# Procesamiento de imágenes #05

## Matrices en Python

Dra. C. Miriela Escobedo Nicot

Cuando cargamos una imagen obtenemos la representación matricial de esta.

In [8]:
```
from pylab import *
from skimage.color import rgb2gray
```

In [3]:
```
A = imread('dataset/mona_lisa.jpg')
```

In [4]:
```
A.shape
```

Out[4]: (197, 197, 3)

In [6]:
```
A.dtype
```

Out[6]: dtype('uint8')

In [9]:
```
A = rgb2gray(A)
```

In [10]:
```
A.shape
```

Out[10]: (197, 197)

In [11]:
```
A.dtype
```

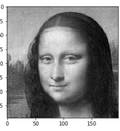
Out[11]: dtype('float64')

In [12]:
```
A
```

Out[12]:
```
array([[0.51197098, 0.51140549, 0.49964078, ..., 0.47782353, 0.49853255,
        0.51421882],
       [0.51729922, 0.51337765, 0.4937698 , ..., 0.48284627, 0.50637569,
        0.52206196],
       [0.54615686, 0.54223529, 0.50694118, ..., 0.47108157, 0.49068941,
        0.49908314],
       ...,
       [0.44683804, 0.41938706, 0.36391961, ..., 0.06802706, 0.05626235,
        0.0758702 ],
       [0.42544824, 0.49272824, 0.4848851 , ..., 0.05626235, 0.03665451,
        0.04841922],
       [0.29404392, 0.42795059, 0.46324471, ..., 0.06886039, 0.04140941,
        0.04952255]])
```

In [16]:
```
rcParams['image.cmap'] = 'gray' # Establecer parametro cmap por defecto a gray
imshow(A)
```

Out[16]:
```
<matplotlib.image.AxesImage at 0x9556dbcb38>
```



Con el código anterior hemos cargado una imagen y la hemos llevado a escala de grises para utilizar su matriz como ejemplo durante esta clase.

Para seleccionar los elementos en una matriz se necesitan dos índices: uno para establecer la localización en la fila y el otro para la columna correspondiente. En nuestro caso como trabajamos con imágenes cada elemento de la matriz le corresponde a un pixel.

#### Obtener el valor de un pixel

In [19]:
```
A[2, 3]
```

Out[19]: 0.5187058823529411

#### Obtener una columna específica con todas las filas

In [26]:
```
A[:, 3]
```

Out[26]:
```
array([0.53660941, 0.52457686, 0.51870588, 0.52544784, 0.53329098,
       0.54615686, 0.54754078, 0.54053098, 0.53436941, 0.52876627,
       0.51308   , 0.50523686, 0.51165843, 0.53126627, 0.54527098,
       0.54527098, 0.54806157, 0.55198314, 0.56234902, 0.57019216,
       0.56374784, 0.55254863, 0.54919255, 0.55087412, 0.54695255,
       0.55087412, 0.56263882, 0.57974667, 0.58366824, 0.58366824,
       0.5853498 , 0.59124353, 0.58366824, 0.57554235, 0.57440353,
       0.59025137, 0.60121235, 0.60155686, 0.60657961, 0.61834431,
       0.60719765, 0.60969765, 0.60185451, 0.60604392, 0.61332157,
       0.60657961, 0.59256   , 0.60040314, 0.60626706, 0.61803176,
       0.6236349 , 0.6118702 , 0.60794863, 0.63147804, 0.63539961,
       0.60402706, 0.59392902, 0.59618392, 0.59674941, 0.61440784,
       0.62533922, 0.6141   , 0.5970698 , 0.60883451, 0.62001098,
       0.56903059, 0.59873647, 0.60547843, 0.56682824, 0.61331373,
       0.65868314, 0.59788745, 0.63123216, 0.60218667, 0.63631451,
       0.62541294, 0.64189922, 0.6461298 , 0.60633373, 0.6813792,
       0.65166627, 0.67883451, 0.62868745, 0.60123647, 0.5855651 ,
       0.53235961, 0.58403843, 0.46965922, 0.45252941, 0.41863412,
       0.40689922, 0.44080157, 0.48842588, 0.49961725, 0.46178549,
       0.41864824, 0.35649922, 0.34473451, 0.32904824, 0.33465137,
       0.34306   , 0.3097302 , 0.28676627, 0.30301804, 0.30632941,
       0.26040157, 0.28814235, 0.26884706, 0.26660706, 0.29770471,
       0.25764078, 0.28676627, 0.31478275, 0.30301804, 0.26548392,
       0.27474863, 0.31956745, 0.33133216, 0.29408863, 0.23918667,
       0.23778   , 0.3464451 , 0.29516706, 0.2853298 , 0.36542078,
       0.33770196, 0.29487725, 0.31339098, 0.37278784, 0.24476784,
       0.31814667, 0.36463216, 0.3604349 , 0.27106431, 0.25145647,
       0.25536314, 0.34666824, 0.32511098, 0.27469608, 0.28535961,
       0.2859251 , 0.30075529, 0.25172431, 0.34051373, 0.25952275,
       0.29873843, 0.32957529, 0.28113255, 0.2520298 , 0.20134706,
       0.30109059, 0.26719529, 0.25738784, 0.24337608, 0.29799529,
       0.2996698 , 0.33020902, 0.29604588, 0.2352651 , 0.27504627,
       0.29995176, 0.22936353, 0.22152039, 0.21089451, 0.32069843,
       0.28540431, 0.28540431, 0.29241412, 0.29219176, 0.41375255,
       0.25267765, 0.2593749 , 0.35431882, 0.33388549, 0.3767549 ,
       0.44316863, 0.35297255, 0.3563863, 0.43978275, 0.41204902,
       0.47030706, 0.36612902, 0.4224298 , 0.32831216, 0.46309686,
       0.41211647, 0.45271608, 0.38015569, 0.40311961, 0.41712431,
       0.35045765, 0.43281059, 0.48127608, 0.41938706, 0.38352745,
       0.44453059, 0.44138196])
```

#### Obtener una columna específica con los valores de las tres primeras filas

In [30]:
```
A[0:3, 3]
```

Out[30]: array([0.53660941, 0.52457686, 0.51870588])

#### De igual modo se puede aplicar para obtener una fila en un rango determinado de columnas

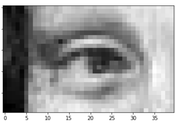
In [32]:
```
A[3,0:4]
```

Out[32]: array([0.55233333, 0.5562549 , 0.52096078, 0.52544784])

#### Utilizando este sistema de indexado podemos extrar regiones determinadas de la imagen (matriz)

In [41]:
```
imshow(A[80:105, 50:90])
```

Out[41]:
```
<matplotlib.image.AxesImage at 0x9556f12a90>
```



Los indices que se utilizan en la matriz pueden ser:

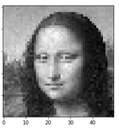
- Un número entero comprendido entre $(0, n-1)$ done $n$ es el tamaño de la fila o columna
- Dos números separados por : en la forma $x:y$ (rango) donde $x$ representa el indice inicial y $y$ el indice final.
- Tres números separados por : en la forma $x:y:z$ (rango) donde $x$ representa el indice inicial y $y$ el indice final, y $z$ representa el "salto" o sea cuánto vamos sumando a $x$ hasta obtener $y$ o un valor mayor que $y$.

#### En el siguiente ejemplo realizamos un recorrido por toda la imagen pero con un "salto" de 4

Podemos ver como la imagen pierde resolución ya que se han descartado algunas filas y columnas

In [50]:
```
imshow(A[::4,::4])
```

Out[50]:
```
<matplotlib.image.AxesImage at 0x9557ae4828>
```



#### También podemos utilizar saltos negativos

In [56]:
```
imshow(A[::-1, :])
```

Out[56]:
```
<matplotlib.image.AxesImage at 0x955Bbc52e8>
```

Esto recorre la imagen en el eje $y$ a la inversa, la mimsa operación se puede realizar en el eje $x$

In [57]:
```
imshow(A[:, ::-1])
```

Out[57]:
```
<matplotlib.image.AxesImage at 0x9558bfa7f0>
```

#### Además podemos utilizar indexado para obtener histogramas sobre una fila o columna determinada

In [59]:
```
plot(A[100,:])
```

Out[59]:
```
[<matplotlib.lines.Line2D at 0x9558cbE898>]
```