Imágenes a color

...

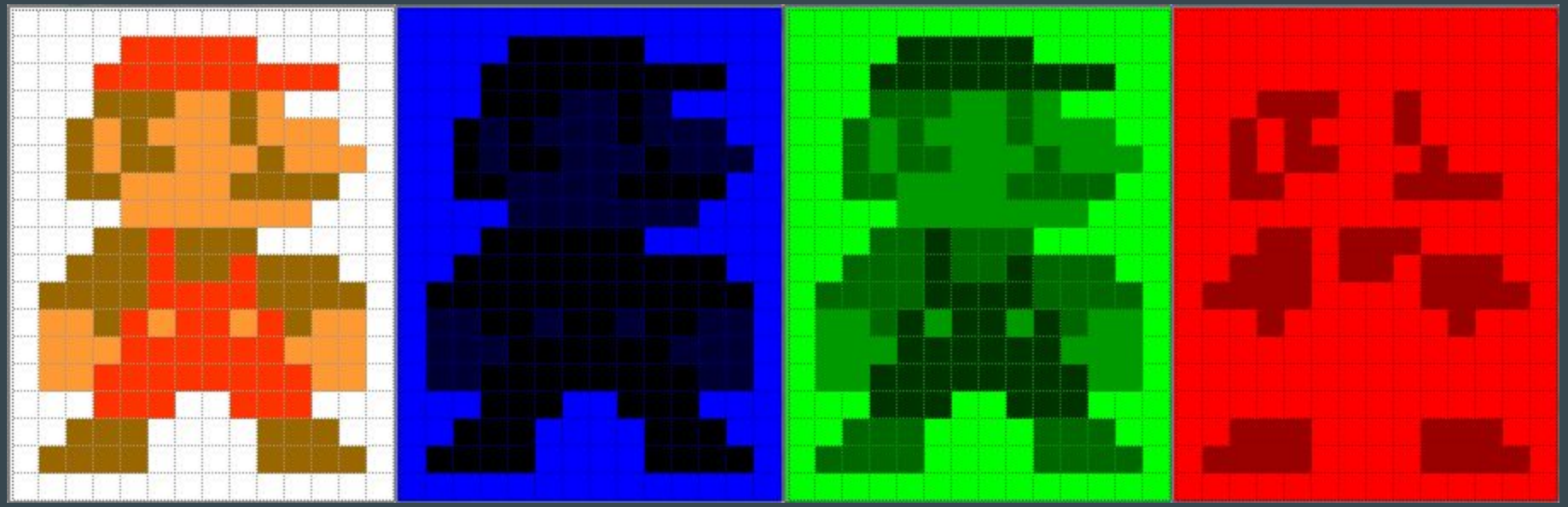
Python CV - Acceso al píxel

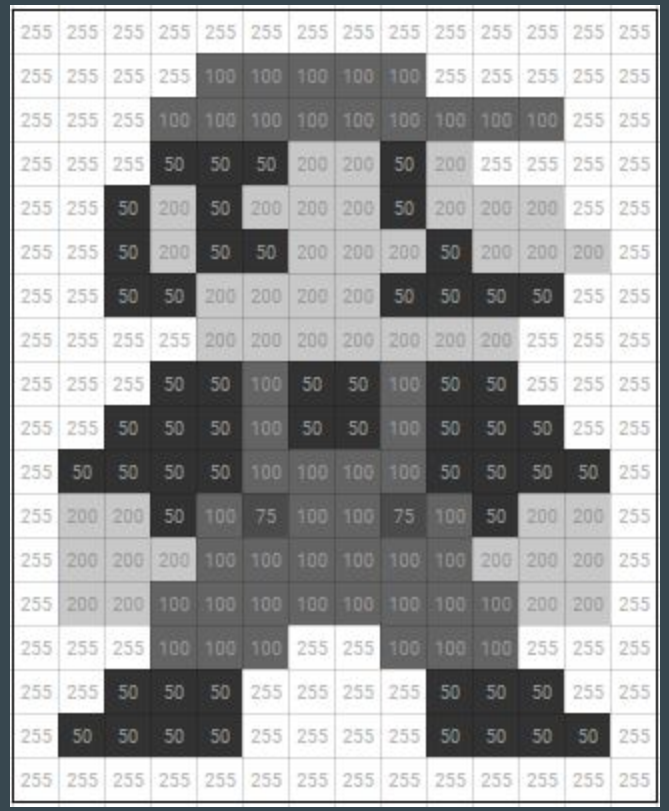
Observaciones:

- En python no es necesario definir el tipo de variable.
- Existe el retorno múltiple, por lo que funciones como .shape pueden retornar múltiples variables

```python
import numpy as np
import cv2
# Lectura de imagen. El parámetro 0 indica que es en escala de grises
entrada = cv2.imread("Lena.png",0)
# De la imagen leida se extrae el alto y el ancho
alto, ancho = entrada.shape
# Se crea una imagen vacia para almacenar el resultado
salida = np.zeros((alto,ancho), np.uint8)
# Permite crear un ciclo infinito. Similar a LOOP de Arduino
while(True):

    for i in range(0, alto): #Se recorre la imagen de 0 al alto
        for j in range(0, ancho): #Se recorre la imagen de 0 al ancho
            #Se lee la información del píxel en la posición i,j
            f = entrada.item(i,j)
            # Se resta 30 al valor original
            g = (f-30)
            #Se asigna el valor calculado, g, a la imagen nueva
            salida.itemset((i,j),g)

    #Se muestra la imagen de entrada en una ventana llamada original
    cv2.imshow("Original",entrada)
    cv2.imshow("Resultado",salida)

    #Se espera a que se presione la tecla Q para finalizar
    ch = 0xFF & cv2.waitKey()
    if ch == ord('q'):

        #Se almacena la imagen y se rompe el ciclo
        cv2.imwrite("Resultado.jpg",salida)
        break

#Se destruyden las ventanas creadas
cv2.destroyAllWindows()
```

Imágenes a color





R	G	B
255	205	148

R	G	B	X		0,95		=	R	G	B
255	205	148						242	195	141

R	G	B	X		0,5		+	R	G	B	X		0,5		=	R	G	B
255	205	148						213	166	189						234	186	169

R	G	B	+	R	G	B	=	R	G	B
75	200	50		180	25	25		255	225	75

Acceso a Píxel RGB

Observaciones:

- Canal 0 = b, canal 1 = g, canal 2 = r

```python
# -*- coding: utf-8 -*-
import numpy as np
import cv2
# Lectura de imagen.
entrada = cv2.imread("Lena.png")
# De la imagen leida se extrae el alto, el ancho y los canales
alto, ancho, canales = entrada.shape
# Se crea una imagen vacia para almacenar el resultado
salida = np.zeros((alto,ancho,3), np.uint8)
# Permite crear un ciclo infinito. Similar a LOOP de Arduino
while(True):
    for i in range(0, alto): #Se recorre la imagen de 0 al alto
        for j in range(0, ancho): #Se recorre la imagen de 0 al ancho
            #Se lee la información del píxel en la posición i,j de cada canal
            b0 = entrada.item(i,j,0)
            g0 = entrada.item(i,j,1)
            r0 = entrada.item(i,j,2)
            # Se incrementa el canal azul un 10% y se diminuye el rojo 50%
            b1 = b0*1.1
            g1 = g0
            r1 = r0*0.5
            #Se asigna cada valor calculado a la imagen nueva
            salida.itemset((i,j,0),b1)
            salida.itemset((i,j,1),g1)
            salida.itemset((i,j,2),r1)
    #Se muestra la imagen de entrada en una ventana llamada original
    cv2.imshow("Original",entrada)
    cv2.imshow("Resultado",salida)
    #Se espera a que se presione la tecla Q para finalizar
    ch = 0xFF & cv2.waitKey()
    if ch == ord('q'):
        #Se almacena la imagen y se rompe el ciclo
        cv2.imwrite("Resultado.jpg",salida)
        break
#Se destruyden las ventanas creadas
cv2.destroyAllWindows()
```

Desafíos

1. Acceder a cada píxel de una imagen RGB, sumar el valor de cada píxel de cada canal y dividirlo entre 3 (promedio), el resultado almacenarlo en una imagen vacía de un único canal. Concluir del resultado
2. Implementar el desafío 3 (de la clase anterior) en una imagen en escala de grises de bajo contraste
3. Implementar el desafío 3 (de la clase anterior) en una imagen a color (a cada canal)
4. Implementar el desafío 3 (de la clase anterior) en una imagen a color con base en el promedio

Nota: el máximo y el mínimo deben ser calculados con condicionales y ciclos no con funciones propias de Python, Numpy u OpenCV. Las formas adecuadas serán discutidas en la próxima sesión

Ayuda - Ejemplo para mínimo en un arreglo

```
x = [5,3,9,2]
minimo = 1000

for i in range(0,4):
    if(x[i] < minimo):
        minimo = x[i]
```

Ejemplo resultado desafío 4