



UNIVERSITÀ  
di **VERONA**

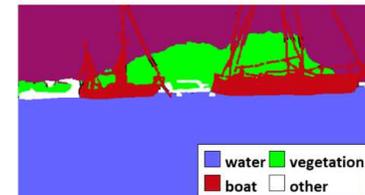
Dipartimento  
di **INFORMATICA**



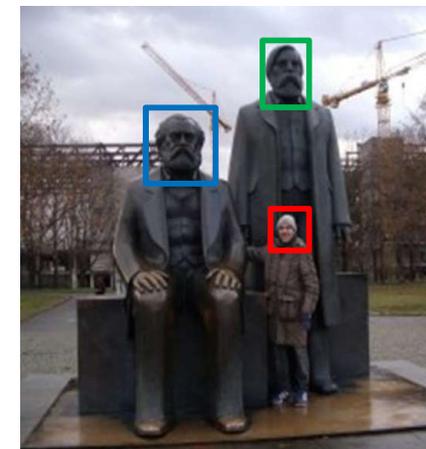
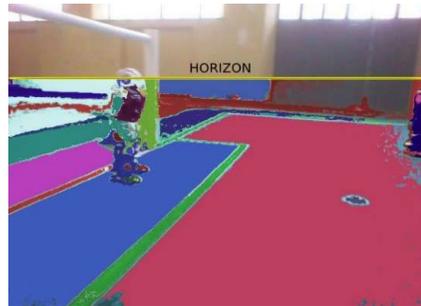
# Processamento delle immagini

 *Alternanza*  
SCUOLA - LAVORO

Docente:  
Domenico Daniele Bloisi



Novembre 2017



# Esercizio 1

---

Utilizzare il linguaggio di programmazione Python e la libreria OpenCV per visualizzare sullo schermo il logo ASL



+

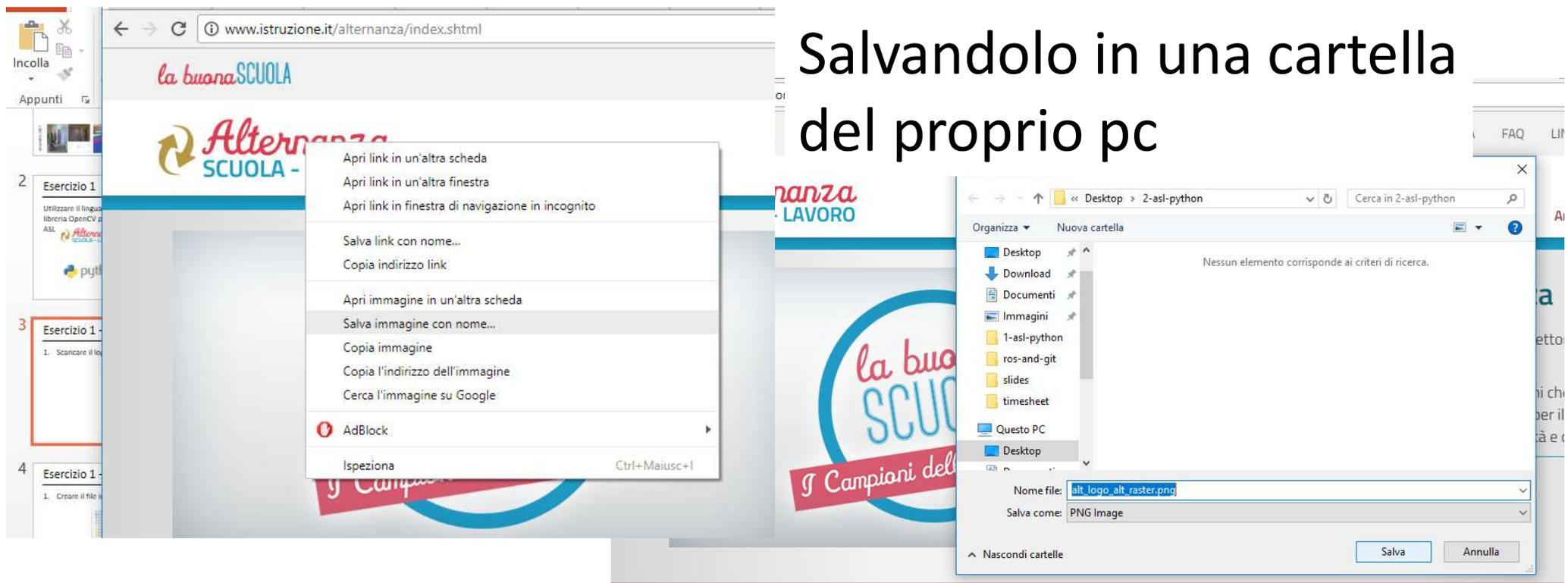


# Esercizio 1 - soluzione

1. Scaricare il logo ASL dal sito del MIUR

<http://www.istruzione.it/alternanza>

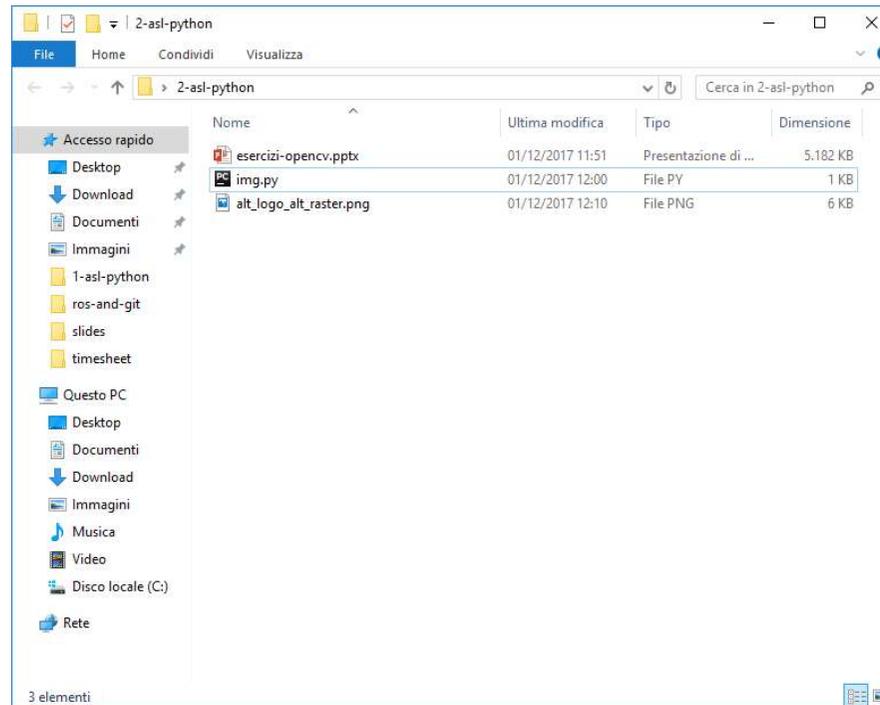
Salvandolo in una cartella  
del proprio pc



# Esercizio 1 - soluzione

---

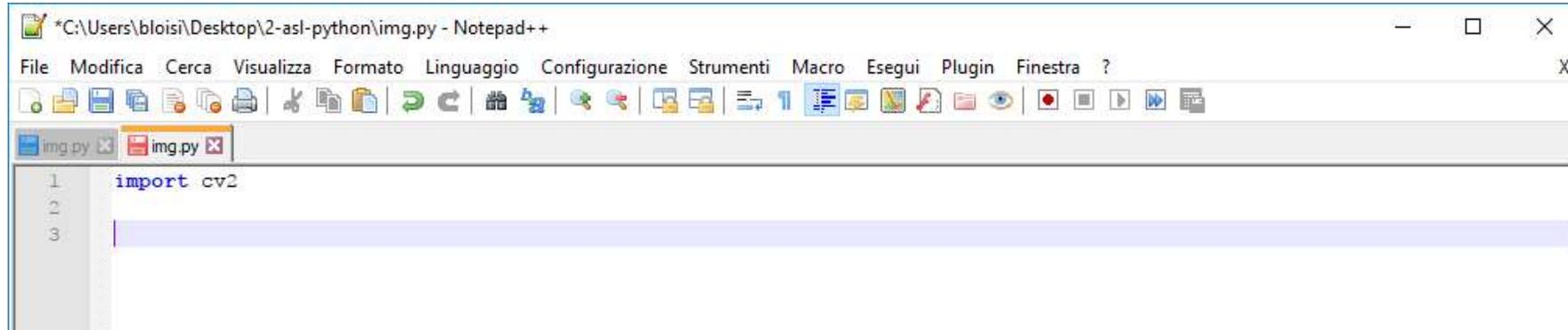
2. Creare il file `img.py` nella stessa cartella in cui era stato salvato in precedenza il logo ASL



# Esercizio 1 - soluzione

---

3. Aprire il file `img.py` in un editor di testo
4. Inserire il codice in linguaggio Python all'interno del file per poter utilizzare la libreria OpenCV

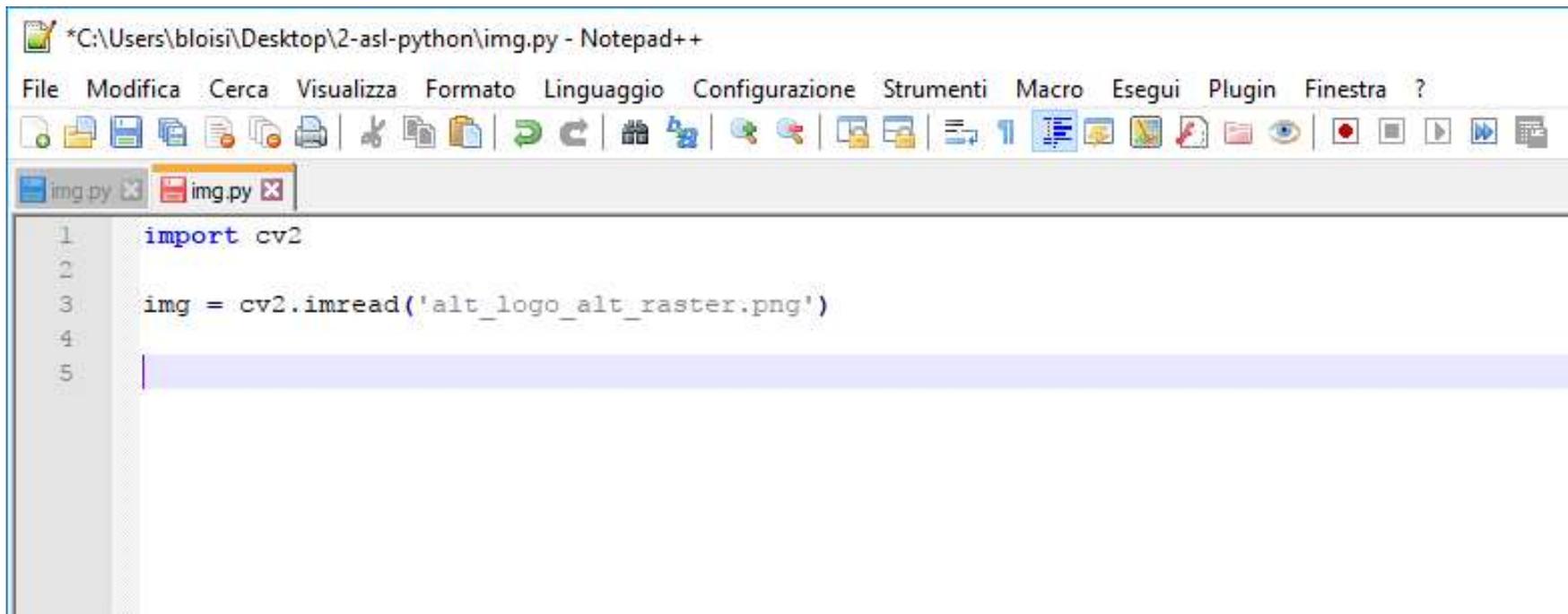


The screenshot shows a Notepad++ window titled '\*C:\Users\bloisi\Desktop\2-asl-python\img.py - Notepad++'. The menu bar includes File, Modifica, Cerca, Visualizza, Formato, Linguaggio, Configurazione, Strumenti, Macro, Esegui, Plugin, Finestra, and ?. The toolbar contains various icons for file operations and editing. The window has two tabs, both labeled 'img.py'. The main editing area shows line numbers 1, 2, and 3 on the left. Line 1 contains the code `import cv2`. Lines 2 and 3 are currently blank and highlighted in light blue.

# Esercizio 1 - soluzione

---

## 5. Leggere l'immagine con imread

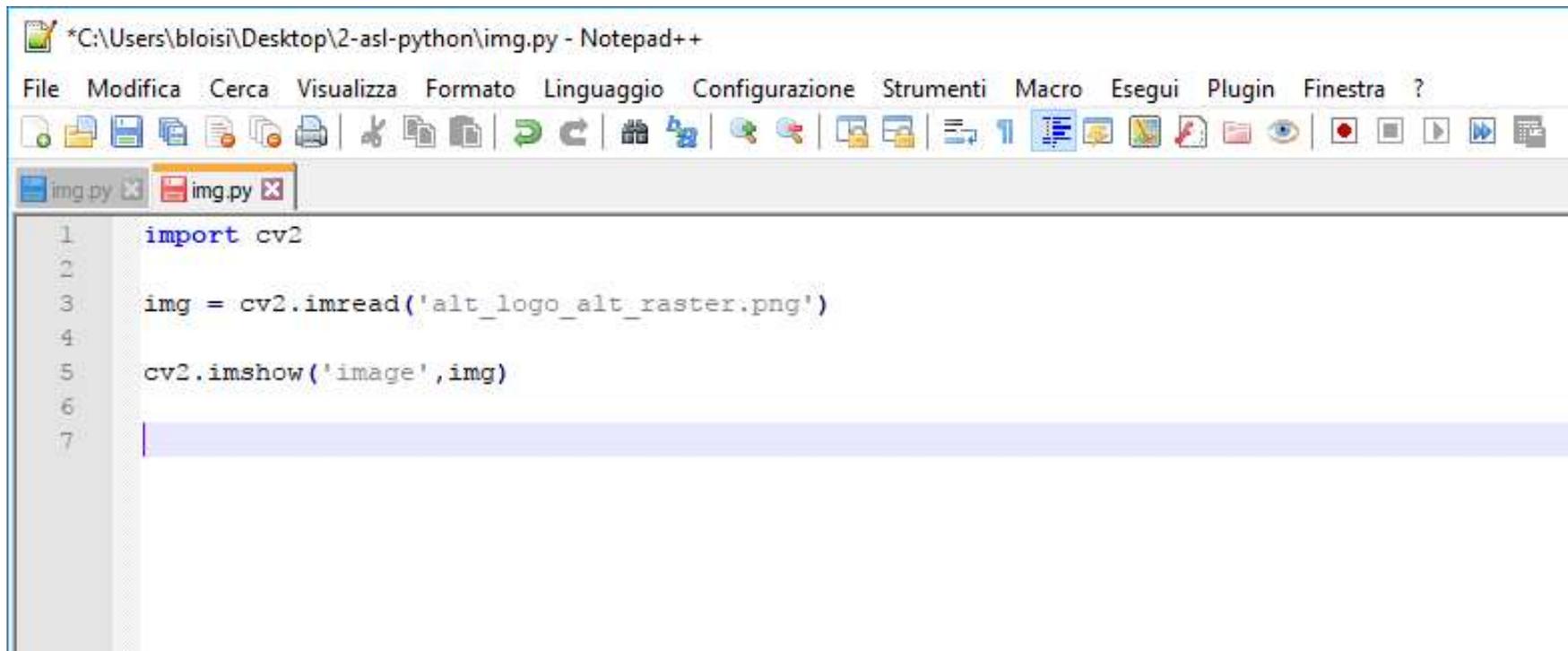


```
*C:\Users\bloisi\Desktop\2-asl-python\img.py - Notepad++
File  Modifica  Cerca  Visualizza  Formato  Linguaggio  Configurazione  Strumenti  Macro  Esegui  Plugin  Finestra  ?
img.py x  img.py x
1  import cv2
2
3  img = cv2.imread('alt_logo_alt_raster.png')
4
5
```

# Esercizio 1 - soluzione

---

## 6. Visualizzare l'immagine con imshow



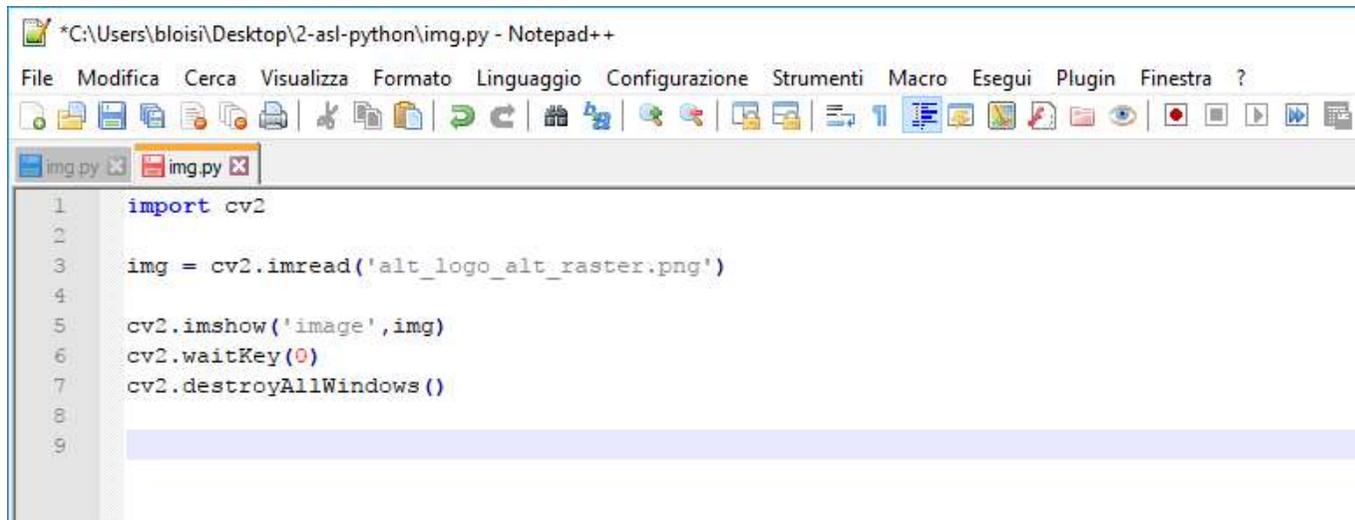
The screenshot shows a Notepad++ window with the following code:

```
*C:\Users\bloisi\Desktop\2-asl-python\img.py - Notepad++
File  Modifica  Cerca  Visualizza  Formato  Linguaggio  Configurazione  Strumenti  Macro  Esegui  Plugin  Finestra  ?
img.py  img.py
1  import cv2
2
3  img = cv2.imread('alt_logo_alt_raster.png')
4
5  cv2.imshow('image',img)
6
7
```

# Esercizio 1 - soluzione

---

7. Per attendere un input da tastiera usare `waitKey`
8. Chiudere la finestra di visualizzazione con `destroyAllWindows`



```
*C:\Users\bloisi\Desktop\2-asl-python\img.py - Notepad++
File Modifica Cerca Visualizza Formato Linguaggio Configurazione Strumenti Macro Esegui Plugin Finestra ?
img.py img.py
1 import cv2
2
3 img = cv2.imread('alt_logo_alt_raster.png')
4
5 cv2.imshow('image',img)
6 cv2.waitKey(0)
7 cv2.destroyAllWindows()
8
9
```

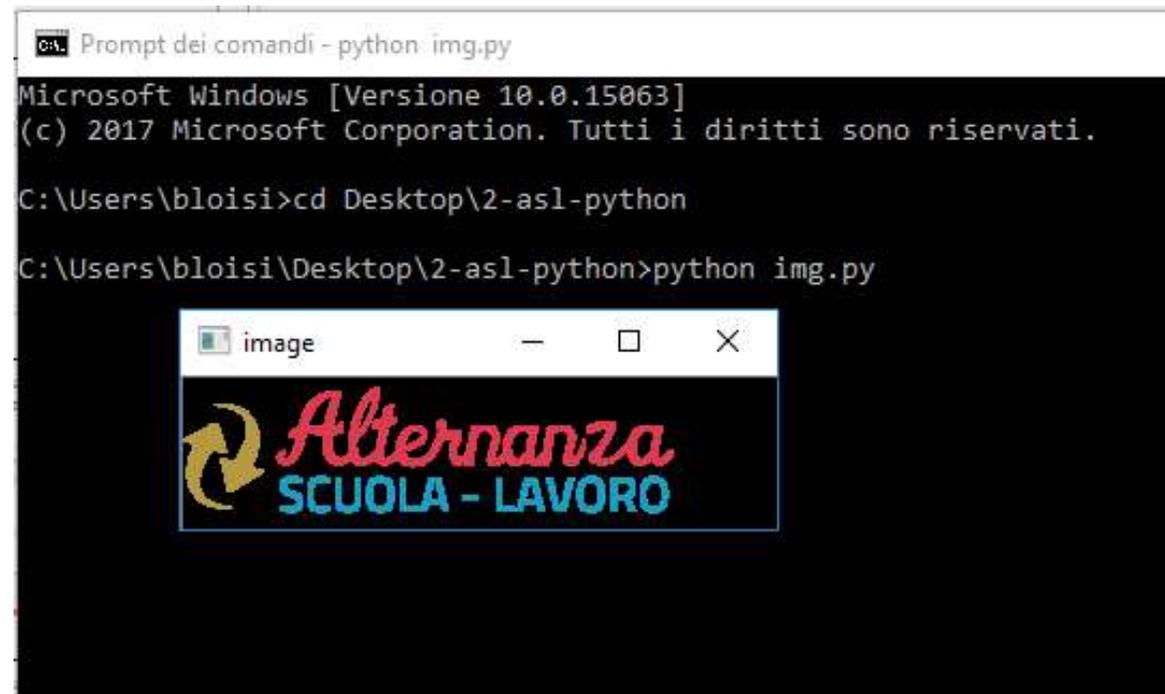
# Esercizio 1 - soluzione

---

9. Usando il terminal, spostarsi nella cartella che contiene il file `img.py` (comando `cd`)

10. digitare

```
python img.py
```



```
C:\> Prompt dei comandi - python img.py
Microsoft Windows [Versione 10.0.15063]
(c) 2017 Microsoft Corporation. Tutti i diritti sono riservati.

C:\Users\bloisi>cd Desktop\2-asl-python

C:\Users\bloisi\Desktop\2-asl-python>python img.py
```

The screenshot shows a Windows command prompt window titled "Prompt dei comandi - python img.py". The terminal output shows the user navigating to the directory "C:\Users\bloisi\Desktop\2-asl-python" and running the command "python img.py". This command successfully displays an image in a separate window titled "image". The image is a logo for "Alternanza SCUOLA - LAVORO", featuring a stylized yellow and red circular arrow icon to the left of the text "Alternanza" in red and "SCUOLA - LAVORO" in blue below it.

# Esercizio 2

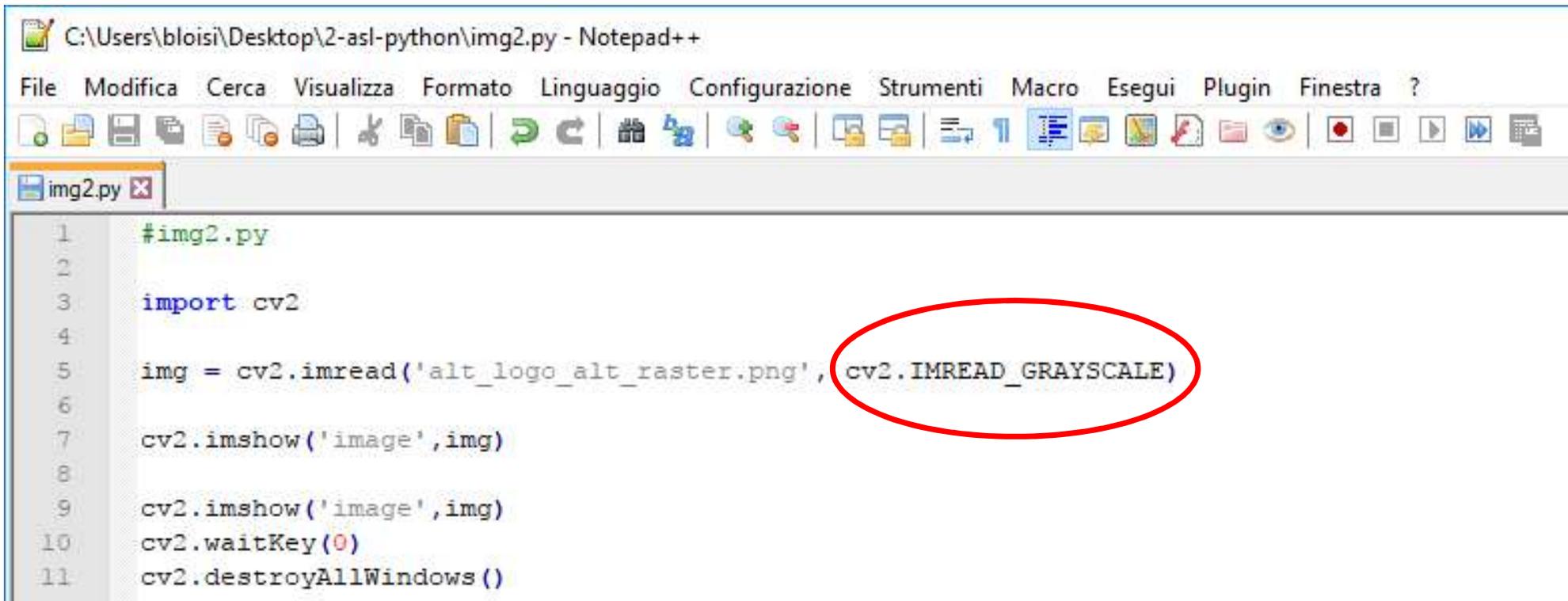
---

Visualizzare l'immagine utilizzata nell'esercizio precedente in bianco e nero

Scrivere il codice Python in un file denominato `img2.py`

# Esercizio 2 - soluzione

---



The image shows a Notepad++ window titled "C:\Users\bloisi\Desktop\2-asl-python\img2.py - Notepad++". The menu bar includes "File", "Modifica", "Cerca", "Visualizza", "Formato", "Linguaggio", "Configurazione", "Strumenti", "Macro", "Esegui", "Plugin", and "Finestra?". The toolbar contains various icons for file operations and editing. The code editor shows the following Python code:

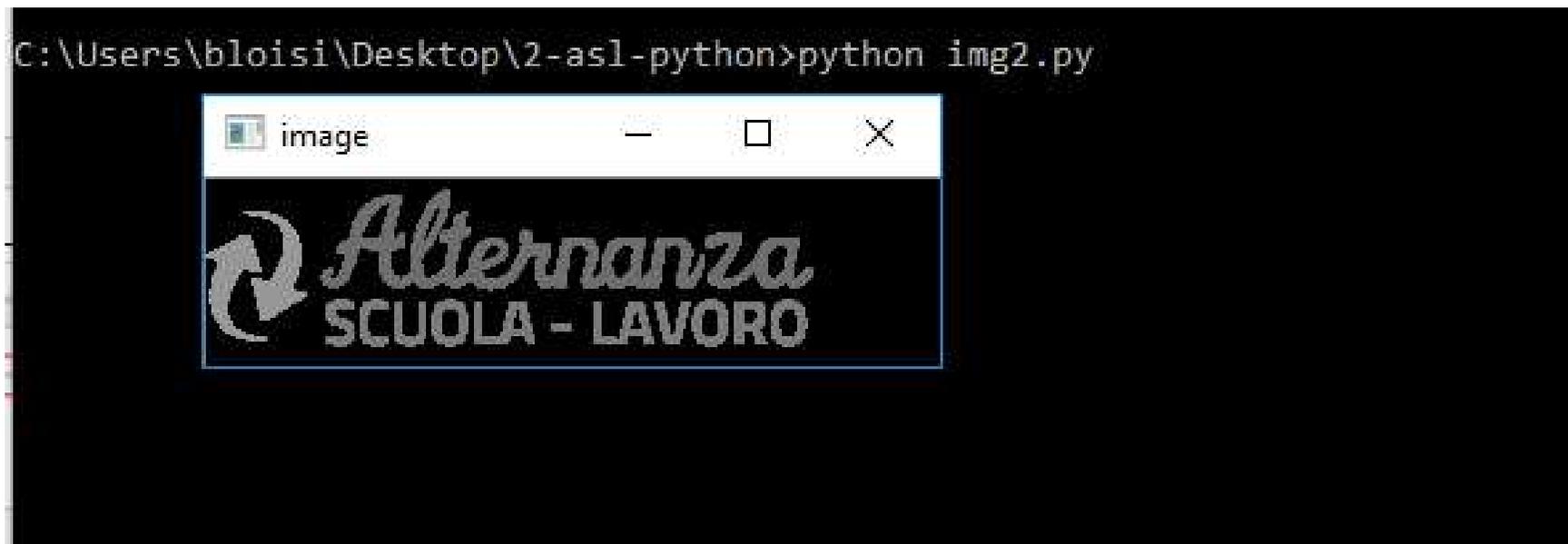
```
1 #img2.py
2
3 import cv2
4
5 img = cv2.imread('alt_logo_alt_raster.png', cv2.IMREAD_GRAYSCALE)
6
7 cv2.imshow('image',img)
8
9 cv2.imshow('image',img)
10 cv2.waitKey(0)
11 cv2.destroyAllWindows()
```

The parameter `cv2.IMREAD_GRAYSCALE` in the `cv2.imread` function call on line 5 is circled in red.

# Esercizio 2 - soluzione

---

Ecco cosa si ottiene eseguendo `img2.py`

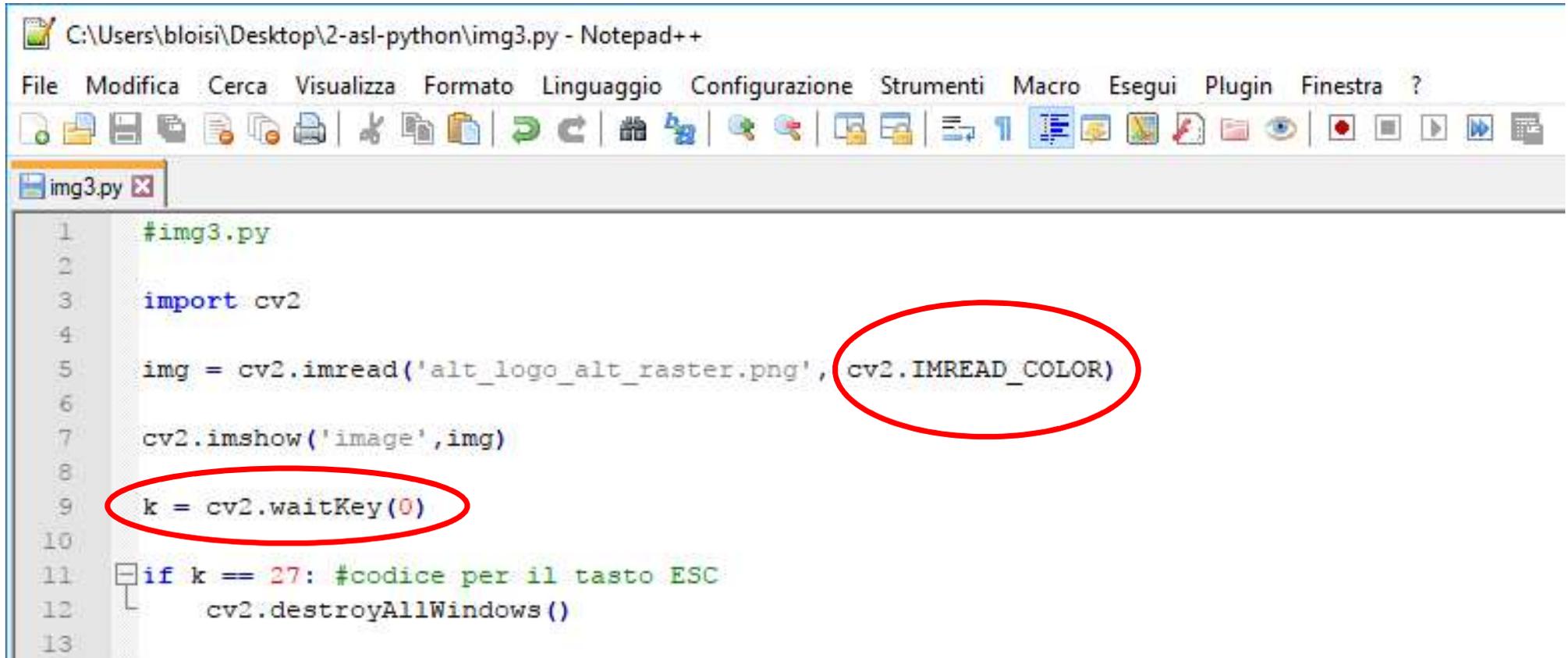


# Esercizio 3

---

Si crei un file `img3.py` che visualizzi il logo ASL a colori e che possa essere terminato solo se viene pigiato il tasto ESC

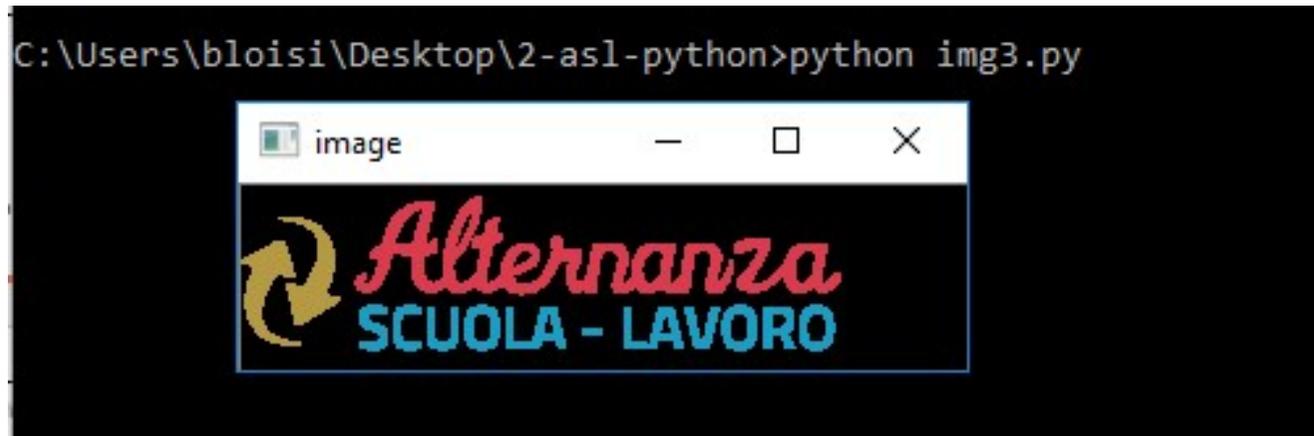
# Esercizio 3 - prima soluzione



```
C:\Users\bloisi\Desktop\2-asl-python\img3.py - Notepad++
File  Modifica  Cerca  Visualizza  Formato  Linguaggio  Configurazione  Strumenti  Macro  Esegui  Plugin  Finestra  ?
img3.py x
1  #img3.py
2
3  import cv2
4
5  img = cv2.imread('alt_logo_alt_raster.png', cv2.IMREAD_COLOR)
6
7  cv2.imshow('image',img)
8
9  k = cv2.waitKey(0)
10
11  if k == 27: #codice per il tasto ESC
12      cv2.destroyAllWindows()
13
```

# Esercizio 3 - prima soluzione

---



Cosa accade se si pigia un tasto diverso da ESC?

# Esercizio 3 - seconda soluzione

```
C:\Users\bloisi\Desktop\2-asl-python\img3-bis.py - Notepad++
File Modifica Cerca Visualizza Formato Linguaggio Configurazione Strumenti Macro Esegui Plugin Finestra ?
img3.py x img3-bis.py x
1 #img3-bis.py
2
3 import cv2
4
5 img = cv2.imread('alt_logo_alt_raster.png', cv2.IMREAD_COLOR)
6
7 cv2.imshow('image',img)
8
9 while True:
10     k = cv2.waitKey(0)
11
12     if k == 27:
13         cv2.destroyAllWindows()
14         break
15     else:
16         print('premere il tasto ESC per terminare')
17
18
```

# Esercizio 3 - seconda soluzione

---

```
C:\Users\bloisi\Desktop\2-asl-python>python img3-bis.py
```



The screenshot shows a terminal window with the command `python img3-bis.py` executed. The output is a window titled "image" displaying a logo with a yellow circular arrow and the text "Alternanza SCUOLA - LAVORO" in red and blue.

```
C:\Users\bloisi\Desktop\2-asl-python>python img3-bis.py
premere il tasto ESC per terminare
premere
premere
```



The screenshot shows a terminal window with the command `python img3-bis.py` executed. The output is a window titled "image" displaying a logo with a yellow circular arrow and the text "Alternanza SCUOLA - LAVORO" in red and blue. The terminal also shows the text "premere il tasto ESC per terminare" and "premere" repeated twice.

```
C:\Users\bloisi\Desktop\2-asl-python>python img3-bis.py
premere il tasto ESC per terminare
premere il tasto ESC per terminare
premere il tasto ESC per terminare

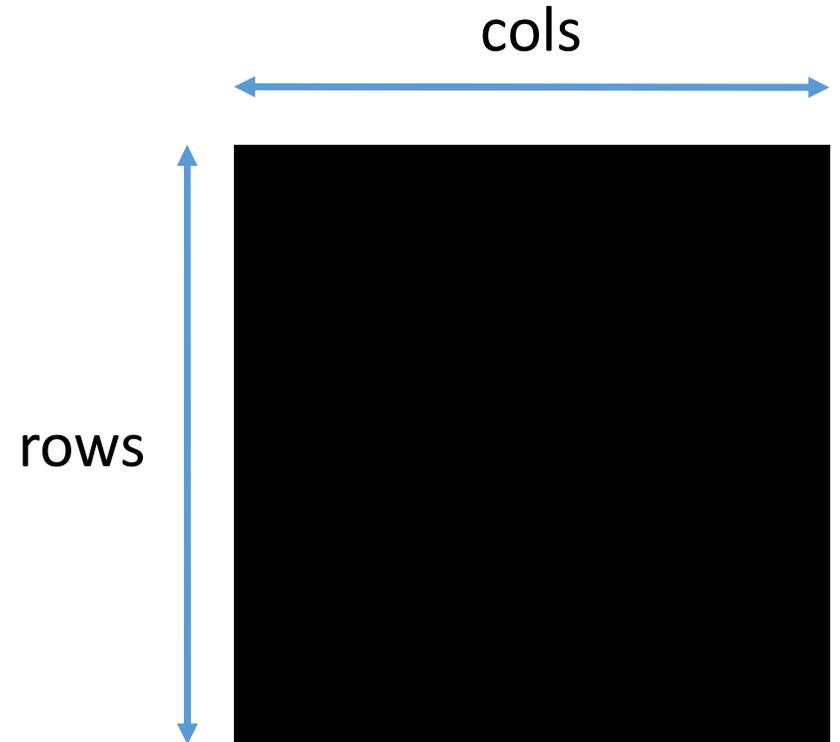
C:\Users\bloisi\Desktop\2-asl-python>
```

# Esercizio 4

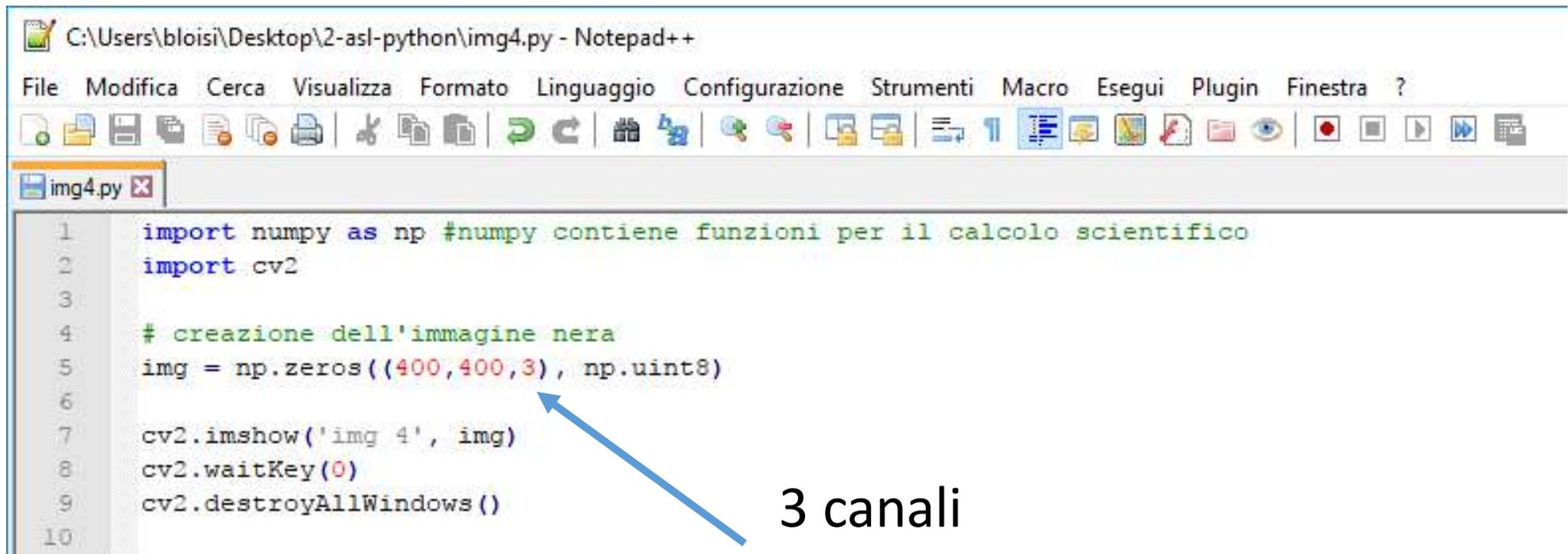
---

Si crei e si visualizzi a schermo una immagine nera di dimensioni 400x400 nera

Una immagine può essere rappresentata come una matrice. Il numero di righe rappresenterà l'altezza dell'immagine, mentre il numero di colonne sarà la larghezza



# Esercizio 4 - soluzione



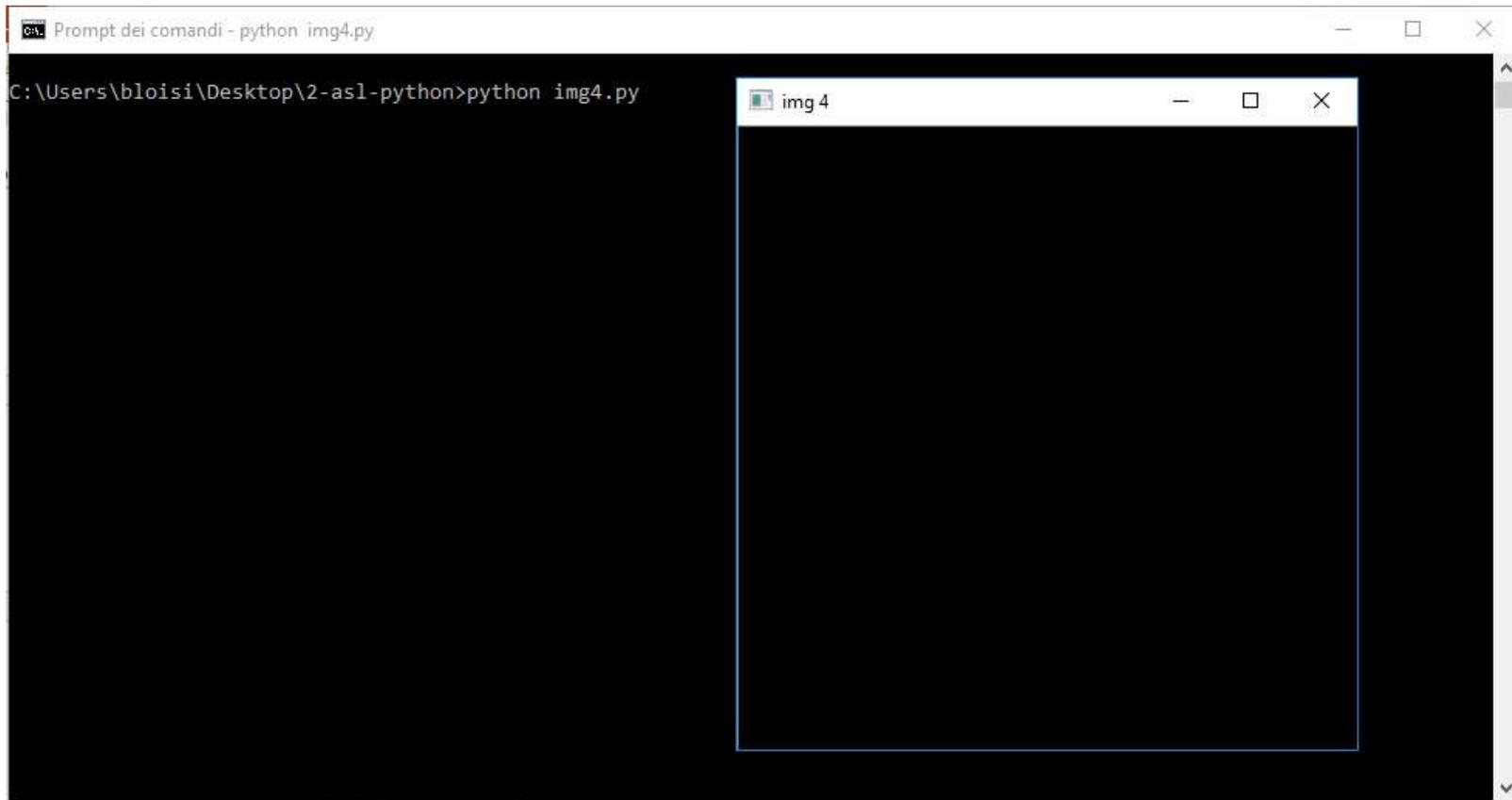
```
1 import numpy as np #numpy contiene funzioni per il calcolo scientifico
2 import cv2
3
4 # creazione dell'immagine nera
5 img = np.zeros((400,400,3), np.uint8)
6
7 cv2.imshow('img 4', img)
8 cv2.waitKey(0)
9 cv2.destroyAllWindows()
10
```

3 canali  
RGB

# Esercizio 4 - soluzione

---

Ecco cosa si ottiene eseguendo `img4.py`

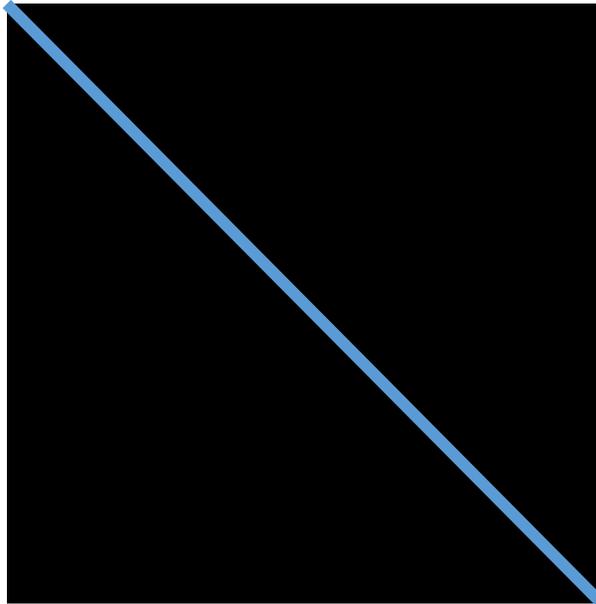


The screenshot shows a Windows command prompt window titled "Prompt dei comandi - python img4.py" with the command `C:\Users\bloisi\Desktop\2-as1-python>python img4.py` entered. A separate window titled "img 4" is open, displaying a solid black image.

# Esercizio 5

---

Disegnare una linea diagonale di colore blu e larghezza 5 pixel sull'immagine creata nell'esercizio precedente



# Esercizio 5 - soluzione

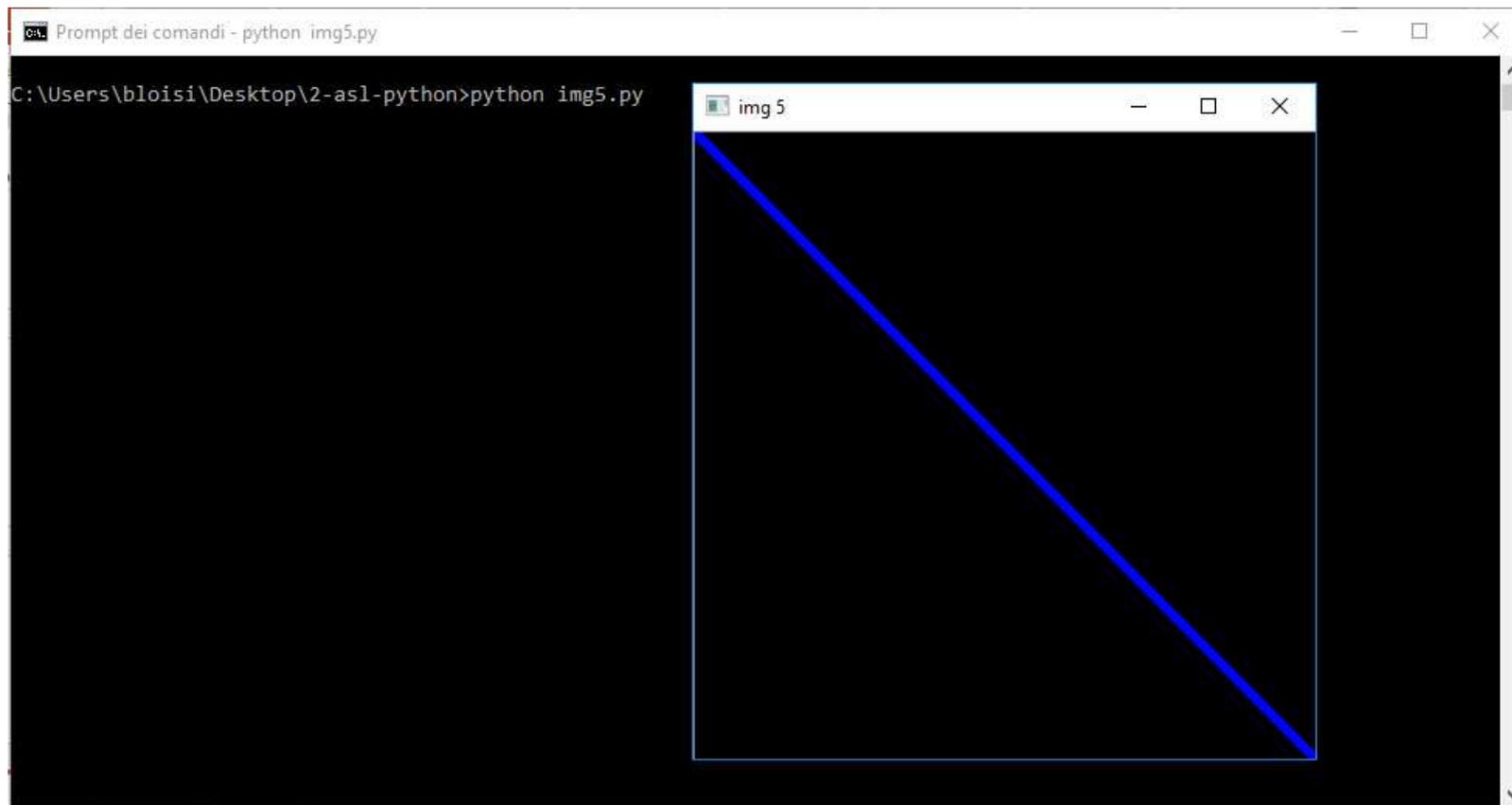
```
C:\Users\bloisi\Desktop\2-asl-python\img5.py - Notepad++
File Modifica Cerca Visualizza Formato Linguaggio Configurazione Strumenti Macro Esegui Plugin Finestra ?
img5.py x
1 import numpy as np #numpy contiene funzioni per il calcolo scientifico
2 import cv2
3
4 # creazione dell'immagine nera
5 img = np.zeros((400,400,3), np.uint8)
6
7 cv2.line(img,(0,0),(399,399),(255,0,0),5)
8
9 cv2.imshow('img 5',img)
10 cv2.waitKey(0)
11 cv2.destroyAllWindows()
12
```

OpenCV usa un modello  
del colore BGR  
(blu, verde, rosso)

# Esercizio 5 - soluzione

---

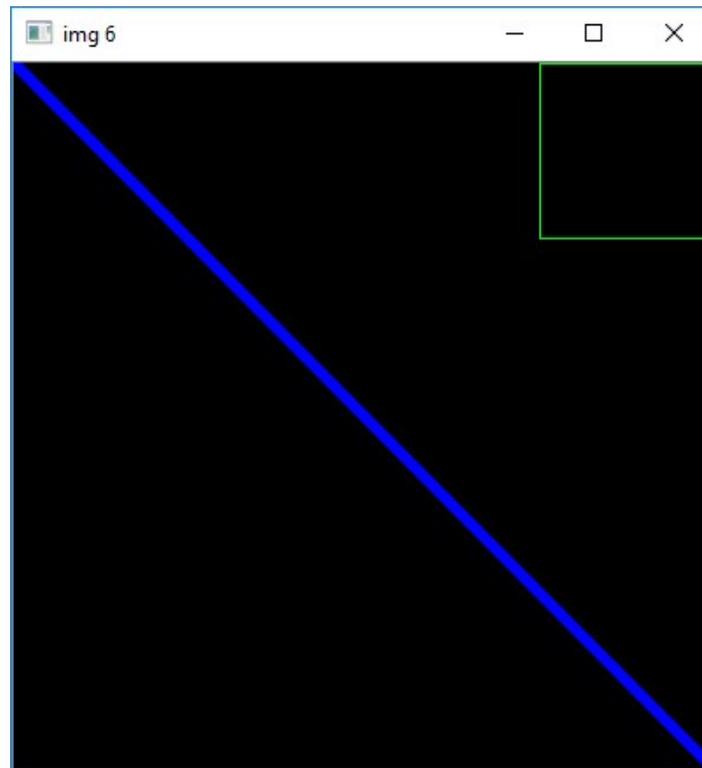
Ecco cosa si ottiene eseguendo `img5.py`



# Esercizio 6

---

Creare un file `img6.py` per disegnare un quadrato verde 100x100 sull'immagine creata nell'esercizio precedente



# Esercizio 6 - soluzione

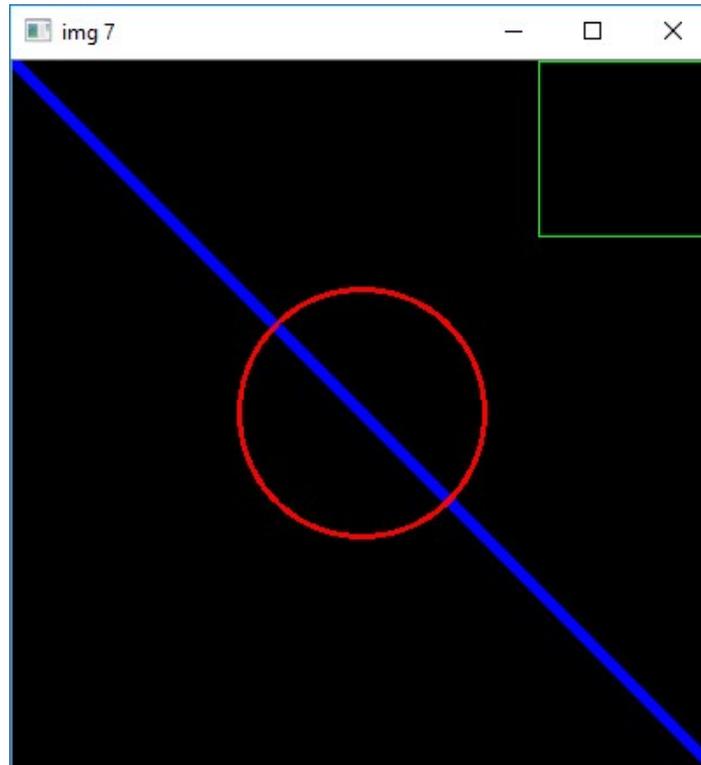
```
C:\Users\bloisi\Desktop\2-asl-python\img6.py - Notepad++
File Modifica Cerca Visualizza Formato Linguaggio Configurazione Strumenti Macro Esegui Plugin Finestra ?
img6.py x
1 #img6.py
2
3 import numpy as np #numpy contiene funzioni per il calcolo scientifico
4 import cv2
5
6 # creazione dell'immagine nera
7 img = np.zeros((400,400,3), np.uint8)
8
9 cv2.line(img, (0,0), (399,399), (255,0,0), 5)
10
11 cv2.rectangle(img, (300,0), (399,99), (0,255,0), 1)
12
13 cv2.imshow('img 6', img)
14 cv2.waitKey(0)
15 cv2.destroyAllWindows()
16
```

In OpenCV possiamo disegnare rettangoli specificandone l'angolo in alto a sinistra e l'angolo in basso a destra

# Esercizio 7

---

Creare un file `img7.py` per disegnare un cerchio rosso di raggio 70 al centro dell'immagine creata nell'esercizio precedente



# Esercizio 7 - soluzione

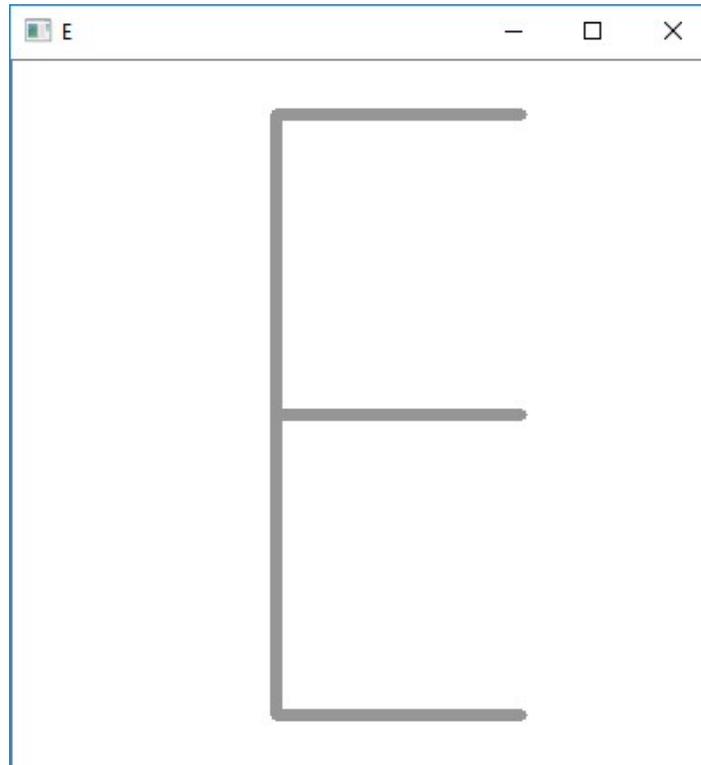
---

```
C:\Users\bloisi\Desktop\2-asl-python\img7.py - Notepad++
File Modifica Cerca Visualizza Formato Linguaggio Configurazione Strumenti Macro Esegui Plugin Finestra ?
img7.py x
1 #img7.py
2
3 import numpy as np #numpy contiene funzioni per il calcolo scientifico
4 import cv2
5
6 # creazione dell'immagine nera
7 img = np.zeros((400,400,3), np.uint8)
8
9 cv2.line(img, (0,0), (399,399), (255,0,0), 5)
10
11 cv2.rectangle(img, (300,0), (399,99), (0,255,0), 1)
12
13 cv2.circle(img, (199,199), 70, (0,0,255), 2)
14
15 cv2.imshow('img 7', img)
16 cv2.waitKey(0)
17 cv2.destroyAllWindows()
18
```

# Esercizio 8

---

Creare un file `img8.py` per disegnare su una immagine bianca la lettera E in grigio



# Esercizio 8 - soluzione

---

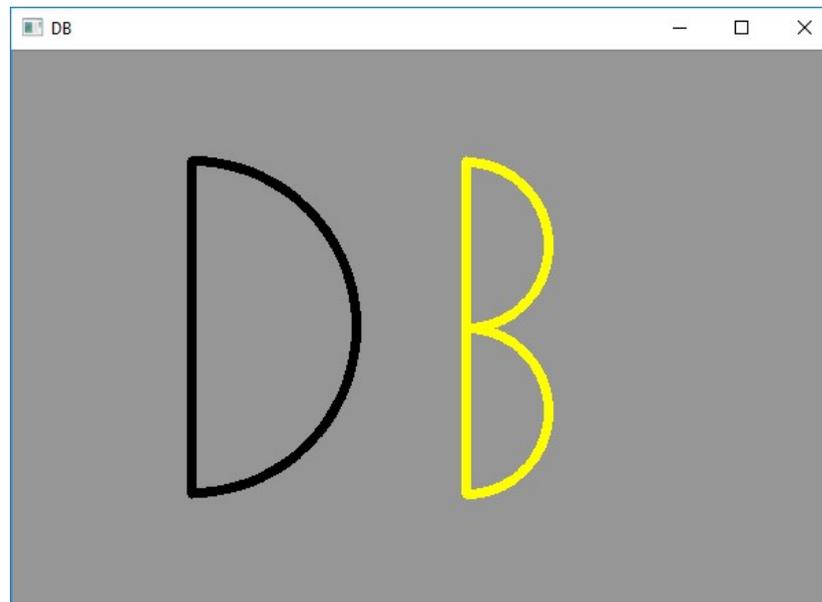
```
C:\Users\blois\Desktop\2-asl-python\img8.py - Notepad++
File Modifica Cerca Visualizza Formato Linguaggio Configurazione Strumenti Macro Esegui Plugin Finestra ?
img8.py x
1 #img8.py
2
3 import numpy as np #numpy contiene funzioni per il calcolo scientifico
4 import cv2
5
6 # creazione dell'immagine nera
7 img = np.zeros((400,400,3), np.uint8)
8
9 img.fill(255)
10
11 cv2.line(img, (150, 30), (150, 370), (150, 150, 150), 5)
12 cv2.line(img, (150, 30), (290, 30), (150, 150, 150), 5)
13 cv2.line(img, (150, 200), (290, 200), (150, 150, 150), 5)
14 cv2.line(img, (150, 370), (290, 370), (150, 150, 150), 5)
15
16 cv2.imshow('E', img)
17 cv2.waitKey(0)
18 cv2.destroyAllWindows()
19
```

# Esercizio 9

---

Creare un file `img9.py` per disegnare su una immagine grigia due lettere, corrispondenti alle iniziali del proprio nome e cognome, la prima in nero e la seconda in giallo

Esempio con DB



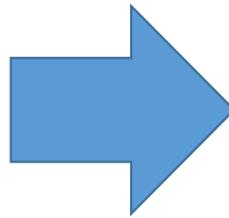
# Esercizio 9 - soluzione con DB

```
C:\Users\bloisi\Desktop\2-asl-python\img9.py - Notepad++
File Modifica Cerca Visualizza Formato Linguaggio Configurazione Strumenti Macro Esegui Plugin Finestra ?
img9.py x
1 #img9.py
2
3 import numpy as np #numpy contiene funzioni per il calcolo scientifico
4 import cv2
5
6 # creazione dell'immagine grigia
7 img = np.zeros((400,600,3), np.uint8)
8 img.fill(150)
9
10 #D
11 cv2.circle(img, (130,199), 120, (0,0,0), 5)
12 cv2.rectangle(img, (0,0), (130,399), (150,150,150), -1)
13 cv2.line(img, (130, 79), (130,320), (0,0,0), 5)
14
15 #B
16 cv2.circle(img, (330,140), 60, (0,255,255), 5)
17 cv2.circle(img, (330,260), 60, (0,255,255), 5)
18 cv2.rectangle(img, (260,0), (330,399), (150,150,150), -1)
19 cv2.line(img, (330, 79), (330,320), (0,255,255), 5)
20
21 cv2.imshow('DB', img)
22 cv2.waitKey(0)
23 cv2.destroyAllWindows()
24
```

# Esercizio 10

---

Scrivere delle istruzioni Python per ruotare di 90 gradi il logo della ASL salvato in precedenza



# Esercizio 10 - prima soluzione

```
C:\Users\bloisi\Desktop\2-asl-python\img10.py - Notepad++
File Modifica Cerca Visualizza Formato Linguaggio Configurazione Strumenti Macro Esegui Plugin Finestra ?
img10.py
1 #img10.py
2
3 import numpy as np #numpy contiene funzioni per il calcolo scientifico
4 import cv2
5
6 img = cv2.imread('alt_logo_alt_raster.png',1)
7 rows,cols,channels = img.shape
8
9 M = cv2.getRotationMatrix2D((cols/2,rows/2),90,1)
10 dst = cv2.warpAffine(img,M,(cols,rows))
11
12 cv2.imshow('rotazione 90', dst)
13 cv2.waitKey(0)
14 cv2.destroyAllWindows()
15
```

Che cosa accade  
eseguendo  
questo codice?

# Esercizio 10 - seconda soluzione

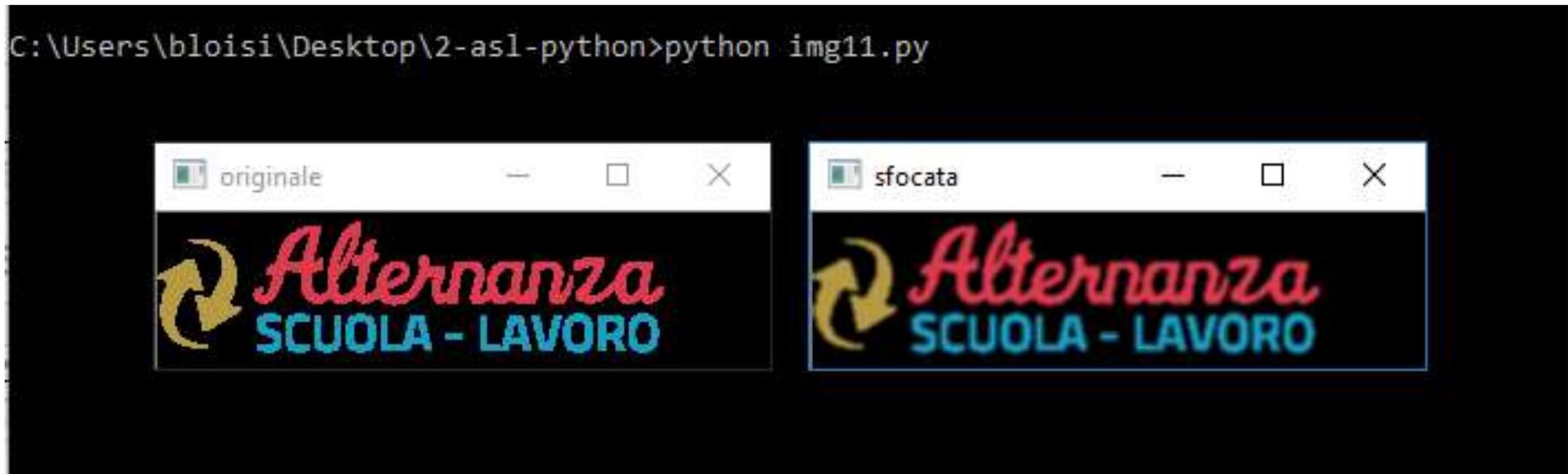
```
C:\Users\bloisi\Desktop\2-asl-python\img10-bis.py - Notepad++
File Modifica Cerca Visualizza Formato Linguaggio Configurazione Strumenti Macro Esegui Plugin Finestra ?
img10-bis.py x
1 #img10-bis.py
2
3 import numpy as np #numpy contiene funzioni per il calcolo scientifico
4 import cv2
5
6 img = cv2.imread('alt_logo_alt_raster.png',1)
7 rows,cols,channels = img.shape
8
9 M = cv2.getRotationMatrix2D((cols,rows/2),90,1)
10 dst = cv2.warpAffine(img,M,(cols*2,rows*6))
11
12 cv2.imshow('rotazione 90', dst)
13 cv2.waitKey(0)
14 cv2.destroyAllWindows()
15
```

E ora che cosa accade?

# Esercizio 11

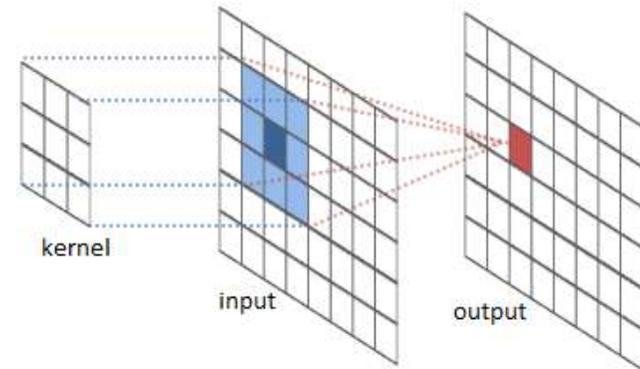
---

Scrivere delle istruzioni Python per sfocare il logo della ASL salvato in precedenza



# Esercizio 11 - soluzione

```
C:\Users\bloisi\Desktop\2-asl-python\img11.py - Notepad++
File Modifica Cerca Visualizza Formato Linguaggio Configurazione
img11.py x
1 #img11.py
2
3 import cv2
4 import numpy as np
5
6 img = cv2.imread('alt_logo_alt_raster.png')
7
8 kernel = np.ones((3,3),np.float32)/9
9 dst = cv2.filter2D(img,-1,kernel)
10
11 cv2.imshow('originale', img)
12 cv2.imshow('sfocata', dst)
13 cv2.waitKey(0)
14 cv2.destroyAllWindows()
15
```



Convoluzione

$$\text{kernel} = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

# Esercizio 12

---

Modificare il codice nel file `img11.py` in modo da ottenere una immagine più sfocata come quella mostrata sotto.

```
C:\Users\bloisi\Desktop\2-as1-python>python img12.py
```



# Esercizio 13

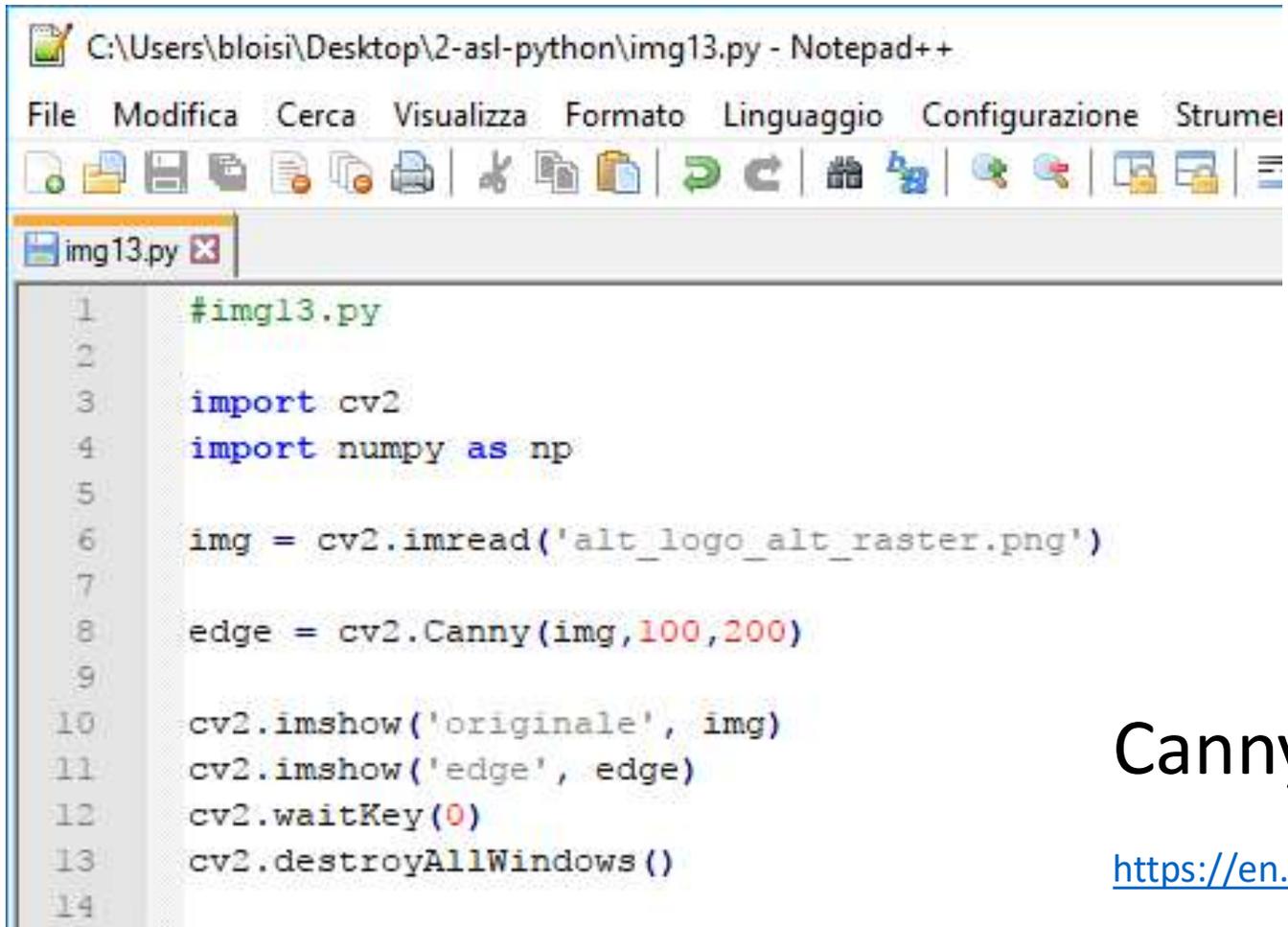
---

Scrivere delle istruzioni Python per estrarre gli edge dal logo della ASL salvato in precedenza

```
C:\Users\bloisi\Desktop\2-asl-python>python img13.py
```



# Esercizio 13 - soluzione



```
C:\Users\bloisi\Desktop\2-asl-python\img13.py - Notepad++
File  Modifica  Cerca  Visualizza  Formato  Linguaggio  Configurazione  Strumen
img13.py x
1  #img13.py
2
3  import cv2
4  import numpy as np
5
6  img = cv2.imread('alt_logo_alt_raster.png')
7
8  edge = cv2.Canny(img, 100, 200)
9
10 cv2.imshow('originale', img)
11 cv2.imshow('edge', edge)
12 cv2.waitKey(0)
13 cv2.destroyAllWindows()
14
```

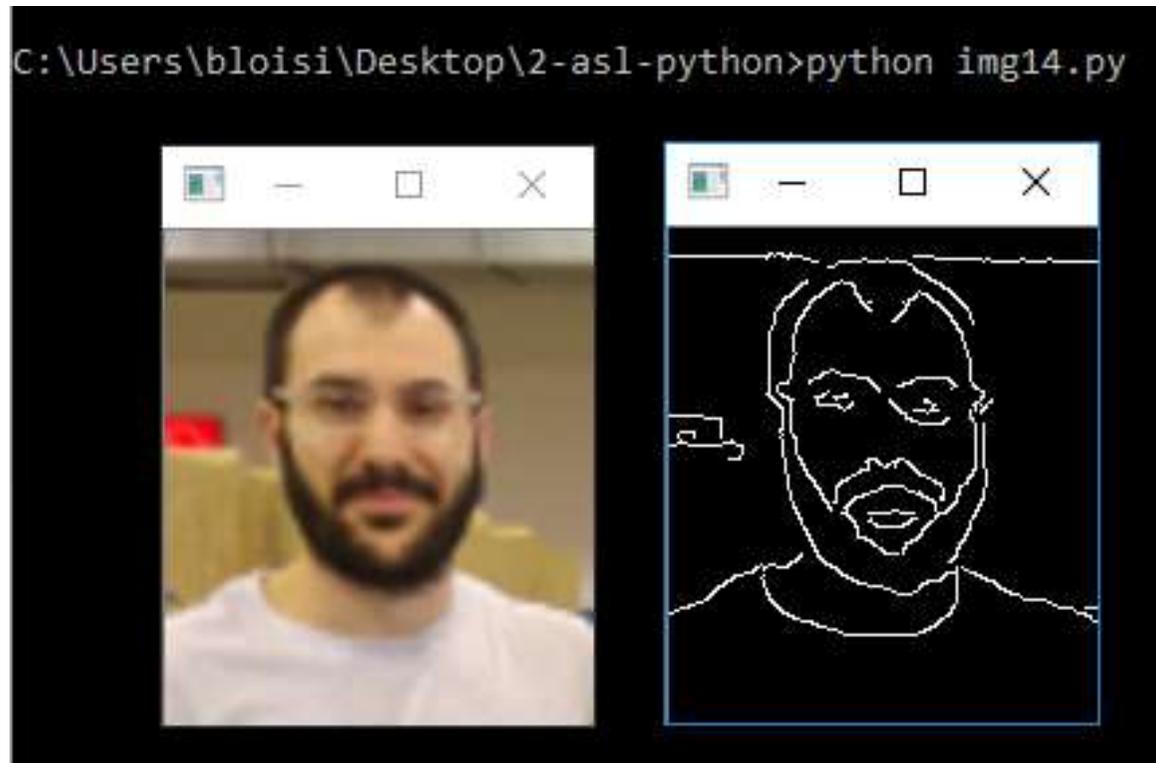
Canny edge detector

[https://en.wikipedia.org/wiki/Canny\\_edge\\_detector](https://en.wikipedia.org/wiki/Canny_edge_detector)

# Esercizio 14

---

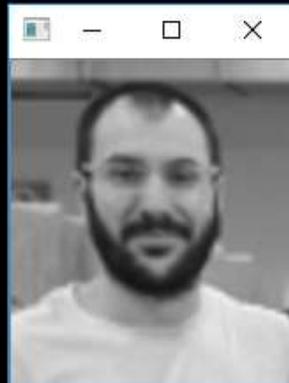
Scrivere un codice Python per estrarre gli edge da una foto del proprio volto



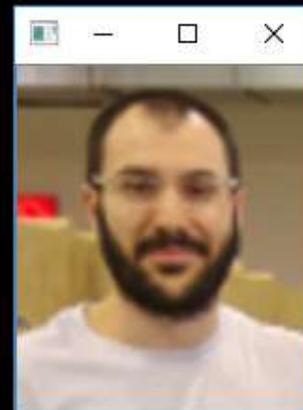
# Esercizio 15

Scrivere un codice Python che permetta di scegliere da tastiera se mostrare l'immagine del proprio volto a colori o in bianco e nero

```
C:\Users\bloisi\Desktop\2-asl-python>python img15.py
Visualizzatore del volto
premere c per immagine a colori
g per immagine in scala di grigi
Scegliere come visualizzare [c/g]: g
visualizzazione in scala di grigi
```



```
C:\Users\bloisi\Desktop\2-asl-python>python img15.py
Visualizzatore del volto
premere c per immagine a colori
g per immagine in scala di grigi
Scegliere come visualizzare [c/g]: c
visualizzazione a colori
```



# Esercizio 15 - soluzione

```
1 #img15.py
2
3 import cv2
4 import numpy as np
5 import os
6
7 print('Visualizzatore del volto')
8 print('premere c per immagine a colori')
9 print('g per immagine in scala di grigi')
10
11 input_volto = input('Scegliere come visualizzare [c/g]: ')
12
13 v = -1 # valore di visualizzazione iniziale
14 if input_volto == 'g':
15     print('visualizzazione in scala di grigi')
16     v = 0
17 elif input_volto == 'c':
18     print('visualizzazione a colori')
19     v = 1
20 else:
21     print('valore incorretto')
22     print('terminazione')
23     os._exit(1)
24
25 img = cv2.imread('volto.png', v)
26
27 cv2.imshow('volto', img)
28 cv2.waitKey(0)
29 cv2.destroyAllWindows()
30
```

Usiamo il parametro `v` per controllare in `imread` la modalità di lettura dell'immagine denominata `volto.png`

# Esercizio 16

---

Scrivere un codice Python che permetta di scegliere da tastiera se mostrare l'immagine del proprio volto sfocata oppure se mostrare gli edge estratti da essa

```
C:\Users\bloisi\Desktop\2-asl-python>python img16.py
Visualizzatore del volto
premere s per immagine sfocata
e per visualizzare gli edge
Scegliere come visualizzare [s/e]: s
visualizzazione sfocata
```

A screenshot of a Windows window titled "Visualizzatore del volto" showing a blurred image of a man's face. The window has standard Windows window controls (minimize, maximize, close) at the top.

```
C:\Users\bloisi\Desktop\2-asl-python>python img16.py
Visualizzatore del volto
premere s per immagine sfocata
e per visualizzare gli edge
Scegliere come visualizzare [s/e]: e
visualizzazione degli edge
```

A screenshot of a Windows window titled "Visualizzatore del volto" showing the edge detection of the man's face. The image is rendered in white lines on a black background. The window has standard Windows window controls at the top.

# Esercizio 16 - soluzione

```
1  #img16.py
2
3  import cv2
4  import numpy as np
5  import os
6
7  print('Visualizzatore del volto')
8  print('premere s per immagine sfocata')
9  print('e per visualizzare gli edge')
10
11  input_volto = input('Scegliere come visualizzare [s/e]: ')
12
13  img = cv2.imread('volto.png')
14
15  if input_volto == 's':
16      print('visualizzazione sfocata')
17      kernel = np.ones((7,7),np.float32)/49
18      dst = cv2.filter2D(img,-1,kernel)
19  elif input_volto == 'e':
20      print('visualizzazione degli edge')
21      dst = cv2.Canny(img,100,200)
22  else:
23      print('valore incorretto')
24      print('terminazione')
25      os._exit(1)
26
27  cv2.imshow('volto', dst)
28  cv2.waitKey(0)
29  cv2.destroyAllWindows()
30
```

img è l'immagine originale, mentre dst è l'immagine processata

# Esercizio 17

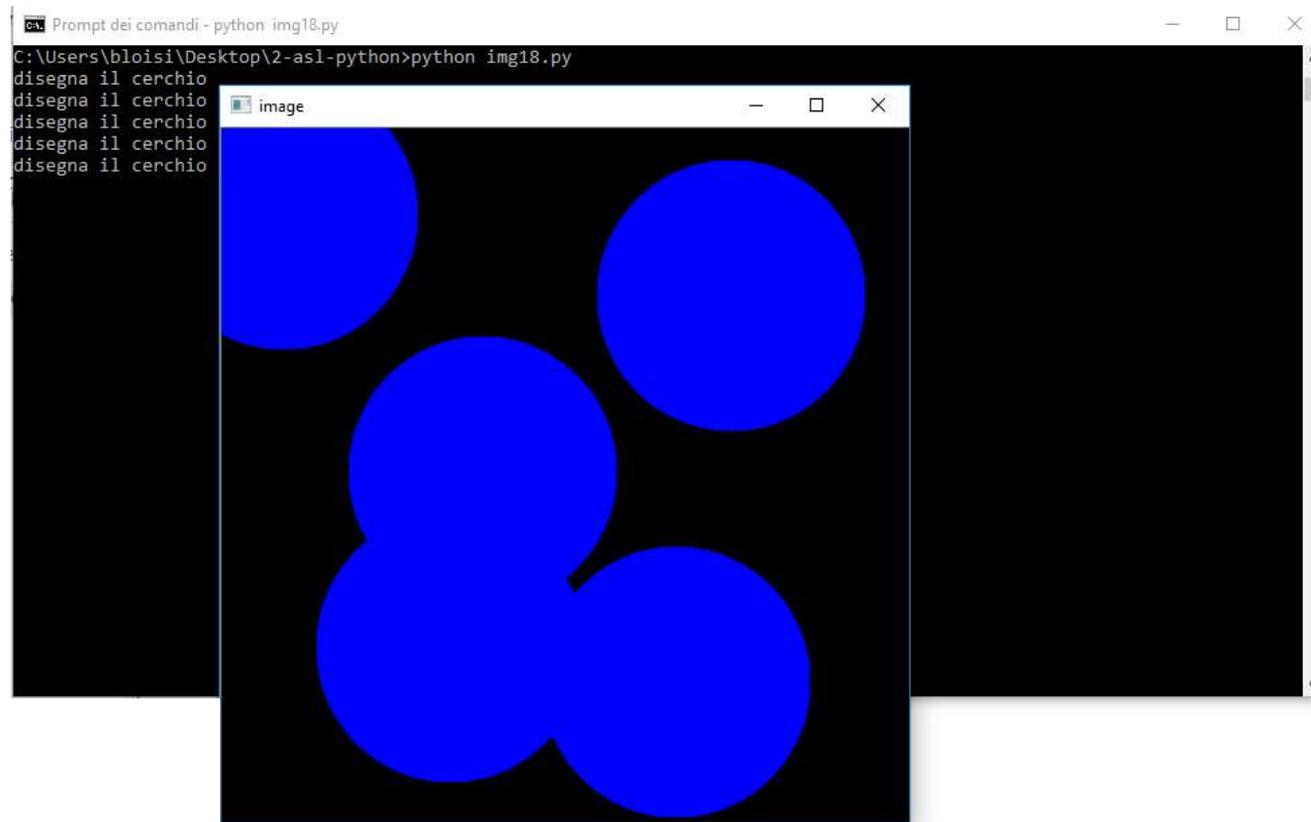
---

Provare a mettere insieme il codice degli esercizi 15 e 16 in modo da consentire all'utente di scegliere da tastiera se visualizzare:

- L'immagine a colori (tasto c)
- L'immagine in scala di grigi (tasto g)
- L'immagine sfocata (tasto s)
- Gli edge (tasto e)

# Esercizio 18

Scrivere un codice Python per disegnare un cerchio blu su una immagine nera con un doppio click del pulsante sinistro del mouse



# Esercizio 18 - soluzione

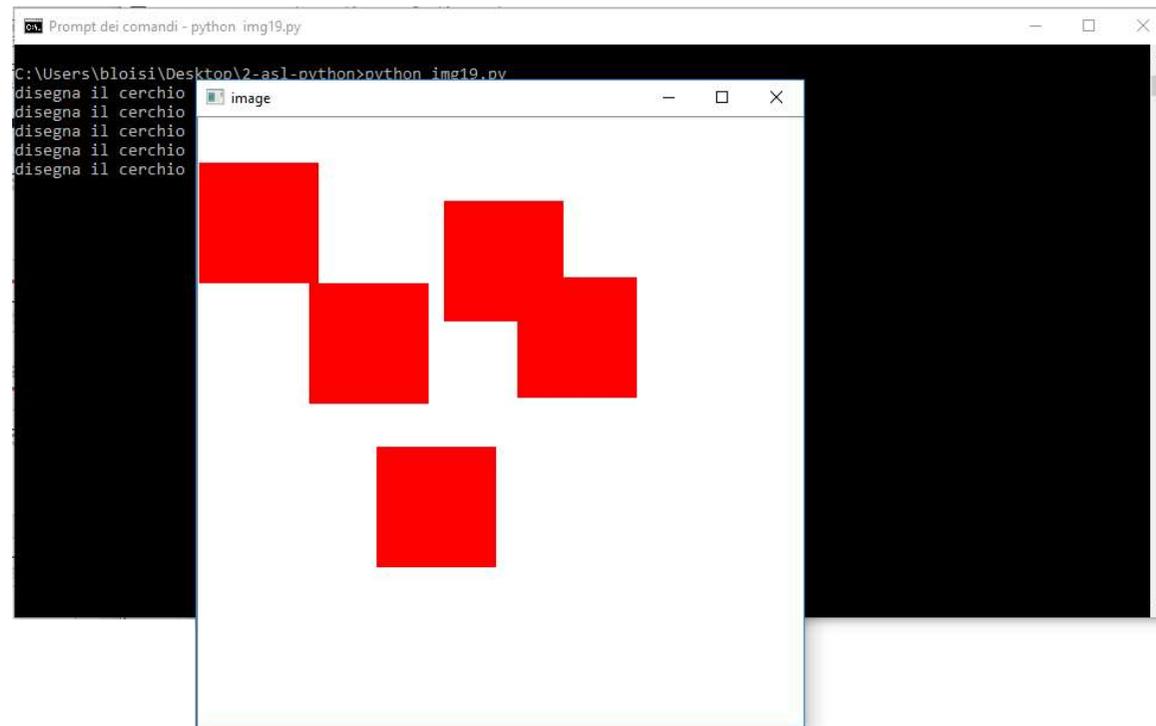
```
1  import cv2
2  import numpy as np
3
4  # mouse callback function
5  def draw_circle(event,x,y,flags,param):
6      if event == cv2.EVENT_LBUTTONDOWN:
7          print('disegna il cerchio')
8          cv2.circle(img, (x,y), 100, (255,0,0), -1)
9
10 # Create a black image, a window and bind the function to window
11 img = np.zeros((512,512,3), np.uint8)
12 cv2.namedWindow('image')
13 cv2.setMouseCallback('image',draw_circle)
14
15 while(1):
16     cv2.imshow('image',img)
17     if cv2.waitKey(30) == 27:
18         break
19
20 cv2.destroyAllWindows()
21
```

draw\_circle è la  
funzione per la  
gestione degli  
eventi generati  
dal mouse

# Esercizio 19

---

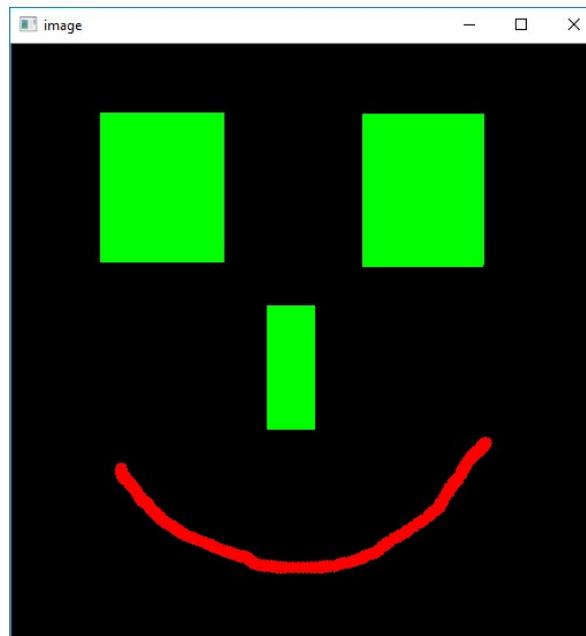
Modificare il codice dell'esercizio precedente per disegnare un quadrato rosso su una immagine bianca con un doppio click del pulsante sinistro del mouse



# Esercizio 20

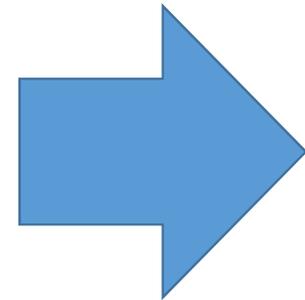
---

Scrivere un codice Python per disegnare rettangoli verdi la cui dimensione possa essere controllata dall'utente. Inoltre, nel caso in cui venga premuto il tasto m, il codice deve spostarsi sulla modalità «pennello» e permettere di disegnare curve rosse al muoversi del mouse



# Esercizio 20 - soluzione

```
1 import cv2
2 import numpy as np
3
4 drawing = False # true if mouse is pressed
5 mode = True # if True, draw rectangle. Press 'm' to toggle to curve
6 ix,iy = -1,-1
7
8 # mouse callback function
9 def draw_circle(event,x,y,flags,param):
10     global ix,iy,drawing,mode
11     if event == cv2.EVENT_LBUTTONDOWN:
12         drawing = True
13         ix,iy = x,y
14     elif event == cv2.EVENT_MOUSEMOVE:
15         if drawing == True:
16             if mode == True:
17                 cv2.rectangle(img, (ix,iy) , (x,y) , (0,255,0) ,-1)
18             else:
19                 cv2.circle(img, (x,y) , 5, (0,0,255) ,-1)
20     elif event == cv2.EVENT_LBUTTONUP:
21         drawing = False
22         if mode == True:
23             cv2.rectangle(img, (ix,iy) , (x,y) , (0,255,0) ,-1)
24         else:
25             cv2.circle(img, (x,y) , 5, (0,0,255) ,-1)
```



# Esercizio 20 - soluzione

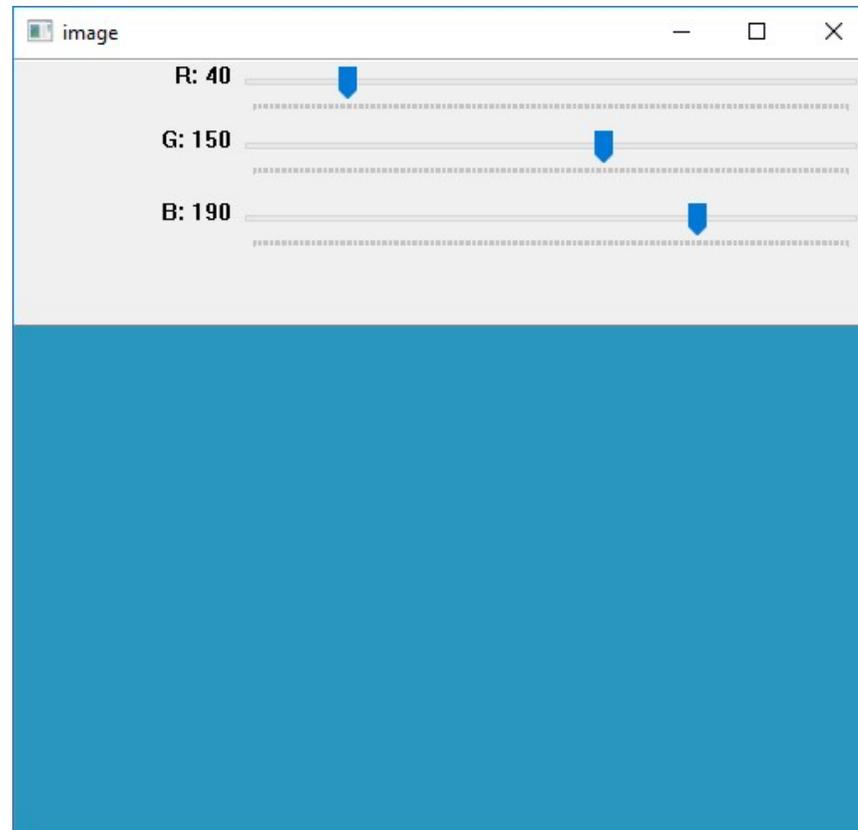
---

```
26     .....
27     img = np.zeros((512,512,3), np.uint8)
28     cv2.namedWindow('image')
29     cv2.setMouseCallback('image',draw_circle)
30
31     while(1):
32         cv2.imshow('image',img)
33         k = cv2.waitKey(1) & 0xFF
34         if k == ord('m'):
35             .....
36             mode = not mode
37         elif k == 27:
38             .....
39             break
40
41     cv2.destroyAllWindows()
```

# Esercizio 21

---

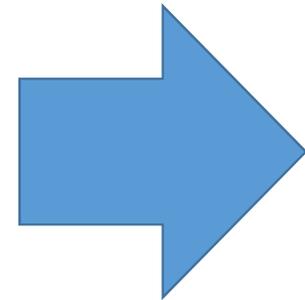
Scrivere un codice Python che consenta all'utente di colorare una immagine con valori RGB scelti usando una trackbar per ogni canale



# Esercizio 21 - soluzione

---

```
1  import cv2
2  import numpy as np
3
4  def nothing(x):
5      pass
6
7  # Create a black image, a window
8  img = np.zeros((300,512,3), np.uint8)
9  cv2.namedWindow('image')
10
11 # create trackbars for color change
12 cv2.createTrackbar('R','image',0,255,nothing)
13 cv2.createTrackbar('G','image',0,255,nothing)
14 cv2.createTrackbar('B','image',0,255,nothing)
```



# Esercizio 21 - soluzione

---

```
16 while(1):
17     cv2.imshow('image',img)
18     k = cv2.waitKey(1) & 0xFF
19     if k == 27:
20         break
21     # get current positions of four trackbars
22     r = cv2.getTrackbarPos('R','image')
23     g = cv2.getTrackbarPos('G','image')
24     b = cv2.getTrackbarPos('B','image')
25
26     img[:] = [b,g,r]
27
28 cv2.destroyAllWindows()
```

# Esercizio 22

---

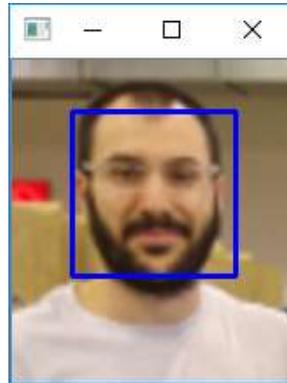
Modificare il codice dell'esercizio precedente in modo da lavorare su immagini in scala di grigi con una unica trackbar



# Esercizio 23

---

Scrivere un codice Python per rilevare un volto in una immagine



# Esercizio 23 - soluzione

---

```
1 import numpy as np
2 import cv2
3
4 face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
5
6 img = cv2.imread('volto.png')
7 gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
8
9 faces = face_cascade.detectMultiScale(gray, 1.1, 2)
10 for (x,y,w,h) in faces:
11     img = cv2.rectangle(img, (x,y), (x+w,y+h), (255,0,0), 2)
12
13 cv2.imshow('img',img)
14 cv2.waitKey(0)
15 cv2.destroyAllWindows()
16
```



# Esercizio 24

---

Provare ad eseguire il codice precedente su una immagine contenente il proprio volto, disegnando un rettangolo verde

# Esercizio 25

Provare ad eseguire il codice precedente su una immagine contenente più volti



# Esercizio 26

---

Scrivere un codice Python che

1. chieda all'utente all'utente di inserire il nome di una immagine da visualizzare
2. Visualizzi l'immagine se esistente
3. Mostri un rettangolo rosso intorno ad ogni volto se l'utente preme il tasto f
4. Termini nel caso venga premuto il tasto q

# References and Credits

---

Alexander Mordvintsev & Abid K

*OpenCV-Python Tutorials Documentation*

*Release 1*



UNIVERSITÀ  
di **VERONA**

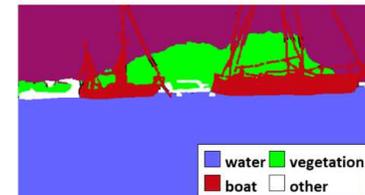
Dipartimento  
di **INFORMATICA**



# Processamento delle immagini

 *Alternanza*  
SCUOLA - LAVORO

Docente:  
Domenico Daniele Bloisi



Novembre 2017

