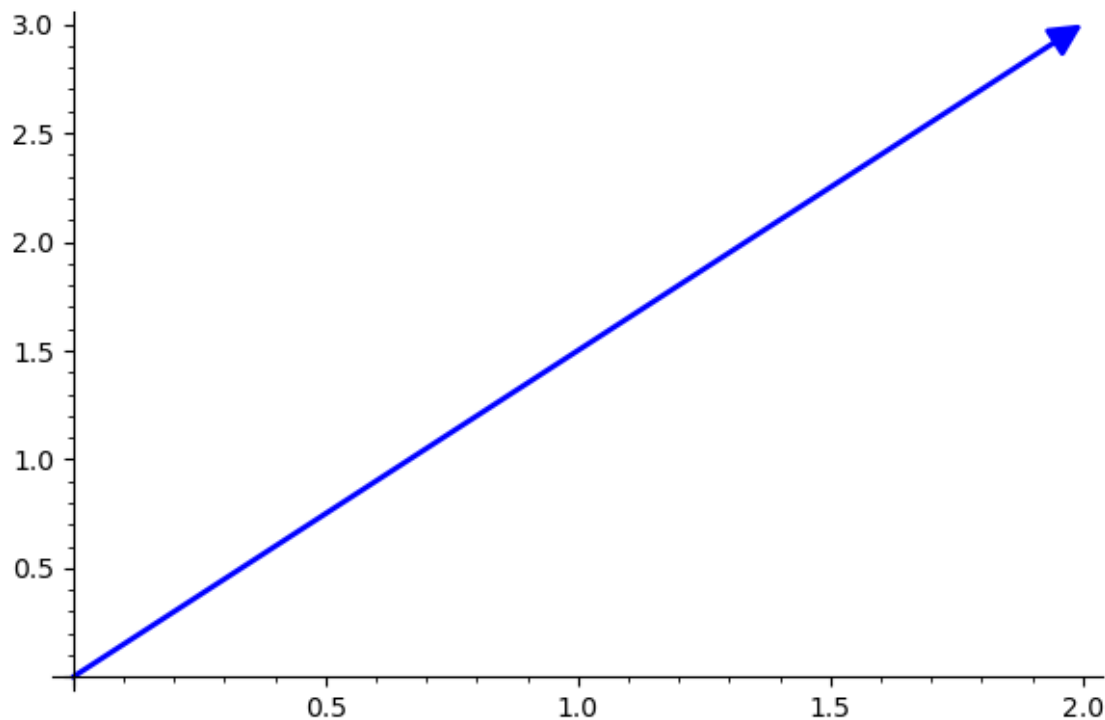
向量

在 Sage 裡
可以用 `vector([a,b])` 來建造一個向量 (a,b)

```
In [1]: v1 = vector([2,3])
```

```
In [2]: v1.plot()
```

Out[2]:



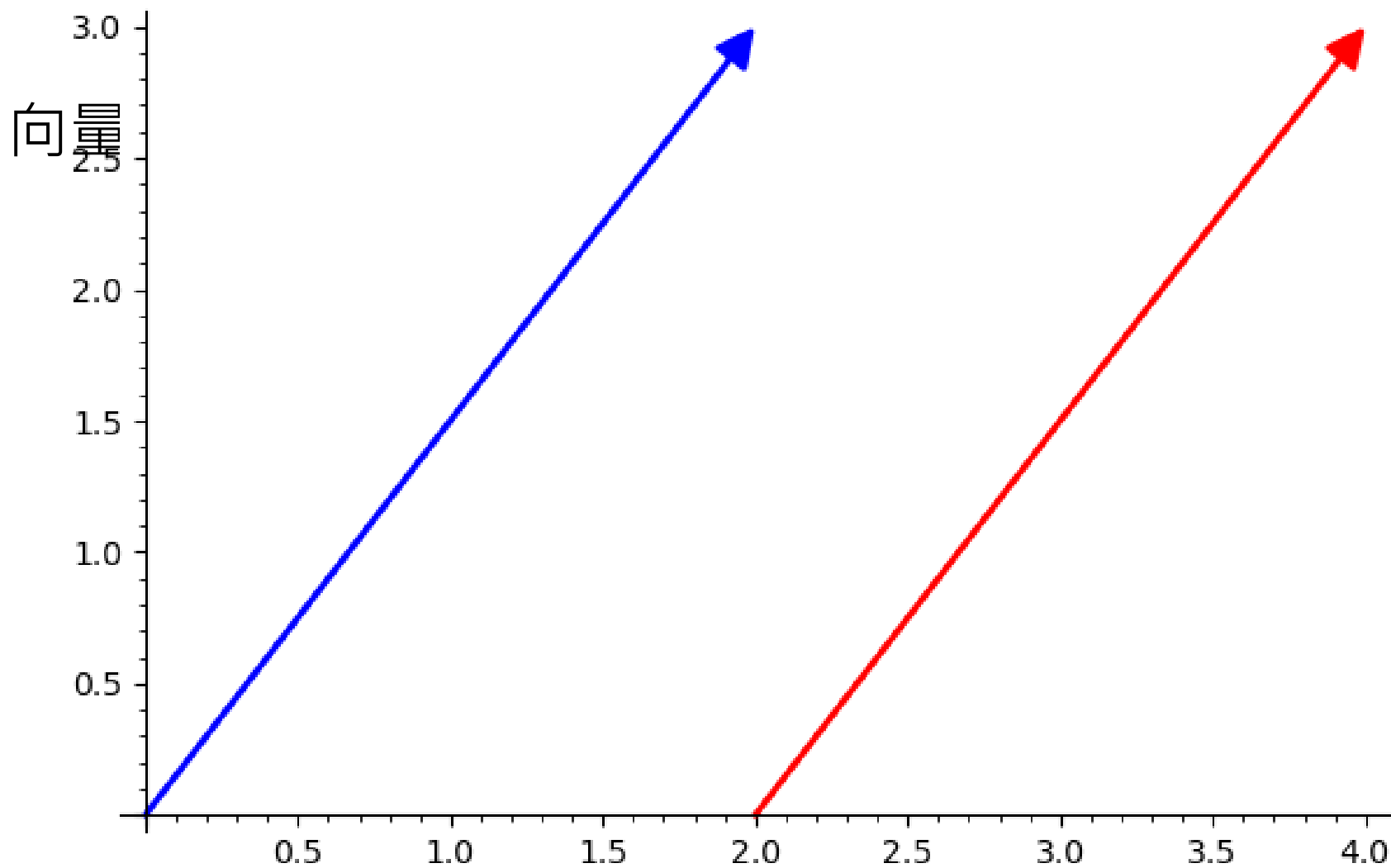
向量

- 我們只在意向量的方向及距離，並不在意它的起始點在哪邊。

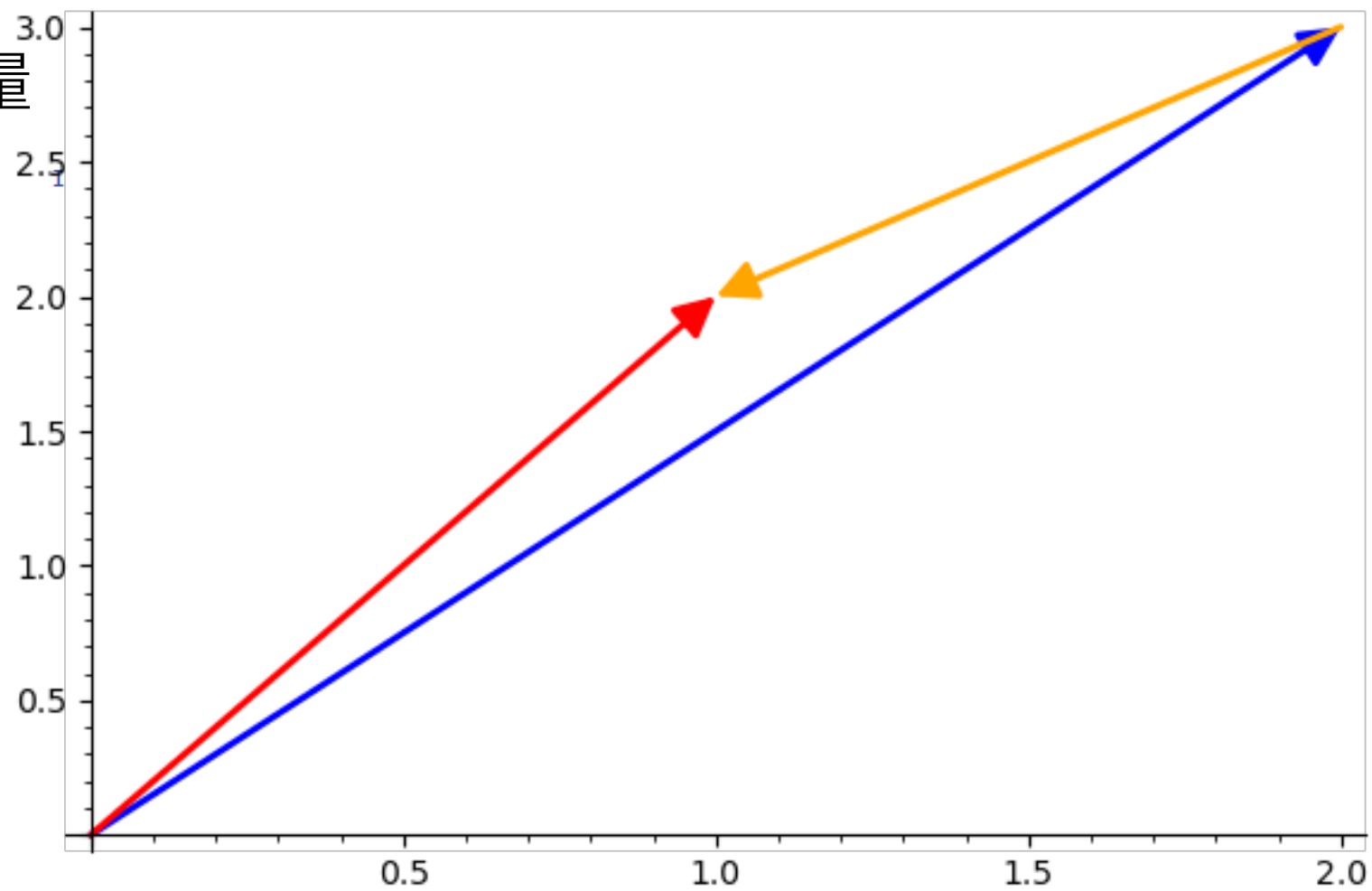
可以用 `start` 參數

來調整繪製時的起始點

```
In [3]: v1 = vector([2,3])  
pic1 = v1.plot(color='blue')  
pic2 = v1.plot(start=(2,0), color='red')  
pic1 + pic2
```



向量



向量

- 向量有**向量加法**：

向量加向量就是把對應的項相加，x 部分加總，y 部分加總。

$$(a, b) + (c, d) = (a + c, b + d)$$

```
In [4]: v1 = vector([2,3])  
        v2 = vector([-1,-1])  
        v3 = v1 + v2  
        v3
```

```
Out[4]: (1, 2)
```

向量

- 向量有純量乘法：
純量和向量相乘就是把純量乘進每一項。

$$k \times (a, b) = (ka, kb)$$

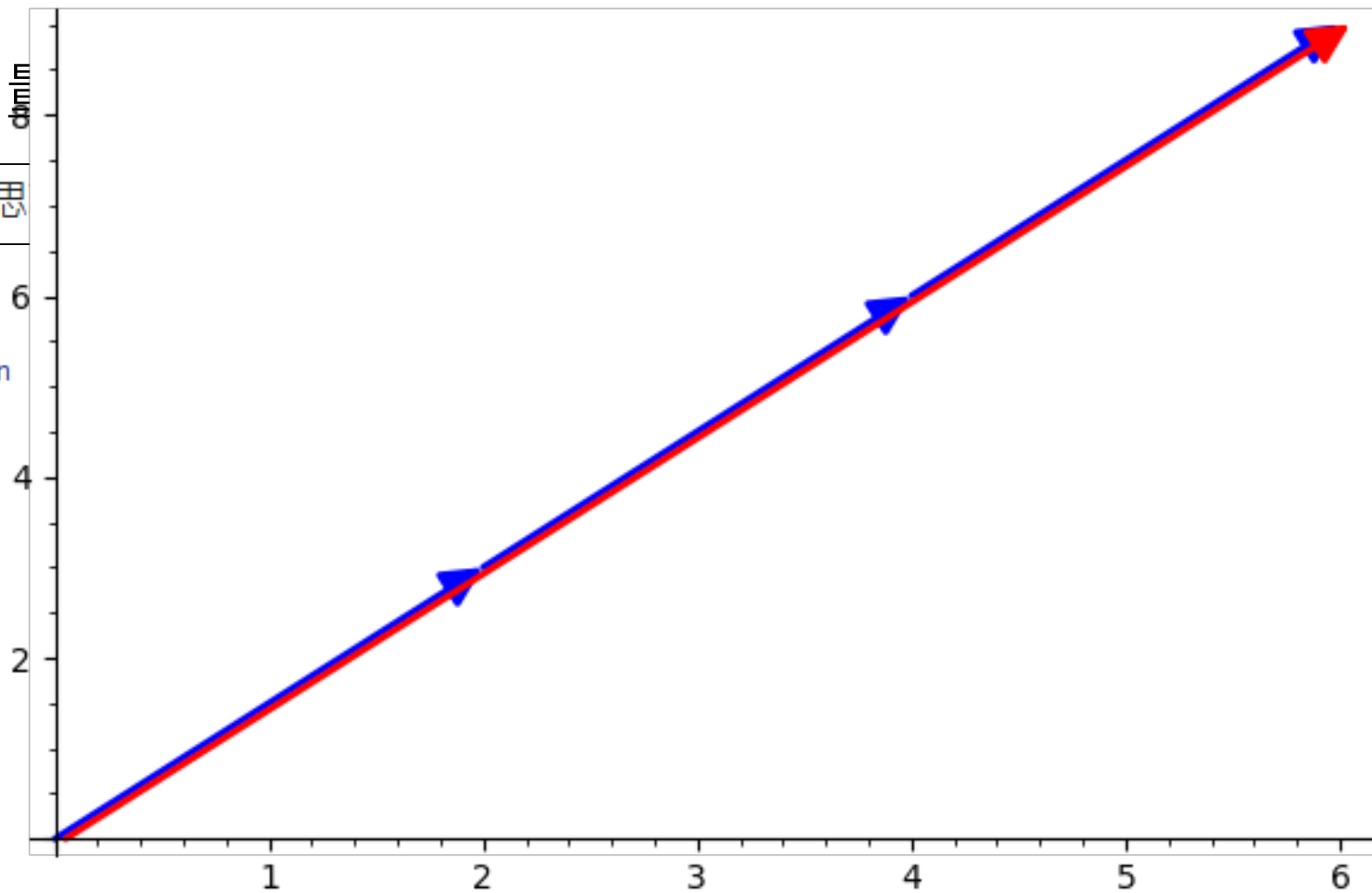
```
In [8]: v1 = vector([2,3])  
3 * v1
```

```
Out[8]: (6, 9)
```

向

意思

In



向量

- 向量和向量之間可以做內積 (dot product)
得到一個純量

$$(a, b) \cdot (c, d) = ac + bd$$

```
In [10]: v1 = vector([2,3])  
v2 = vector([-1,-1])  
v1 * v2
```

```
Out[10]: -5
```

高維度向量

一個 n 個數字的數對

$(a_1, \dots, a_n)$

叫作一個 n 維向量

```
In [11]: v1 = vector([1,2,3,4,5])
```

高維度向量

如同二維向量一般

兩個高維度的向量可以相加

$$(a_1, \dots, a_n) + (b_1, \dots, b_n) = (a_1 + b_1, \dots, a_n + b_n)$$

```
In [12]: v1 = vector([1,2,3,4,5])  
         v2 = vector([5,4,3,2,1])  
         v1 + v2
```

```
Out[12]: (6, 6, 6, 6, 6)
```

高維度向量

純量和向量可以做純量乘法

$$k \times (a_1, \dots, a_n) = (ka_1, \dots, ka_n)$$

```
In [13]: v1 = vector([1,2,3,4,5])  
3 * v1
```

```
Out[13]: (3, 6, 9, 12, 15)
```

高維度向量

- 向量和向量可以做內積
得到一個純量

$$(a_1, \dots, a_n) \cdot (b_1, \dots, b_n) = a_1 b_1 + \dots + a_n b_n$$

```
In [14]: v1 = vector([1,2,3,4,5])  
         v2 = vector([5,4,3,2,1])  
         v1 * v2
```

```
Out[14]: 35
```

第5-1課

Python 中的物件 (Objects of Python)

授課教師：黃啟賢

Python 中的物件

- 在 Python 裡
我們可以創造出全新的東西
然後定義它的許多性質及**相關函數**
甚至定義它們的加法
如何比大小
如何顯示出來
- 因為它是一個幾乎沒有侷限的東西，所以我們叫它作**物件**

Python 中的物件

- 自行定義一個類別

接下來

我們試著在 Sage 中

定義一個新的類別叫作 `fraction`

來實現數學中的分數

```
class fraction
```

定義了類別的名稱

```
__init__
```

定義了如何創造一個新物件

Python 中的物件

- 一個分數其實只須要兩個輸入的值：
分子 (numerator) 和分母 (denominator)

每個物件都可以設定許多屬性 (attribute)

像這裡我們設定了 `numerator` 、 `denominator` 、以及 `gcd`

當 `q` 是一個物件

`gcd` 是它的屬性

可以用 `q.gcd` 叫出這個屬性的值

```
In [19]: class fraction:
          def __init__(self,a,b):
              self.numerator = a
              self.denominator = b
              self.gcd = gcd(a,b)
```

```
q = fraction(4,6)
q.gcd
```

(greatest common divisor, gcd → 最大公因數)

Out[19]: 2

Python 中的物件

```
class fraction:
    def __init__(self,a,b):
        self.numerator = a
        self.denominator = b
        self.gcd = gcd(a,b)

    def reciprocal(self):
        return fraction(self.denominator,self.numerator)

q = fraction(2,3)
print('q')
print(q.numerator)
print(q.denominator)
print('---')

p = q.reciprocal();
print('1/q')
print(p.numerator)
print(p.denominator)
```

```
q
2
3
---
1/q
3
2
```

學習資源

- 線上課程
 - [A Tour of Sage](#)
 - [Official Sage Tutorials](#)
 - [Modular Sage](#)