

$$(X, O) = e^{-\frac{x^2}{2\sigma^2}}$$

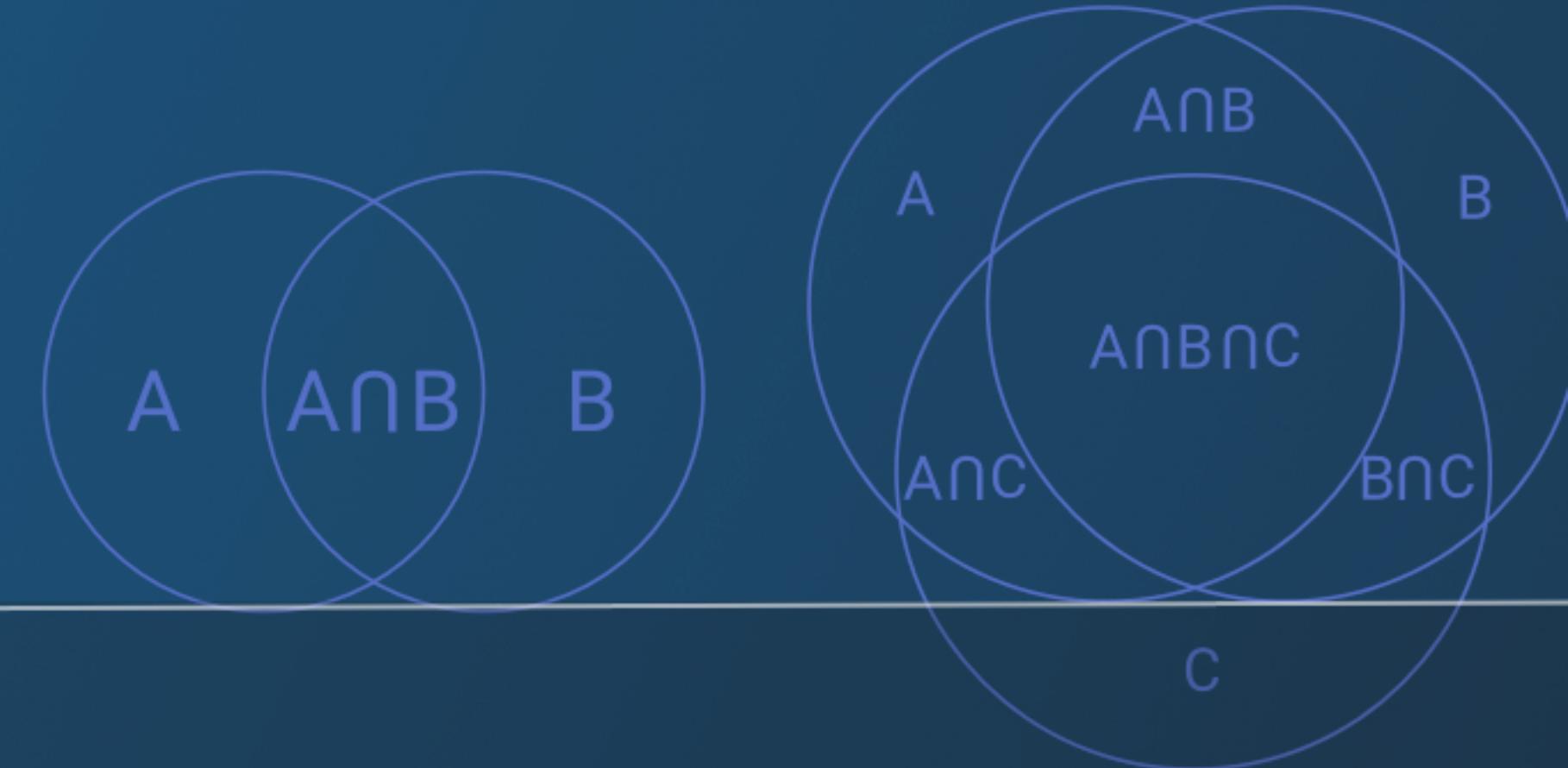
$$x(X, O) = -\frac{x}{\sigma^2} G(X, O) = -\frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}}$$

$$xx(X, O) = \frac{x^2 - \sigma^2}{\sigma^4} G(X, O) = \frac{x^2 - \sigma^2}{\sigma^4} e^{-\frac{x^2}{2\sigma^2}}$$

$$xxx(X, O) = -\frac{x^3 - x\sigma^2}{\sigma^6} G(X, O) = -\frac{x^3 - x\sigma^2}{\sigma^6} e^{-\frac{x^2}{2\sigma^2}}$$

# Julia 程式語言學習馬拉松

Day 25



$$\ln(x + \sqrt{1+x^2}) + x - \frac{1}{x + \sqrt{1+x^2}} \left( 1 + \frac{x}{\sqrt{1+x^2}} \right)$$

$$\ln(x + \sqrt{1+x^2}) + x - \frac{1}{x + \sqrt{1+x^2}} \left( \frac{\sqrt{1+x^2} + x}{\sqrt{1+x^2}} \right)$$



cupay 陪跑專家 : Andy Tu

# (程式安裝) PyCall 及 pyjulia 安裝





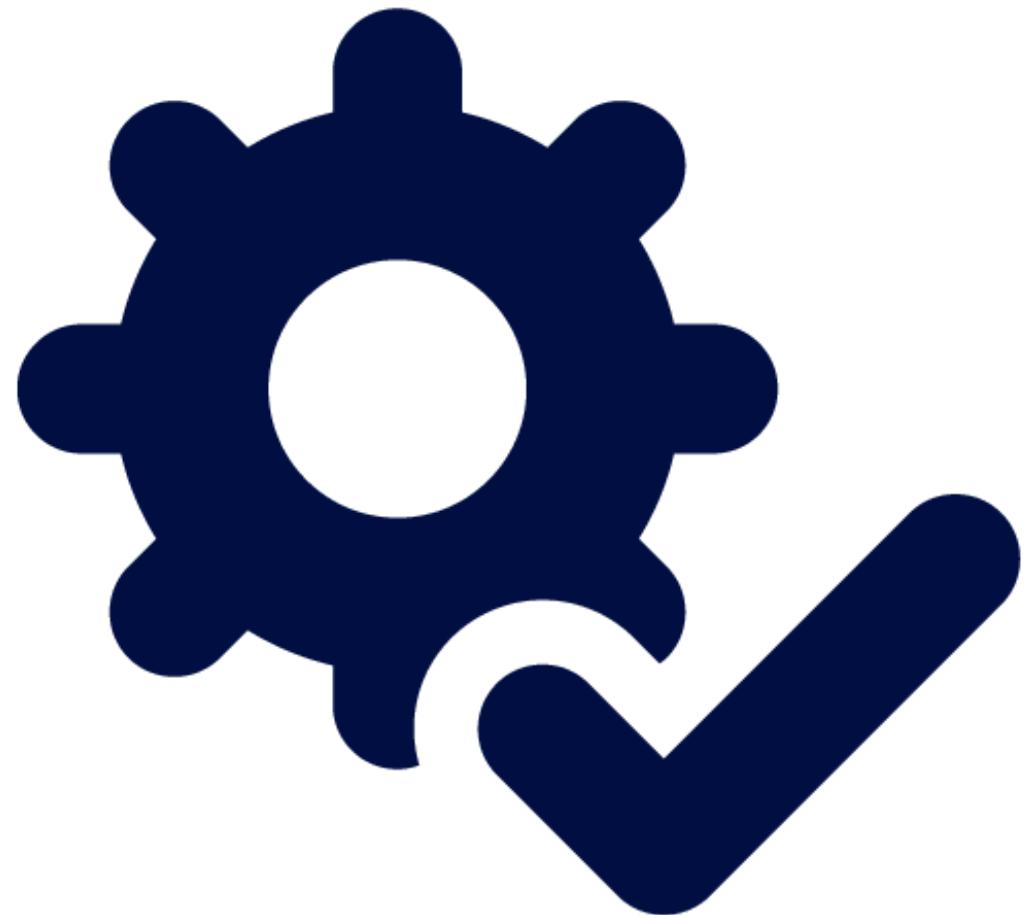
# 安裝 PyCall.jl



- 進入 julia REPL，進入套件模式（按下 `]`）
- `>update`
- `>add PyCall`



# 安裝 pyjulia



- 需要事先安裝 julia
  - pip install julia
  - python> import julia
  - python> julia.install()
- 最後一個步驟會在 julia 上安裝 PyCall，它是使用 pyjulia 所需要的套件。



# 安裝 pyjulia



- 注意
  - julia執行檔必須在 path 下，否則會無法執行 `julia.install()`
  - 請務必先在 Julia 中安裝 PyCall。如果尚未安裝 PyCall, 會先顯示錯誤訊息, 但隨後會自動安裝 PyCall。



## 推薦閱讀

- [PyCall.jl 官方文件](#)
- [PyJulia 官方文件](#)



# 程式語言外部呼叫 在 Julia 內使用 Python





# 重要知識點



- 認識 PyCall.jl
- 了解如何透過 PyCall 呼叫及使用 Python 套件
- 了解在 Julia 與 Python 中不同物件的使用方式



# PyCall.jl



- PyCall 套件提供了 Julia 與 Python 之間的介面，讓我們能夠在 Julia 內部去利用 Python，包含：
  - A. 呼叫 Python 函數與套件
  - B. 從 Julia 去操作 Python



# 初次嘗試



- 我們使用 PyCall 需要將 python 程式碼放到字串中。
- 程式碼字串前綴的 py 是個 macro，它會包裹整個程式碼並且將程式碼送至 python 執行。

```
py"1 + 2"  
py"type(1 + 2)"
```



# 使用 Python

- 在 python 程式碼中可以使用已安裝的 python 套件。
- 以下是將整段程式碼包裹到字串中，執行完成並不會回傳執行結果。

```
py"""
import numpy as np
np.sin(np.pi)
"""
```



# 使用 Python 套件



- 這邊介紹如何將 julia 物件送到 python 計算，我們可以在 julia 中使用 python 的函數進行運算。
- `py"np.sin"` 對應 python 的函式或物件，可以試著將 julia 物件作為參數傳遞給它進行運算。

```
py"np.sin"(1.0)
py"np.sin"(1//2)
py"np.sin"([0.5π, π, 2π])
```



# 使用 Python 套件



- pyimport 可以將 python 套件載入，並對應到一個 julia 的物件。
- 使用 pyimport 可以更方便在 julia 使用 python 的套件。
- np 會直接對應 python 的套件，並直接存取底下的功能。

```
np = pyimport("numpy")
np.sin(1)
```



# 使用 Python 的 With



- 資源管理器 with 在 python 中使用非常頻繁，我們一樣可以在 julia 中使用，@pywith 是其對應的語法。
- pybuiltin 則是呼叫 python 的內建語法，這邊我們用到 python 的 open。

```
@pywith pybuiltin("open")("file.txt","w")
as f begin
    f.write("hello")
end
```



# 呼吸 Python 的物件



- 如同前面我們做的，如果計算結果儲存在 python 物件中，我們可以直接呼叫 python 物件來取得其中的值。

```
julia> py"""\nimport numpy as np\ni = np.sin(np.pi)\n"""\n\njulia> py"i"\n1.2246467991473532e-16
```



# 內插 Julia 物件在 Python 程式碼中



- 如果希望將 julia 物件內插到 python 程式碼中，我們用 \$ 來內插這些變數。

```
julia> a = 1;  
julia> b = 2;  
julia> py"$a + $b"  
3
```

# 知識點 回顧

- 認識 PyCall.jl
- 了解如何透過 PyCall 呼叫及使用 Python 套件
- 使用 pyimport
- 呼叫 python 物件
- 將 julia 物件內插到 python 程式碼中



## 推薦閱讀

- [PyCall.jl 官方文件](#)





解題時間

請跳出 PDF 至官網 Sample Code  
& 作業開始解題