第一次作品：Python 函數繪製的觀念與技巧

學號：411073088

姓名：陳敬翰

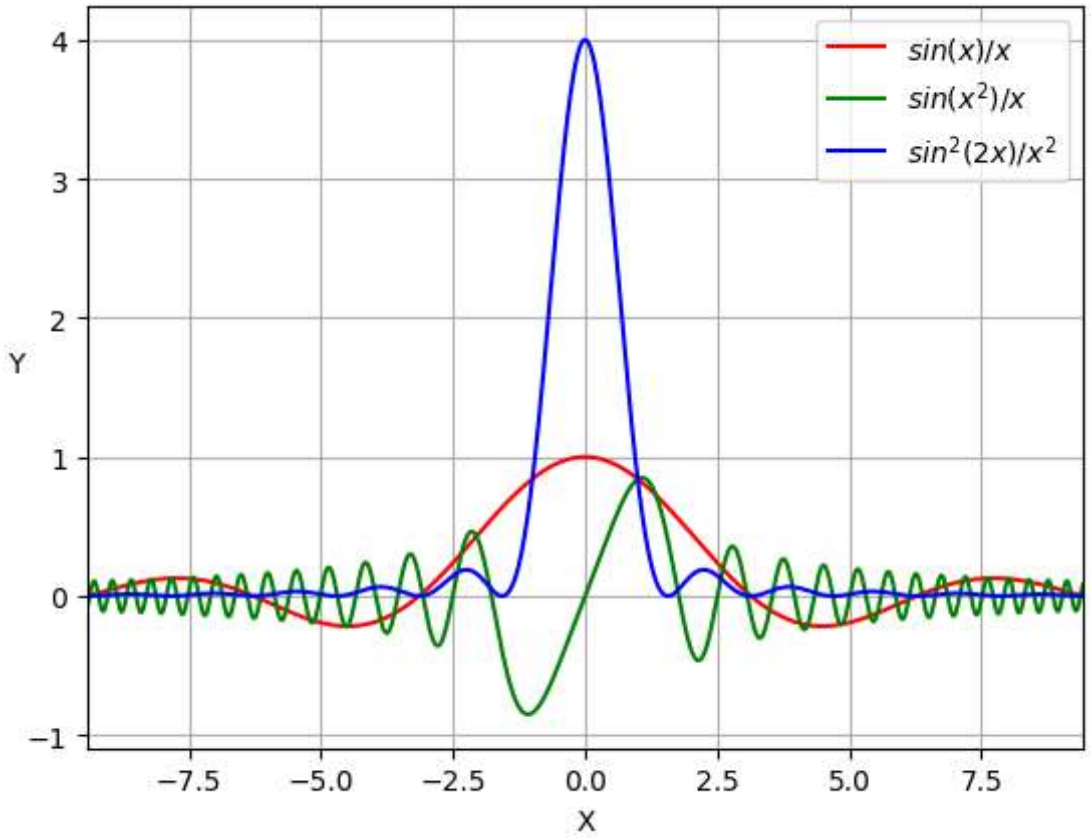
作品目標：本作品藉著繪製 11 個函數（如網站講義所列），學習 Python 的繪圖觀念與指令的運用，並透過函數繪製的過程，更深入了解每個函數的特色與精彩之處，譬如，函數的極值位置（local extrema and (or) global extrema）、函數是否通過 $y=0$（即 $f(x)=0$ 是否有實數根）、是否有漸進線？函數的範圍... 等。

1. 繪製函數

$$f(x)=\frac{\sin(x)}{x}, g(x)=\frac{\sin(x^2)}{x}, h(x)=\frac{\sin^2(2x)}{x^2}$$

$$-3\pi \le x \le 3\pi$$

In [ ]:
```
import matplotlib.pyplot as plt
import numpy as np

n = 1000
x = np.linspace(-3*np.pi, 3*np.pi, n)
f = np.sin(x) / x
g = np.sin(x**2) / x
h = np.sin(2*x)**2 / (x**2)

fig, ax = plt.subplots(1)

ax.plot(x, f, color="r", label="$sin(x) / x$")
ax.plot(x, g, color="g", label="$sin(x^2) / x$")
ax.plot(x, h, color="b", label="$sin^2(2x) / x^2$")
ax.grid(True)
ax.set_xlabel('X')
ax.set_ylabel('Y', rotation=0)
# ax.set_title(
#     '$f(x) = \\frac{\\sin(x)}{x}, g(x) = \\frac{\\sin(x^{2})}{x}, h(x) = \\frac{\
ax.set_xlim(-3*np.pi, 3*np.pi)
plt.legend()
plt.show()
```



注意事項與討論：

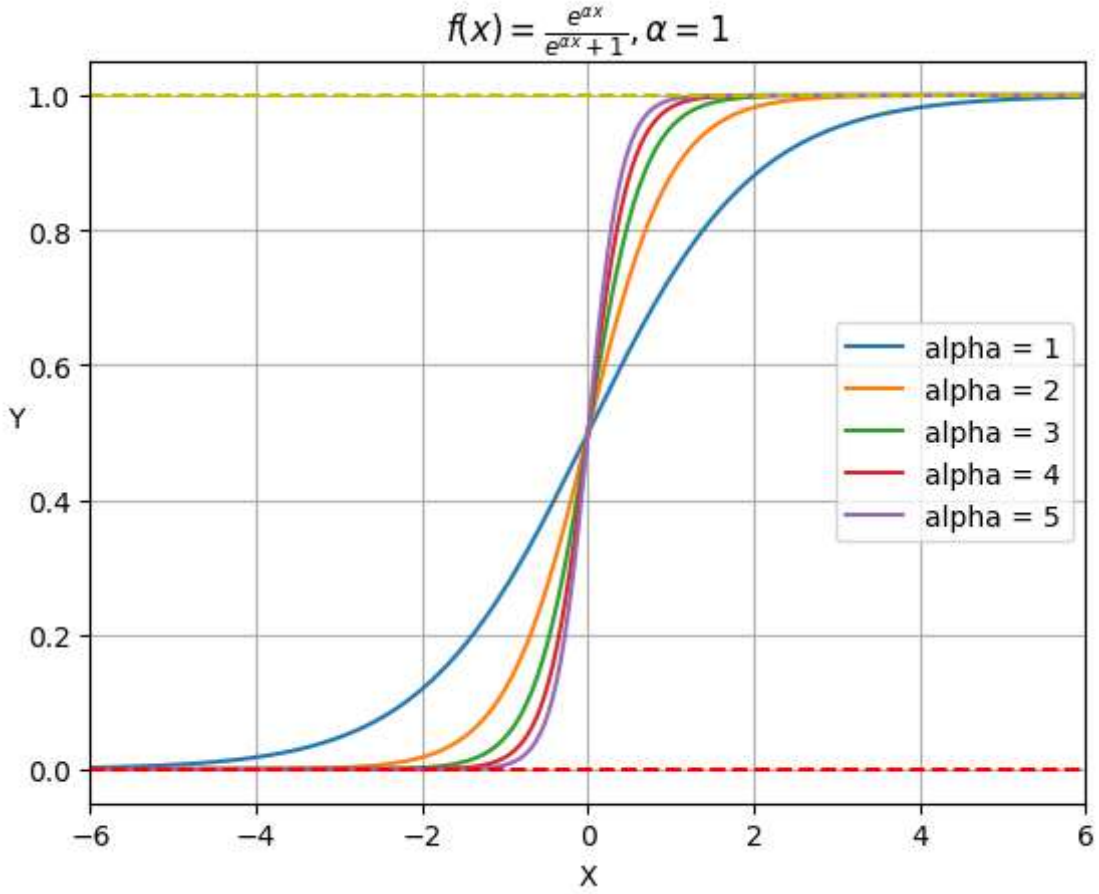
- $f(x) = \frac{\sin(x)}{x}$ 這個圖形的最高點會出現在x趨近於0的時候
- $g(x) = \frac{\sin(x^2)}{x}$ 這個圖形的最高點會出現在x趨近於0的時候
- $h(x) = \frac{\sin^2(2x)}{x^2}$ 這個圖形的最高點會出現在x趨近於0的時候

---

1. 繪製函數

$$f(x) = \frac{e^{\alpha x}}{e^{\alpha x} + 1}, \alpha = 1$$

In [ ]:
```
import numpy as np
import matplotlib.pyplot as plt

n = 2000
x = np.linspace(-50, 50, n)
fig, ax = plt.subplots(1)
for alpha in range(1,6):
    f = np.exp(alpha * x) / (np.exp(alpha * x) + 1)
    ax.plot(x, f,   label=f"alpha = {alpha}")
ax.grid(True)
ax.set_xlabel('X')
ax.set_ylabel('Y', rotation=0)
ax.axhline(y=1, color="y", linestyle="--")
ax.axhline(y=0, color="r", linestyle="--")
ax.set_xlim(-6, 6)
ax.set_title('$f(x) = \\frac{e^{\\alpha x}}{e^{\\alpha x} + 1}, \\alpha = 1$')
ax.legend()
plt.show()
```



注意事項與討論：

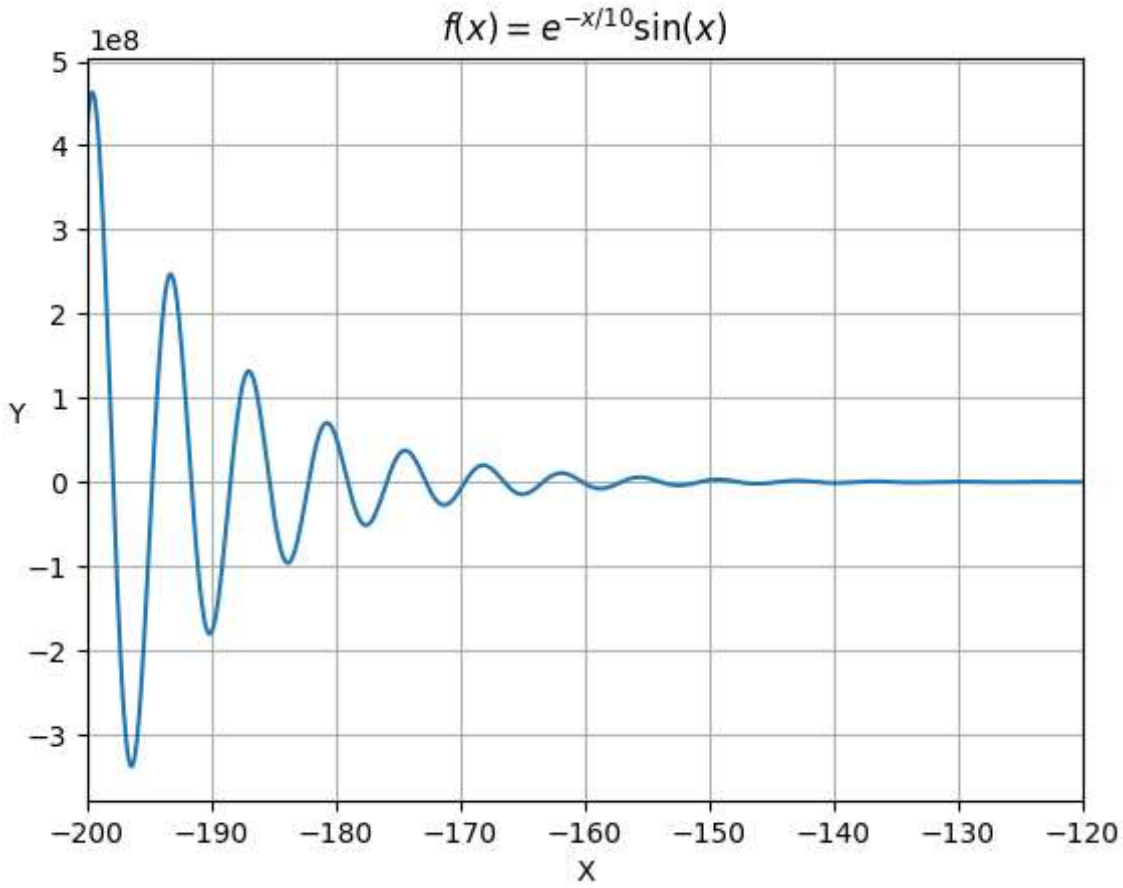
- 當alpha越來越大時，f(x)上升的速度就會越來越快，最後分別會趨近於0及1

---

1. 繪製函數

$$f(x) = e^{-x/10}\sin(x)$$

In [ ]:
```
import numpy as np
import matplotlib.pyplot as plt

n = 3000
x = np.linspace(-200,200,n)
f = (np.exp(-x/10))*(np.sin(x))

plt.plot(x,f)
plt.grid(True)
plt.xlabel('X')
plt.ylabel('Y',rotation=0)
plt.title("$f(x)=e^{-x/10} \\sin(x)$")
plt.xlim(-200,-120)
plt.show()
```



注意事項與討論：

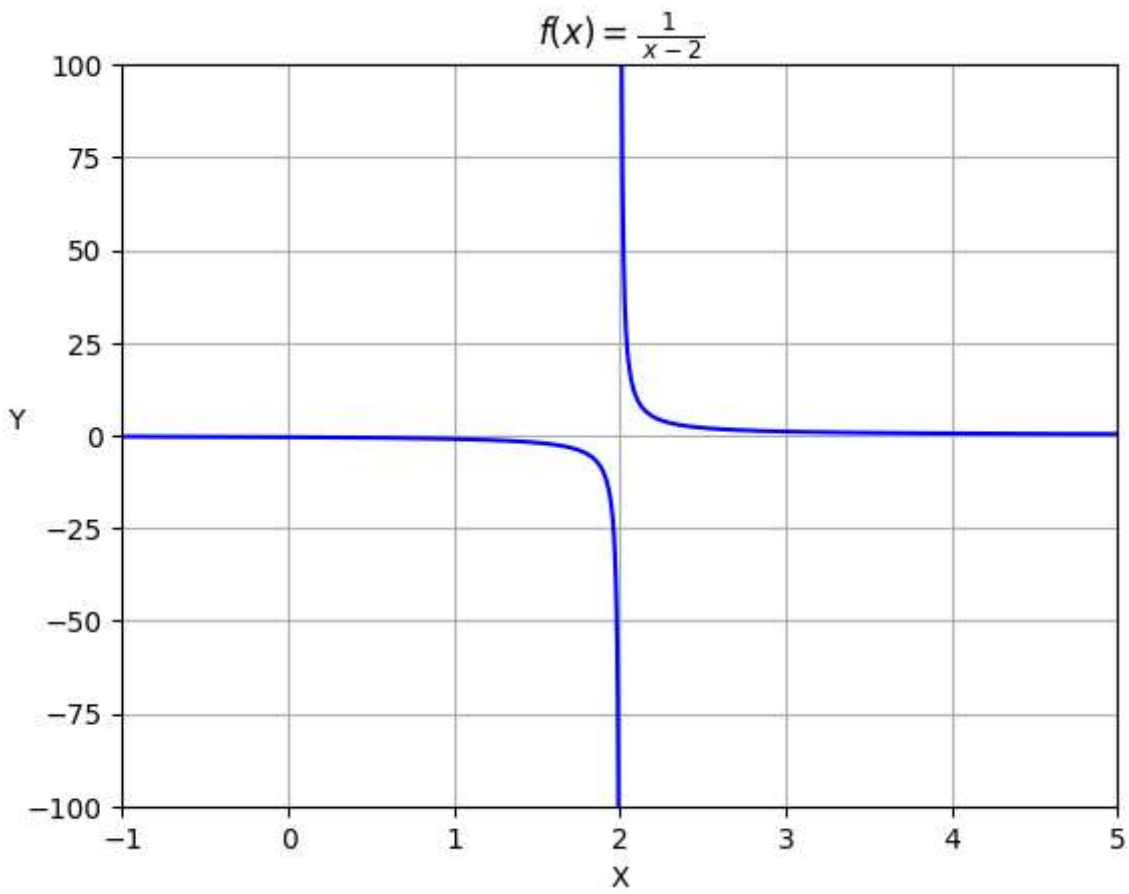
- 圖形在 x < -100 時呈現大幅度波動，當x逐漸變大時，f(x)會趨近於0

---

1. 繪製函數

$$f(x) = \frac{1}{x-2}$$

In [ ]:
```
import numpy as np
import matplotlib.pyplot as plt

n = 2000
x = np.setdiff1d(np.linspace(-5, 2, n),[2])
y = 1 / (x-2)
plt.plot(x,y,color='blue')

x = np.setdiff1d(np.linspace(2, 9, n),[2])
y = 1 / (x-2)
plt.plot(x,y,color='blue')

plt.grid(True)
plt.xlabel("X")
plt.ylabel("Y",rotation=0)
plt.title("$f(x)=\\frac{1}{\\ x-2}$")
plt.ylim(-100,100)
plt.xlim(-1,5)
plt.xticks([-1,0,1,2,3,4,5])
plt.show()
```



注意事項與討論：

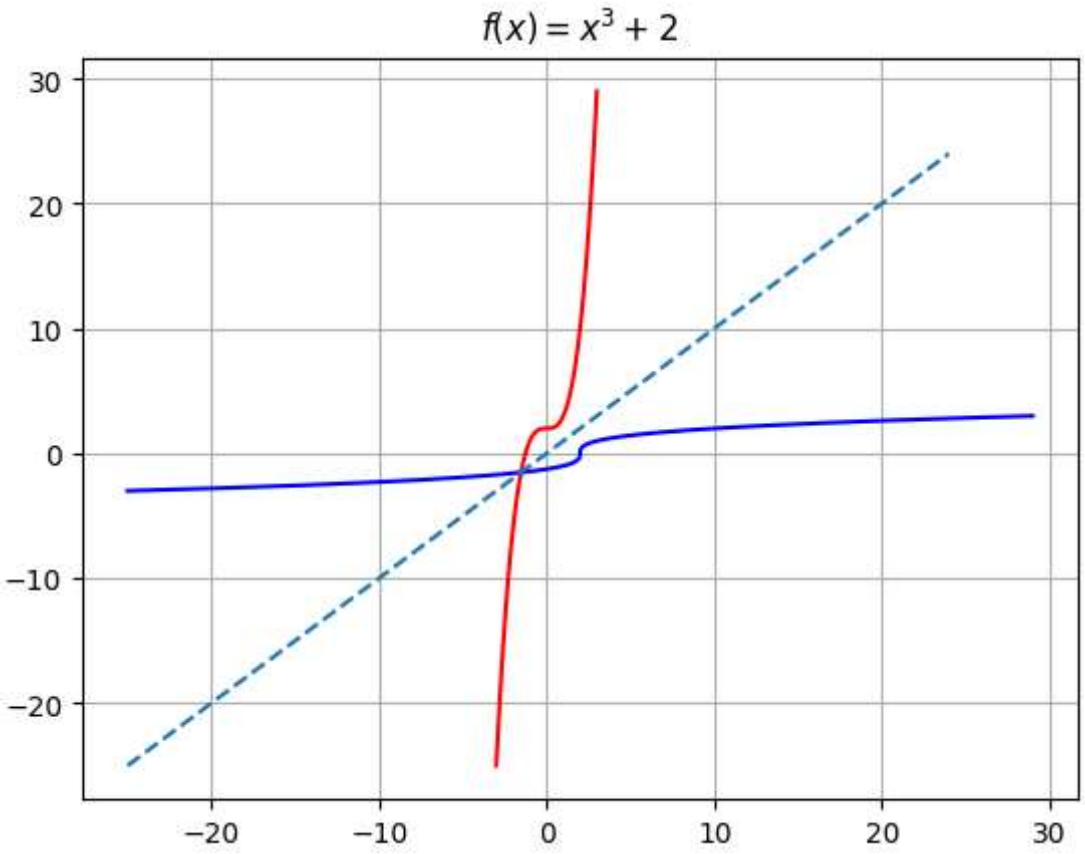
- 圖形以x=2為界，分成正負兩段，兩段最後都會趨近於0，負的那段在x接近2時會遞減，正的那段則相反
- 在x=2時，f(x)會無限大

---

1. 繪製函數

$$f(x) = x^3 + 2$$

In [ ]:
```
import numpy as np
import matplotlib.pyplot as plt

n = 200
x = np.linspace(-3,3,n)
f = x**3 + 2

plt.grid(True)
plt.plot(x,f,label='$f(x) = x^3 + 2$',color="r")
plt.plot(f,x,label='$f^(-1)(x^3) + 2',color="b")
plt.plot([i for i in range(-25,25)],[i for i in range(-25,25)], linestyle='--')
plt.title("$f(x)= x^3 + 2$")
plt.show()
```



注意事項與討論：

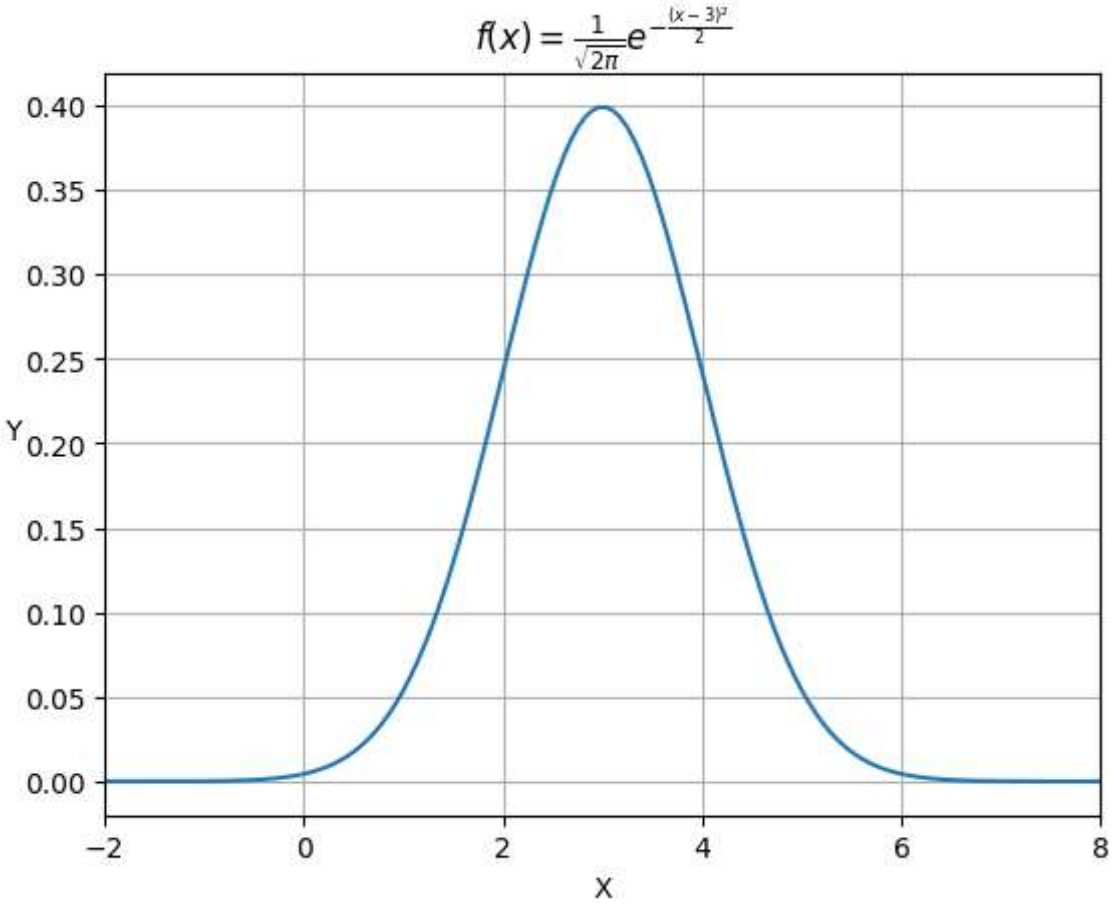
- 該圖形有一個實數解及兩個虛數解，反函數以x=y作為對稱軸
- 圖形在x趨近於無限大時，會趨近於無限大
- 圖形在x趨近於無限小時，會趨近於2
- 圖形在x=0時，會趨近於2

---

1. 繪製函數

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-3)^2}{2}}$$

In [ ]:
```
import numpy as np
import matplotlib.pyplot as plt

n = 200
x = np.linspace(-4,10,n)
f = (1/np.sqrt(2 * np.pi))*(np.exp(-(x-3)**2/2))
plt.grid(True)
plt.plot(x,f)
plt.xlabel("X")
plt.ylabel("Y",rotation=360)
plt.title("$f(x) = \\frac{1}{\\sqrt{2 \\pi}} e^{-\\frac{\\left(x-3\\right)^{2}}{2}}
plt.xlim(-2,8)
plt.show()
```



注意事項與討論：

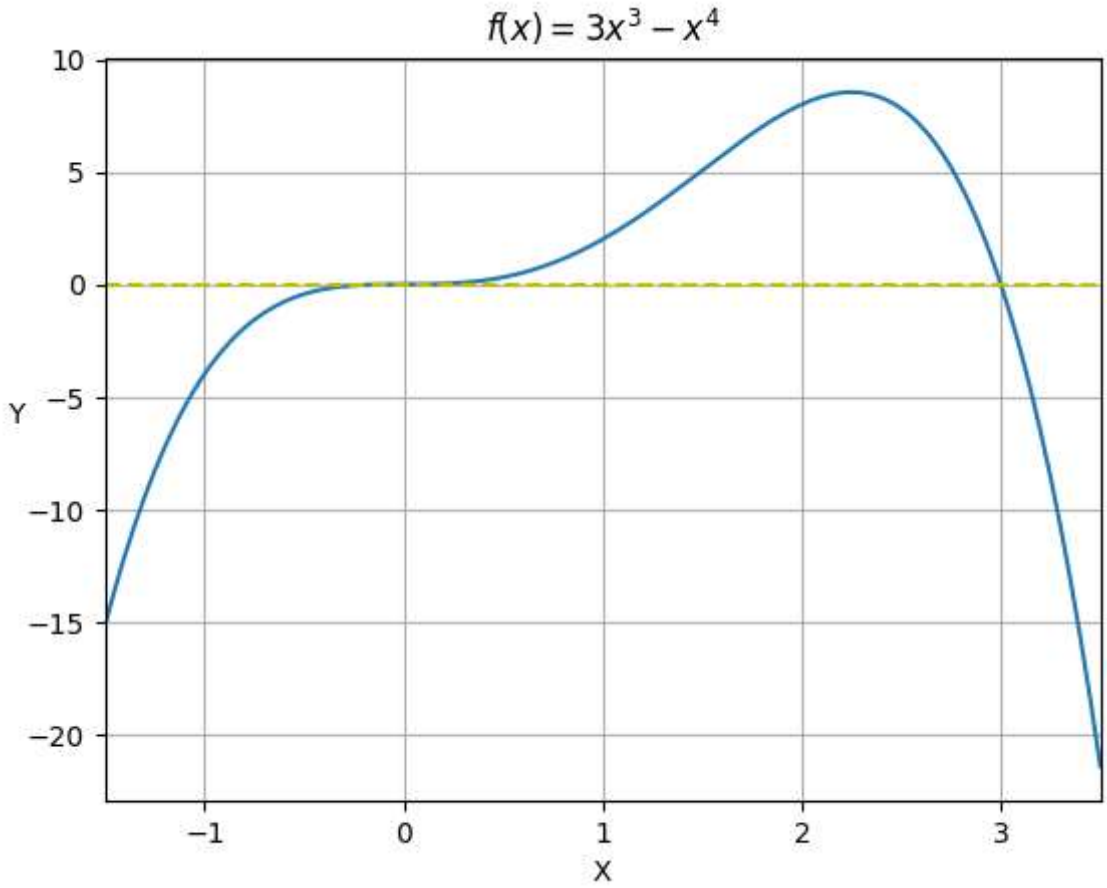
- 該圖形以x=3為中心，左右對稱
- 該圖形在x=3時，會趨近於0

---

1. 繪製函數

$$f(x) = 3x^3 - x^4$$

In [ ]:
```
import numpy as np
import matplotlib.pyplot as plt

n = 200
x = np.linspace(-1.5,3.5,n) #區間為X=-2, X=4
f = 3*(x**3) - x**4


plt.grid(True)
plt.plot(x,f)
plt.xlabel("X")
plt.ylabel("Y",rotation=360)
plt.axhline(y=0,color="y",linestyle="--")
plt.title("$f(x)=3x^3 - x^4$")
plt.xlim(-1.5,3.5)
plt.show()
```



注意事項與討論：

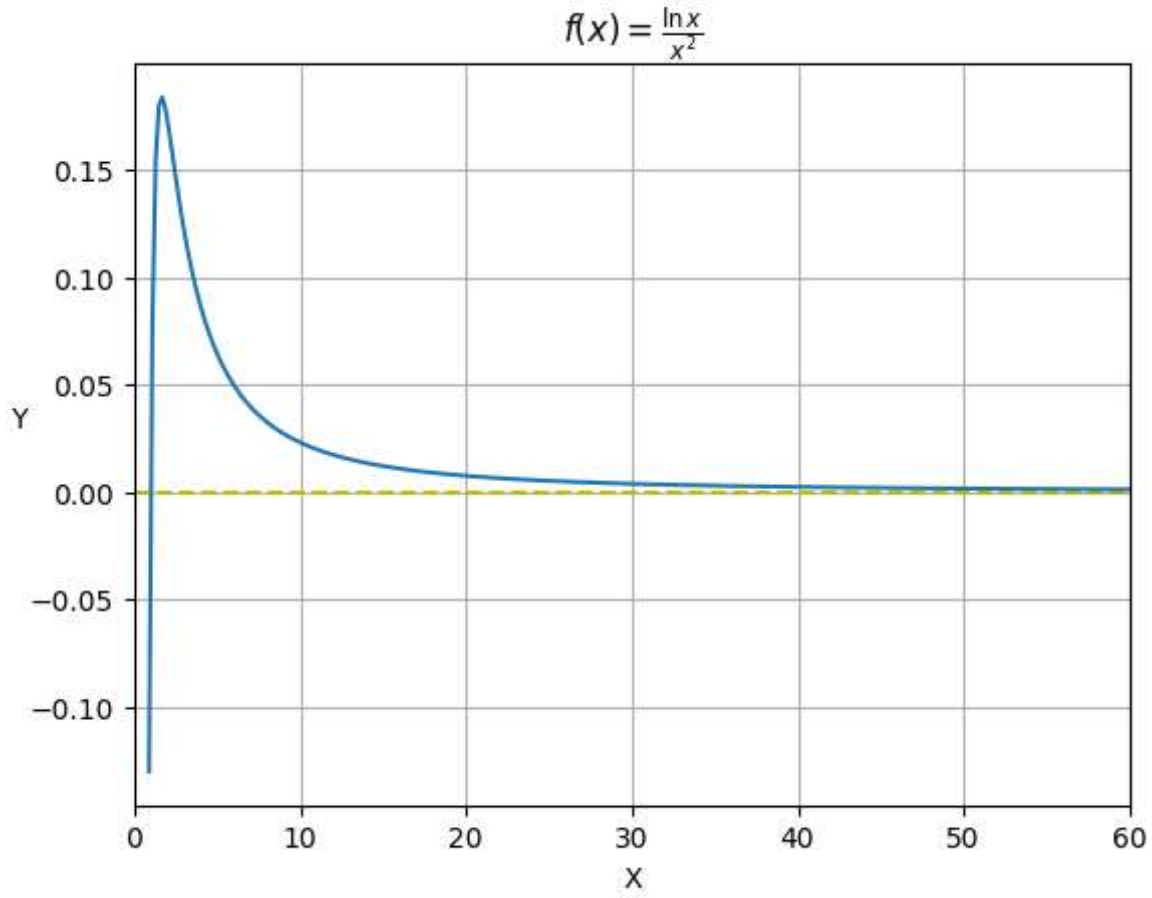
- 這個函數有兩個解，分別是x=0及x=3

---

1. 繪製函數

$$f(x) = \frac{\ln x}{x^2}$$

In [ ]:
```
import numpy as np
import matplotlib.pyplot as plt

n = 300
x = np.linspace(0.9,60,n)
f = np.log(x) / x**2

plt.grid(True)
plt.xlabel("X")
plt.ylabel("Y",rotation=360)
plt.axhline(y=0,color="y",linestyle="--")
plt.plot(x,f)
plt.title("$f(x) = \\frac{\\ln x}{\\ x^2}$")
plt.xlim(0,60)
plt.show()
```



注意事項與討論：

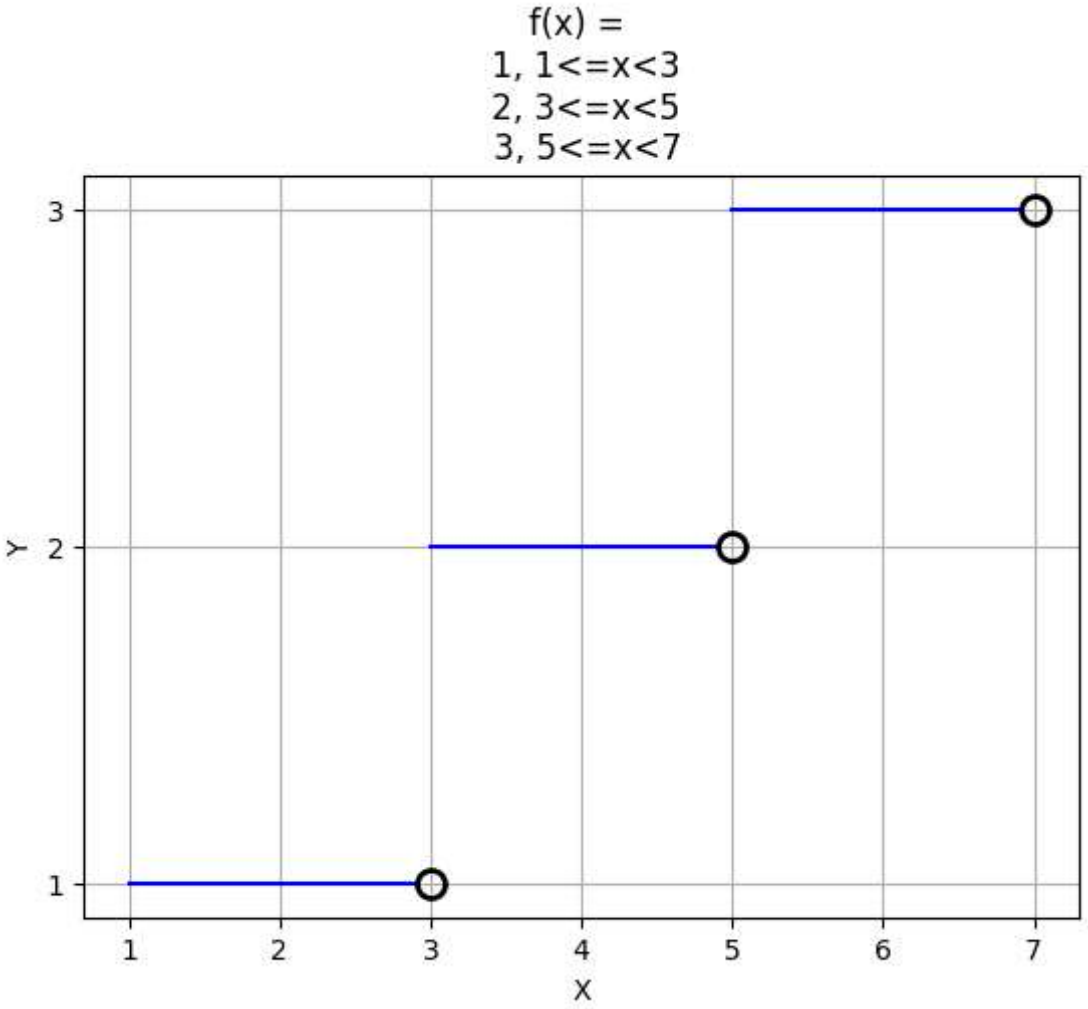
- 該圖形在x=0時，會趨近於負無限大
- 該圖形在x=無限時，會趨近於0

9.繪製函數

$$f(x) = \begin{cases} 1 & \text{if } 1 \leq x < 3 \\ 2 & \text{if } 3 \leq x < 5 \\ 3 & \text{if } 5 \leq x < 7 \end{cases}$$

In [ ]:
```
import matplotlib.pyplot as plt
import numpy as np

# 创建 x 向量
x = np.linspace(1, 7, 1000)
# 计算 y
y = np.where(x < 3, 1, np.where(x < 5, 2, 3))

# 创建一个空白的圆圈标记，标记在 x=3 处
import matplotlib.pyplot as plt
import numpy as np

x = [1,2,2.9]
y = [1,1,1]
plt.plot(x, y, color='blue')
x = [3,4,4.9]
y = [2,2,2]
plt.plot(x, y, color='blue')
x = [5,6,6.9]
```

```
y = [3,3,3]
plt.plot(x, y,color='blue')

# 绘制一个有边框的空心圆圈标记，标记在 x=3 处
plt.plot([3], [1], 'wo', markerfacecolor='none', markersize=10, markeredgewidth=2,
plt.plot([5], [2], 'wo', markerfacecolor='none', markersize=10, markeredgewidth=2,
plt.plot([7], [3], 'wo', markerfacecolor='none', markersize=10, markeredgewidth=2,

# 设置 y 轴刻度值
plt.yticks([1, 2, 3])
plt.grid(True)
plt.xlabel('X')
plt.ylabel('Y')
plt.title("f(x) = \n 1, 1<=x<3\n 2, 3<=x<5\n 3, 5<=x<7")
plt.show()
```



注意事項與討論：

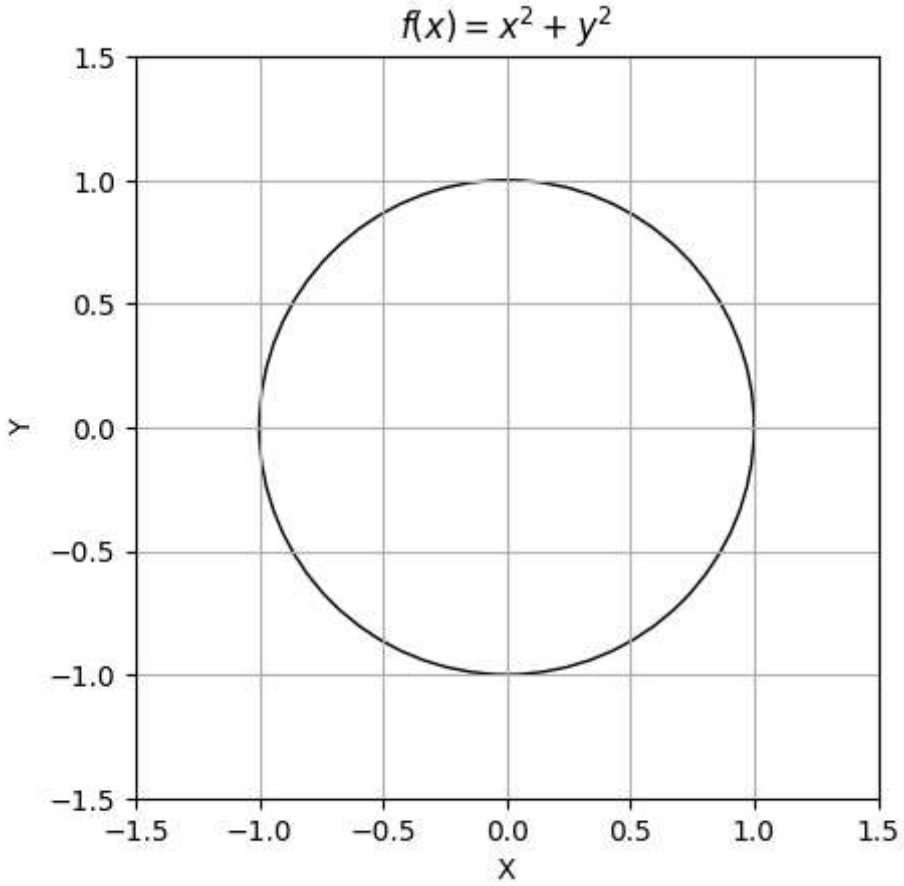
- 該圖形有三個區間，在x=1時，y=1，在x=3時，y=2，在x=5時，y=3

10.畫圓

$$f(x) = x^2 + y^2$$

In [ ]:
```
import matplotlib.pyplot as plt
import matplotlib.patches as patches

fig, ax = plt.subplots()

circle = patches.Circle((0, 0), radius=1, fill=False)
```

```
ax.add_patch(circle)

ax.set_xlim(-1.5, 1.5)
ax.set_ylim(-1.5, 1.5)

plt.xlabel('X')
plt.ylabel('Y')
plt.title('$f (x)=x^2+y^2$')

plt.gca().set_aspect('equal', adjustable='box')
plt.grid(True)
plt.show()
```



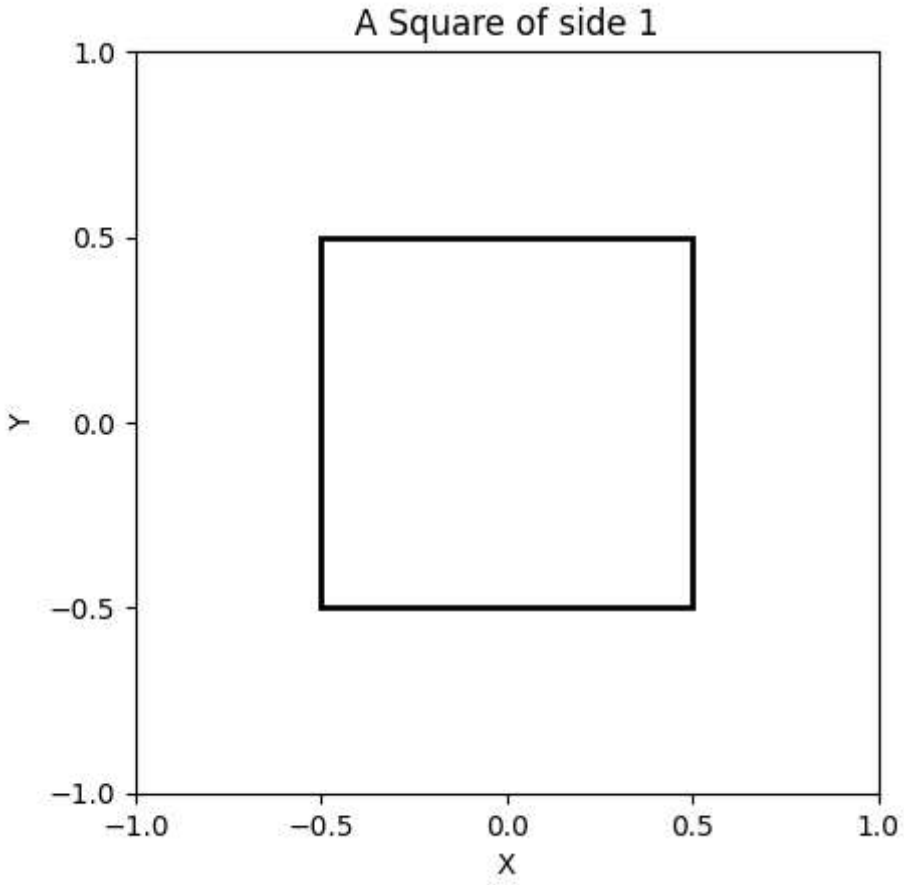
注意事項與討論：

- 該圖形以x=0為中心，y=0為中心，畫出一個圓

10.正方形

In [ ]:
```
import matplotlib.pyplot as plt
import matplotlib.patches as patches

fig, ax = plt.subplots()

square = patches.Rectangle((-0.5, -0.5), 1, 1, fill=False, edgecolor='black', linew

ax.add_patch(square)

ax.set_xlim(-0.7, 0.7)
ax.set_ylim(-0.7, 0.7)

plt.xticks([i * 0.5 for i in range(-2, 3)])  # x軸刻度
plt.yticks([i * 0.5 for i in range(-2, 3)])  # y軸刻度
```

```
plt.xlabel('X')
plt.ylabel('Y')
plt.title('A Square of side 1')

plt.gca().set_aspect('equal', adjustable='box')  # 使正方形看起来更正方形
plt.show()
```



注意事項與討論：

- 這個圖形是一個以x=0、y=0為中心，邊長為1的正方形