

實戰一

題目：

實戰 (1)

- 題目：設計 Python 程式，計算下列函數值
1. $\ln e^2$
 2. $\log_2(1024)$
 3. $\sin(\pi/6)$
 4. $\cos(\pi/6)$
 5. $\sin^{-1}(1)$
 6. $\cos^{-1}(1/2)$
- 執行結果

```
2.0
10.0
0.49999999999999994
-1.0
1.5707963267948966
1.0471975511965979
```

程式碼：

```
1 import numpy as np
2 from math import e
3 print(np.log(e ** 2))
4 print(np.log2(1024))
5 print(np.sin(np.pi/ 6))
6 print(np.cos(np.pi))
7 print(np.arcsin(1))
8 print(np.arccos(1/2))
```

執行結果：

```
2.0
10.0
0.49999999999999994
-1.0
1.5707963267948966
1.0471975511965979
```

說明:

第一題因需要用到 e 因此引進 `math` 函式庫來使用 e

而計算皆採用 `numpy` 內的數學公式來進行計算

實戰二

題目:

實戰 (2)

- 題目：設計 Python 程式，產生漸層的灰階影像
- 要求：灰階影像，影像大小為 512x100，灰階由左而右分別為 0~255，顯示影像結果
- 執行結果



程式碼:

```
1 import numpy as np
2 from google.colab.patches import cv2_imshow
3
4 img = np.zeros([100, 512], dtype = 'uint8')
5
6 gray = 0
7
8 for i in range(100):
9     for j in range(512):
10         img[i, j] = gray
11         if ((j + 1) % 2 == 0) & ((j + 1) > 0):
12             gray += 1
13
14 print(img)
15 cv2_imshow(img)
```

結果:

```
[[ 0  0  1 ... 254 255 255]
 [ 0  0  1 ... 254 255 255]
 [ 0  0  1 ... 254 255 255]
 ...
 [ 0  0  1 ... 254 255 255]
 [ 0  0  1 ... 254 255 255]
 [ 0  0  1 ... 254 255 255]]
```



說明:

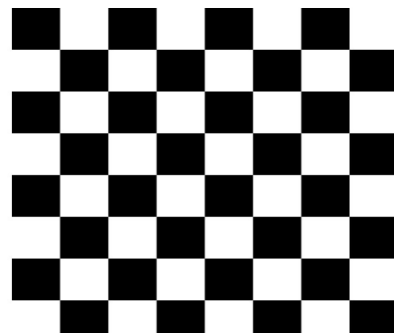
利用 **for** 迴圈來賦予每個像素點的灰階值，並透過 **if** 來將共 512 行的像素點分成 216 行，並賦予其灰階值

實戰二

題目:

實戰 (3)

- 題目：設計 Python 程式，產生 8x8 西洋棋盤
- 要求：灰階影像，影像大小為 512x512，灰階值僅包含 0 或 255，顯示結果並儲存影像檔
- 執行結果

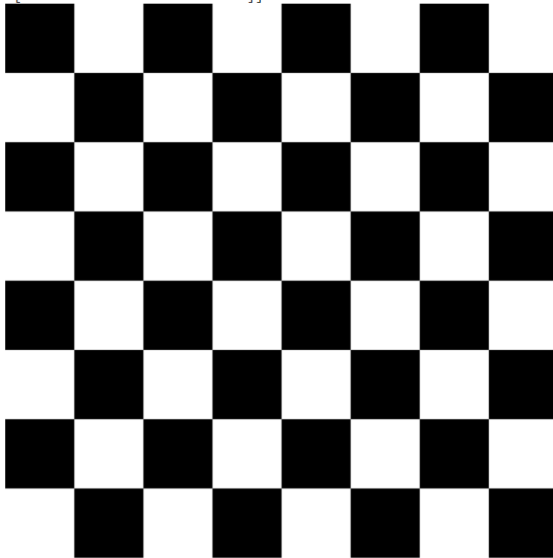


程式碼:

```
1 import numpy as np
2 import cv2
3 from google.colab import drive
4 from google.colab.patches import cv2_imshow
5
6 drive.mount('/content/drive')
7
8 imagePath = "/content/drive/My_Drive/Colab Notebooks/image_processing/"
9
10 img = np.zeros([512, 512], dtype = 'uint8')
11
12 for i in range(8):
13     for j in range(8):
14         if((i + j) % 2 == 0):
15             gray = 0
16         else:
17             gray = 255
18
19         for l in range(64):
20             for k in range(64):
21                 img[(i * 64) + l, (j * 64) + k] = gray
22
23 cv2.imwrite(imagePath+'chessboard.bmp', img)
24 print(img)
25 cv2_imshow(img)
```

結果:

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
[[ 0  0  0 ... 255 255 255]
 [ 0  0  0 ... 255 255 255]
 [ 0  0  0 ... 255 255 255]
 ...
 [255 255 255 ... 0 0 0]
 [255 255 255 ... 0 0 0]
 [255 255 255 ... 0 0 0]]
```



說明:

利用 for 迴圈將 512*512 的棋盤分成 8*8 塊，並

且得出了每塊皆有 $64*64$ 塊的像素點，因此再用
for 迴圈賦予每個 $64*64$ 的像素點灰階值，之後
將影像儲存至
Colab Notebooks/image_processing/ 資料夾內，
檔名為 chessboard