

lezione sulle immagini

October 27, 2017

Definiamo una funzione `create()` che crea un'immagine di larghezza `w` e altezza `h` con tutti i pixel colorati con `c`

```
In [1]: def create(w,h,c=(0,0,0)):
        img = []
        for _ in range(h):
            riga = []
            for _ in range(w):
                riga+=[c]
            img+=riga
        return img
```

```
In [2]: verde = create(5,20, (0,255,0))
        verde
```

```
Out[2]: [[(0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0)],
          [(0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0)],
          [(0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0)],
          [(0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0)],
          [(0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0)],
          [(0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0)],
          [(0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0)],
          [(0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0)],
          [(0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0)],
          [(0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0)],
          [(0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0)],
          [(0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0)],
          [(0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0)],
          [(0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0)],
          [(0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0)],
          [(0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0)],
          [(0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0)],
          [(0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0)],
          [(0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0)],
          [(0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0), (0, 255, 0)]]
```

Salvataggio di immagini: il modulo `png.py` scaricabile dalla libreria `PyPNG` e i metodi `from_array()` e `save()`

```
In [3]: import png
```

```
In [4]: def save(img, filename):  
        pngimg = png.from_array(img, 'RGB')  
        pngimg.save(filename)
```

```
In [5]: save(verde, 'verde.png')
```

la funzione Image() nella libreria IPython.display per mostrare immagini .png all'interno di IPython

```
In [6]: from IPython.display import Image  
        Image('verde.png')
```

Out[6]:



Creo una immagine 400x200 azzurra

```
In [7]: azzurro = create(400, 200, (0, 127, 255) )  
        save(azzurro, 'azzurro.png')  
        Image('azzurro.png')
```

Out[7]:



funzioni per calcolare ampiezza e lunghezza dell'immagine

```
In [8]: def righe(img) : return len(img)  
  
        def colonne(img) : return len(img[0])
```

```
In [9]: print("l'immagine 'azzurro' ha", righe(azzurro), 'righe e', colonne(azzurro), 'colonne')
l'immagine 'azzurro' ha 200 righe e 400 colonne
```

Una funzione per controllare se un dato pixel è all'interno dell'immagine o meno

```
In [10]: def inside(img, x, y):
         return 0 <= y < righe(img) and 0 <= x < colonne(img)
```

```
In [11]: inside(azzurro,100,100)
```

```
Out[11]: True
```

```
In [12]: inside(azzurro,500,100)
```

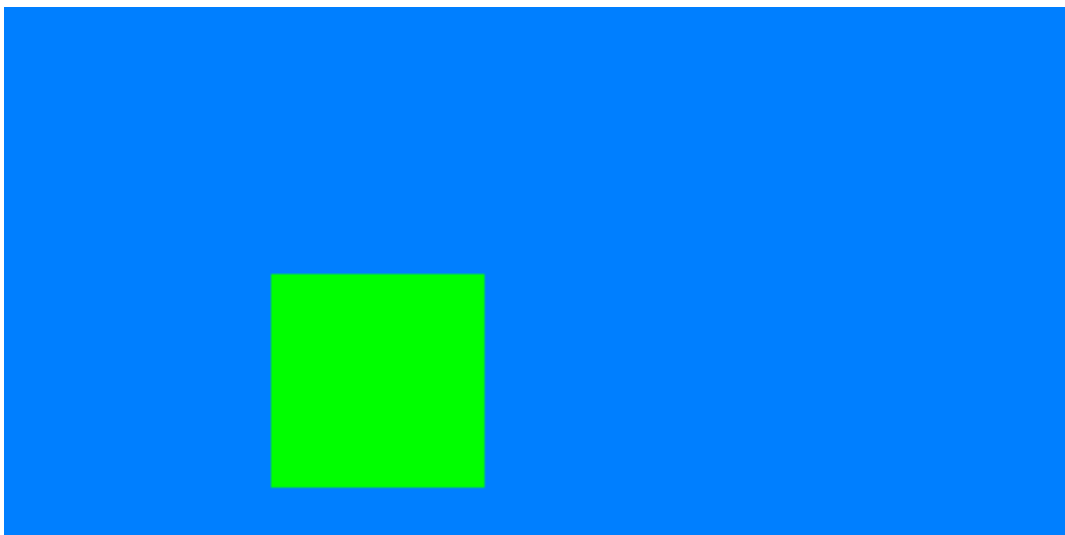
```
Out[12]: False
```

Una funzione per disegnare rettangoli colorati su di un'immagine:

```
In [13]: def draw_rect(img, x, y, w, h, c):
         '''disegna sull'immagine img un rettangolo di colore c, ampiezza w e altezza h a pa'''
         for px in range(x, x+w):
             for py in range(y, y+h):
                 if inside(img,px,py):
                     img[py][px] = c
```

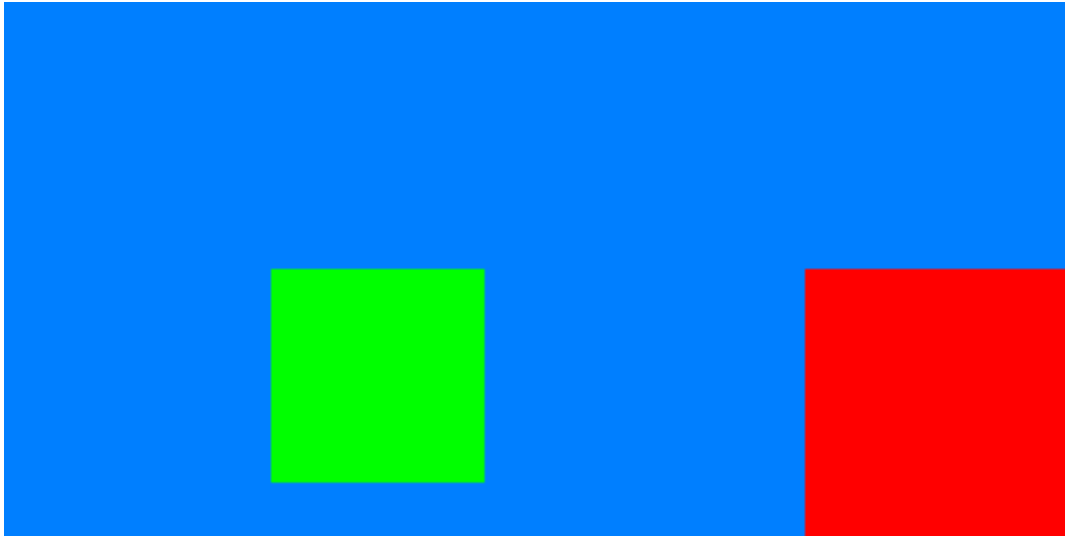
```
In [14]: draw_rect(azzurro, 100, 100, 80, 80, (0, 255, 0))
         save(azzurro, 'azzurro_verde.png')
         Image('azzurro_verde.png')
```

```
Out[14]:
```



```
In [15]: draw_rect(azzurro, 300, 100, 500, 500, (255, 0, 0))
         save(azzurro, 'azzurro_verde_rosso.png')
         Image('azzurro_verde_rosso.png')
```

Out[15]:



Posso in alternativa usare l'istruzione try/except per intercettare l'eccezione:

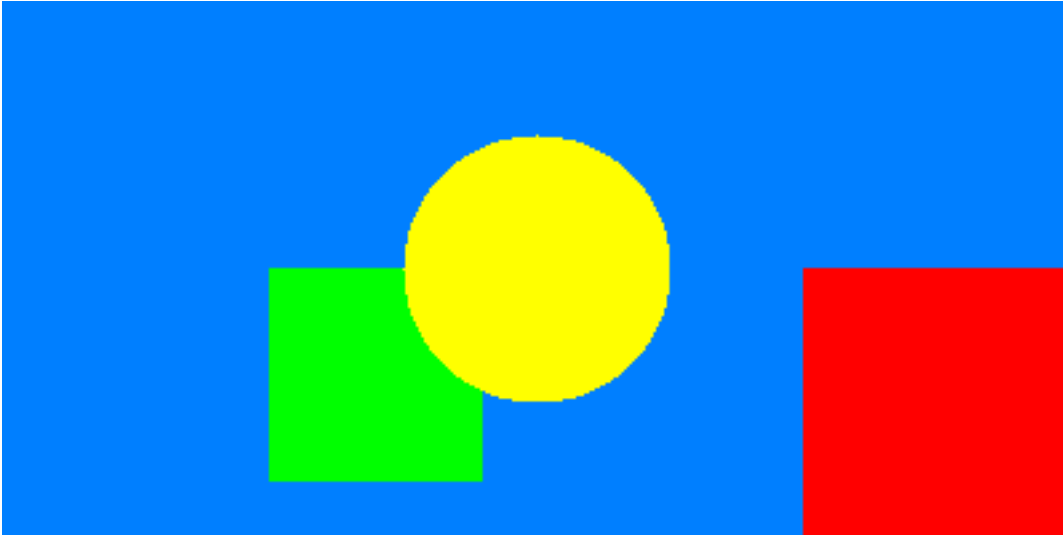
```
In [16]: def draw_rect(img, x, y, w, h, color):
         '''disegna sull'immagine img un rettangolo di colore color e ampiezza w e altezza h'''
         for px in range(x, x+w):
             for py in range(y, y+h):
                 try:
                     img[py][px] = color
                 except IndexError:
                     pass
```

Una funzione per disegnare cerchi di raggio r e colore c su un'immagine

```
In [17]: def draw_circle(img, xc, yc, r, c):
         r2 = r*r
         w = colonne(img)
         h = righe(img)
         for dx in range(-r, r):
             x = xc+dx
             for dy in range(-r, r):
                 y = yc+dy
                 if dx*dx+dy*dy <= r2 and inside(img, x,y):
                     img[y][x] = c

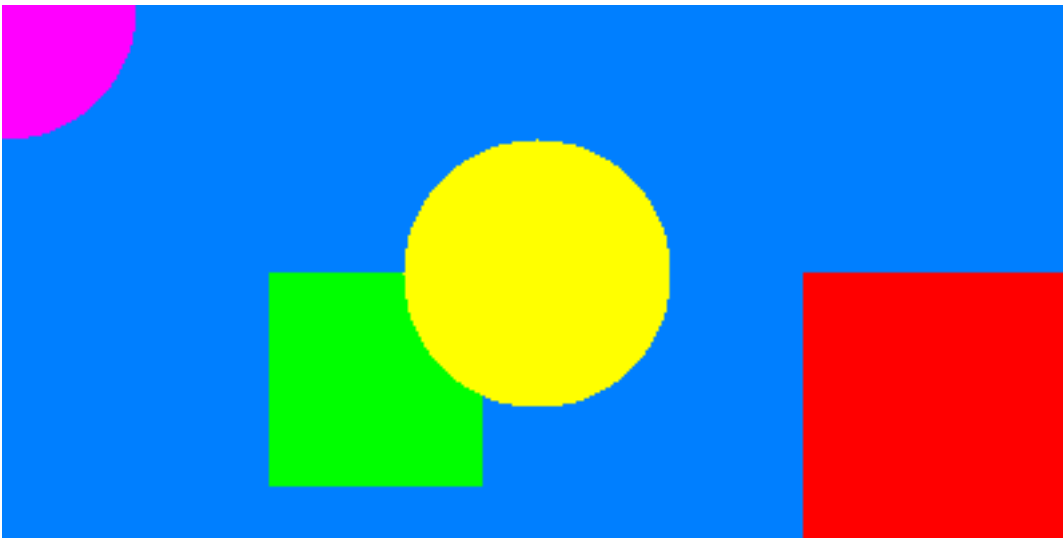
In [18]: draw_circle(azzurro, 200, 100, 50, (255,255,0) )
         save(azzurro, 'cerchio.png')
         Image('cerchio.png')
```

Out[18]:



```
In [19]: draw_circle(azzurro, 0, 0, 50, (255,0,255) )  
         save(azzurro, 'cerchio.png')  
         Image('cerchio.png')
```

Out[19]:



Una funzione per leggere una foto .png dal disco e convertirla in lista di liste: i metodi `reader()` e `asRGB8()`

```
In [20]: def load(filename):
    with open(filename, mode='rb') as f:
        reader = png.Reader(file=f)
        w, h, png_img, _ = reader.asRGB8()
        img = []
        for line in png_img:
            l = []
            for i in range(0, len(line), 3):
                l+=[(line[i], line[i+1], line[i+2])]
            img+=l
        return img
```

Per testare la funzione leggiamo e scriviamo lo stesso file

```
In [21]: prova = load('azzurro.png')
    save(prova, 'prova.png')
    Image('prova.png')
```

Out[21]:

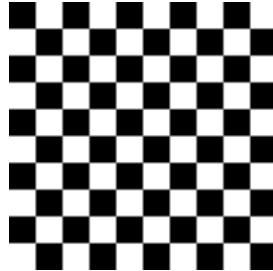


Creare scacchiera di lato $l*s$ con quadrati di lato s usando colori $c0$ e $c1$

```
In [22]: def scacchiera(l, s, c0, c1):
    ''' restituisce immagine scacchiera quadrata con l quadrati di lato s per riga us
    img=create(l*s,l*s,c0)
    for i in range(l):
        for j in range(l):
            if (i + j) % 2:
                draw_rect(img,i*s,j*s,s,s,c1)
    return img
```

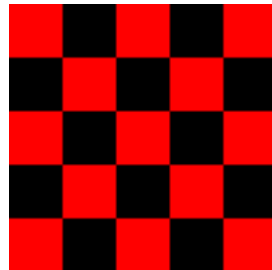
```
In [23]: sca= scacchiera(10,10,(0,0,0),(255,255,255))  
         save(sca, 'scacchiera.png')  
         Image('scacchiera.png')
```

Out[23]:



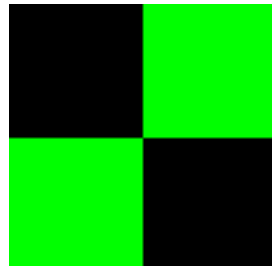
```
In [24]: sca= scacchiera(5,20,(255,0,0),(0,0,0))  
         save(sca, 'scacchiera.png')  
         Image('scacchiera.png')
```

Out[24]:



```
In [36]: sca= scacchiera(2,50,(0,0,0),(0,255,0))  
         save(sca, 'scacchiera.png')  
         Image('scacchiera.png')
```

Out[36]:



Vogliamo ruotare un'immagine attorno ad un suo asse: Ruotare rispetto all'asse verticale:

```
In [26]: foto=load('foto.png')  
         Image('foto.png')
```

Out[26]:



```
In [27]: def ruotaVerticale(img):  
         '''ritorna una nuova immagine che e' l'immagine specchio dell'originale,  
         ruota rispetto all'asse verticale'''  
         w, h = len(img[0]), len(img)  
         img1 = create(w,h)  
         for y in range(h):  
             for x in range(w):  
                 img1[y][w-1-x] = img[y][x]  
         return img1
```

```
In [28]: foto1=ruotaVerticale(foto)  
         save(foto1,'fotoRuotata.png')  
         Image('fotoRuotata.png')
```

Out[28]:



Ruotare rispetto all'asse orizzontale:

```
In [29]: def ruotaOrizzontale(img):
          '''ritorna una nuova immagine che e' la rotazione rispetto all'asse orizzontale'''
          w, h = len(img[0]), len(img)
          img1 = create(w,h)
          for y in range(h):
              for x in range(w):
                  img1[h-1-y][x] = img[y][x]
          return img1
```

```
In [30]: foto1=ruotaOrizzontale(foto)
          save(foto1,'fotoRuotata.png')
          Image('fotoRuotata.png')
```

Out[30]:



Ruotare l'immagine di 90 gradi rispetto all'asse in basso a sinistra

```
In [31]: def ruota90(img):  
    '''ritorna una nuova immagine che e' l'immagine ruotata di 90 gradi rispetto  
    all'angolo inferiore sinistro equivalente a scambiare righe con colonne'''  
    w, h = len(img[0]), len(img)  
    img1 = create(h,w)  
    for y in range(h):  
        for x in range(w):  
            img1[x][h-1-y] = img[y][x]  
    return img1  
  
In [32]: foto1=ruota90(foto)  
    save(foto1,'fotoRuotata.png')  
    Image('fotoRuotata.png')
```

Out[32]:



Vogliamo aggiungere una cornice di larghezza s e colore c ad un'immagine

```
In [33]: def bordo(img, s, c):  
    ''' restituisce immagine img con bordo di spessore s e colore c'''  
    w,h=len(img[0]),len(img)  
    img1 = create(w+s*2, h+s*2, c)  
    for y in range(h):  
        for x in range(w):  
            img1[y+s][x+s]=img[y][x]  
    return img1
```

```
In [34]: foto1=bordo(foto,50,(100,100,100))  
         save(foto1,'foto1.png')  
         Image('foto1.png')
```

Out[34]:

