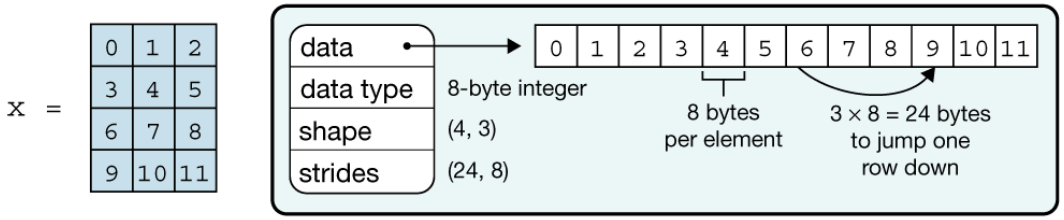
Institución Universitaria
Reacreditada en Alta Calidad

Repaso Numpy
Operaciones con arreglos

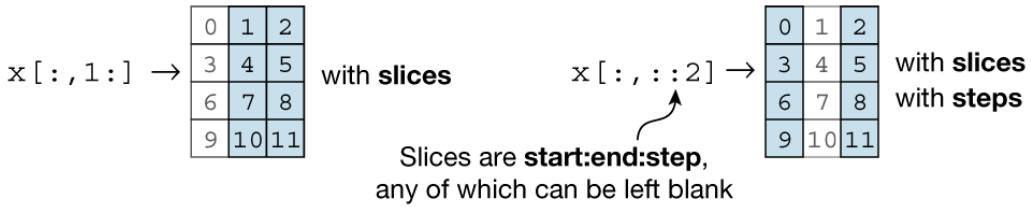
Optimización

Resumen de Numpy

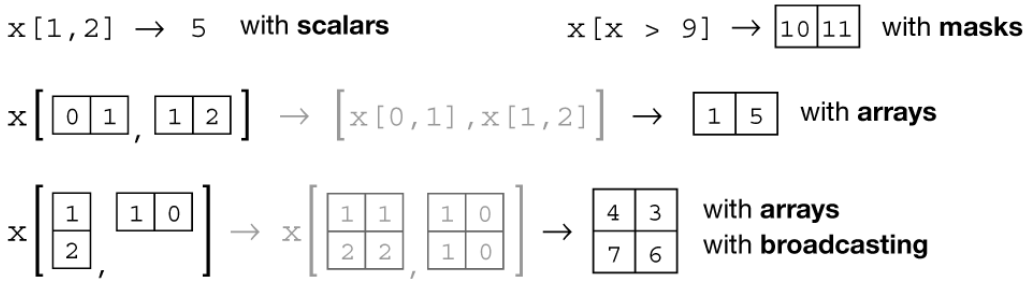
a Data structure



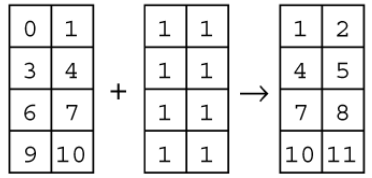
b Indexing (view)



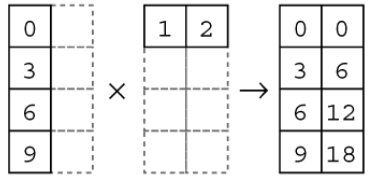
c Indexing (copy)



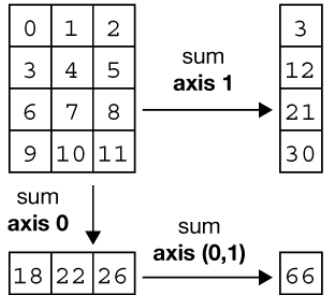
d Vectorization



e Broadcasting



f Reduction



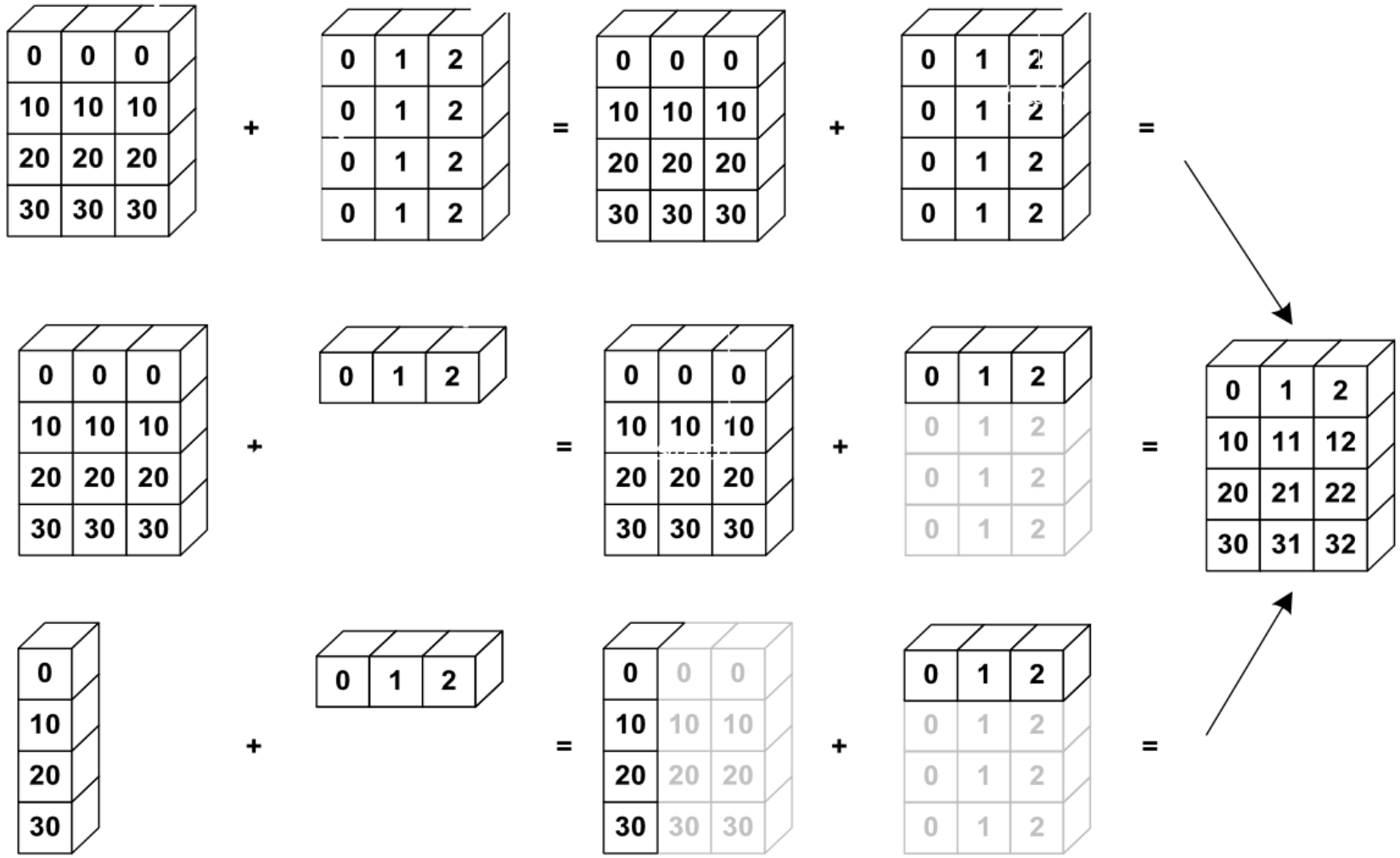
g Example

```
In [1]: import numpy as np

In [2]: x = np.arange(12)

In [3]: x = x.reshape(4, 3)

In [4]: x
Out[4]:
array([[ 0,  1,  2],
       [ 3,  4,  5],
       [ 6,  7,  8],
       [ 9, 10, 11]])

In [5]: np.mean(x, axis=0)
Out[5]: array([4.5, 5.5, 6.5])

In [6]: x = x - np.mean(x, axis=0)

In [7]: x
Out[7]:
array([[-4.5, -4.5, -4.5],
       [-1.5, -1.5, -1.5],
       [ 1.5,  1.5,  1.5],
       [ 4.5,  4.5,  4.5]])
```

https://www.nature.com/articles/s41586-020-2649-2

Operaciones con arreglos

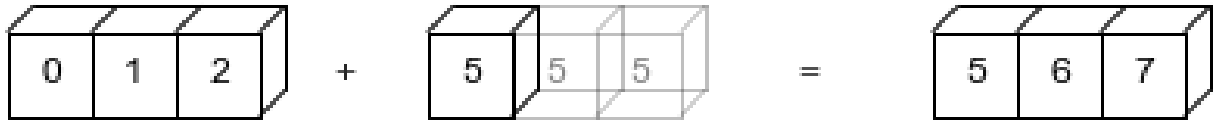
Toda las operaciones con arreglos son element-wise (elemento a elemento), broadcasting



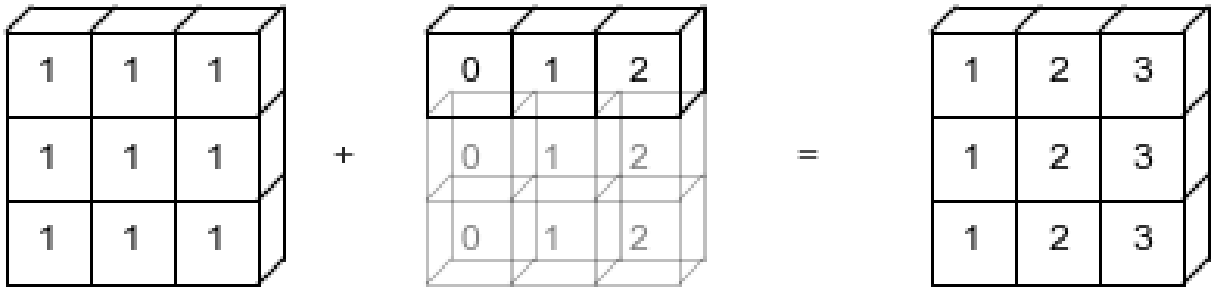
https://scipy-lectures.org/intro/numpy/operations.html

Broadcasting

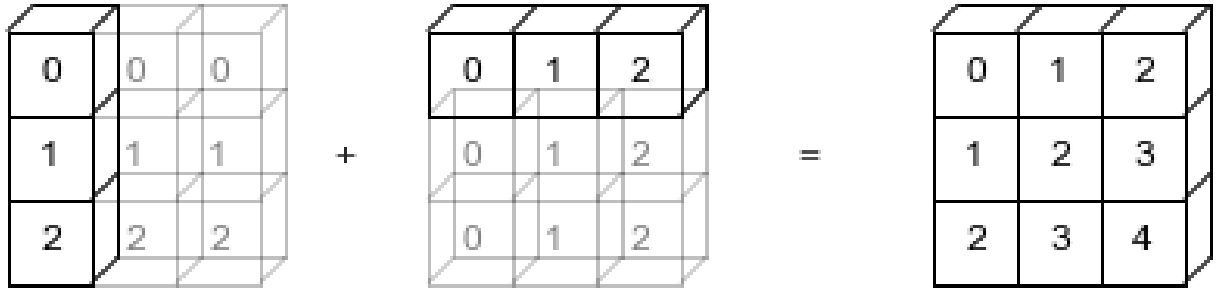
$\texttt{np.arange}(3)+5$



$\texttt{np.ones}((3,3))+\texttt{np.arange}(3)$



$\texttt{np.arange}(3).\texttt{reshape}((3,1))+\texttt{np.arange}(3)$



https://www.freecodecamp.org/news/exploratory-data-analysis-with-numpy-pandas-matplotlib-seaborn/

Slicing – Cortes o selección de submatrices

Índices matriz 4x3
(fila, columna)

(0,0)	(0,1)	(0,2)
(1,0)	(1,1)	(1,2)
(2,0)	(2,1)	(2,2)
(3,0)	(3,1)	(3,2)

Valores X

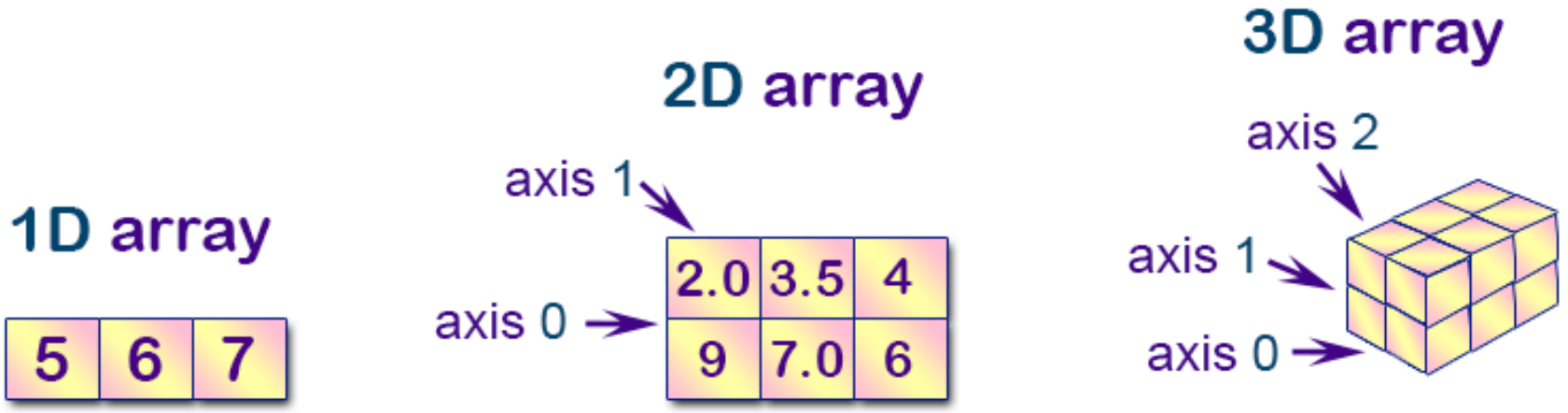
1	2	3
4	5	6
7	8	9
10	11	12

X[0,1] = 2
X[1,:] = [4,5,6]
X[1:,1:] = [[5,6],[8,9],[11,12]]
X[::2,:] = [[1,2,3],[7,8,9]]
X[[0,1,2],[0,1,2]] = [1,5,9]

np.min(X) → 1
np.argmin(X) → 0
np.unravel_index(x.argmin(), x.shape)→ [0,0]

https://www.freecodecamp.org/news/exploratory-data-analysis-with-numpy-pandas-matplotlib-seaborn/

Creación de arreglos aleatorias

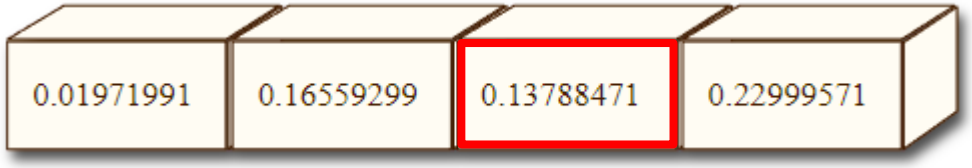


https://www.w3resource.com/

Creación de arreglos aleatorias

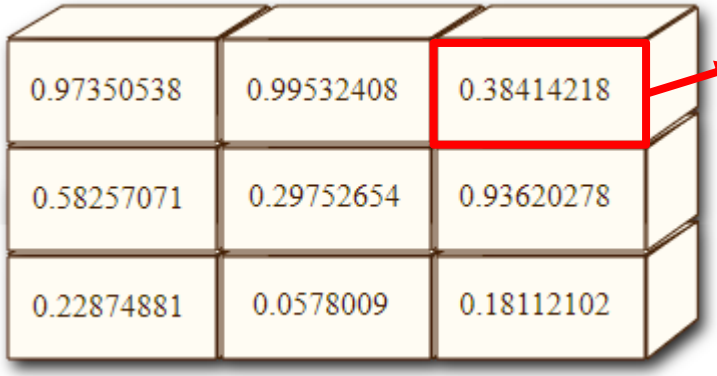
Forma general:
np.random.rand(d1,d2,…,dn)
La función rand genera valores flotantes entre 0 y 1.

x1 =np.random.rand(4)



x1[2]

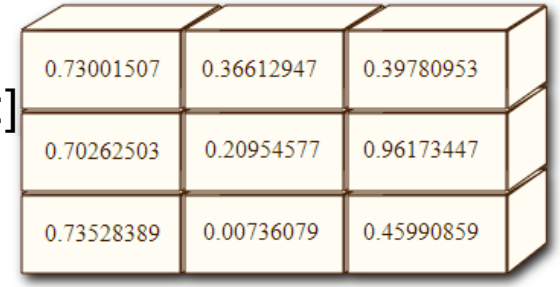
x2 = np.random.rand(3,3)



x2[0,2]

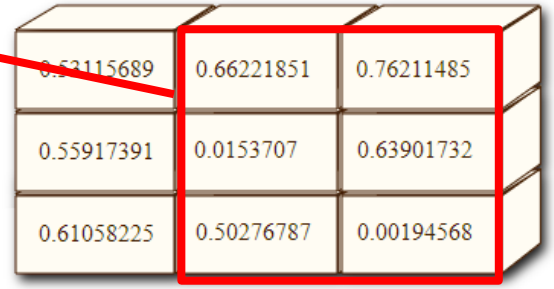
x3 =np.random.rand(3,3,3)

x3[0,:,:]



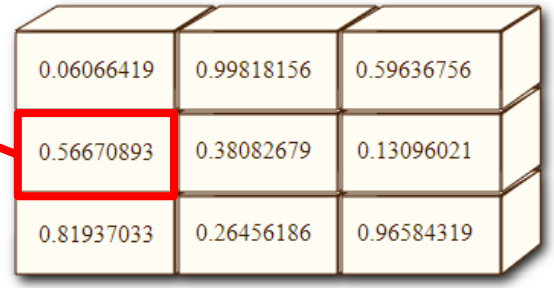
?

x3[1,:,:]



x3[2,1,0]

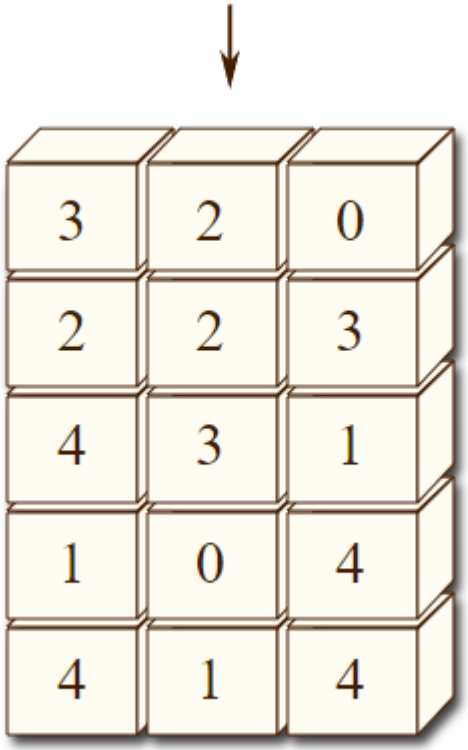
x3[2,:,:]



https://www.w3resource.com/

Creación de arreglos enteros aleatorios

np.random.randint(5, size=(5,3))



https://www.w3resource.com/

Comparaciones relacionales

```
import numpy as np
a = np.array([1, 3, 7, 9, 10, 13, 14, 17,
29]) print("Arreglo Original:",a)

result = np.where(np.logical_and(a>=7, a<=20))
print("\nElementos en el rango: posicion del indice")
print(result)

result2 = np.where((a>=7) & (a<=20))
print("\nElementos en el rango: posicion del indice ")
print(result2)
```

(a>=7)

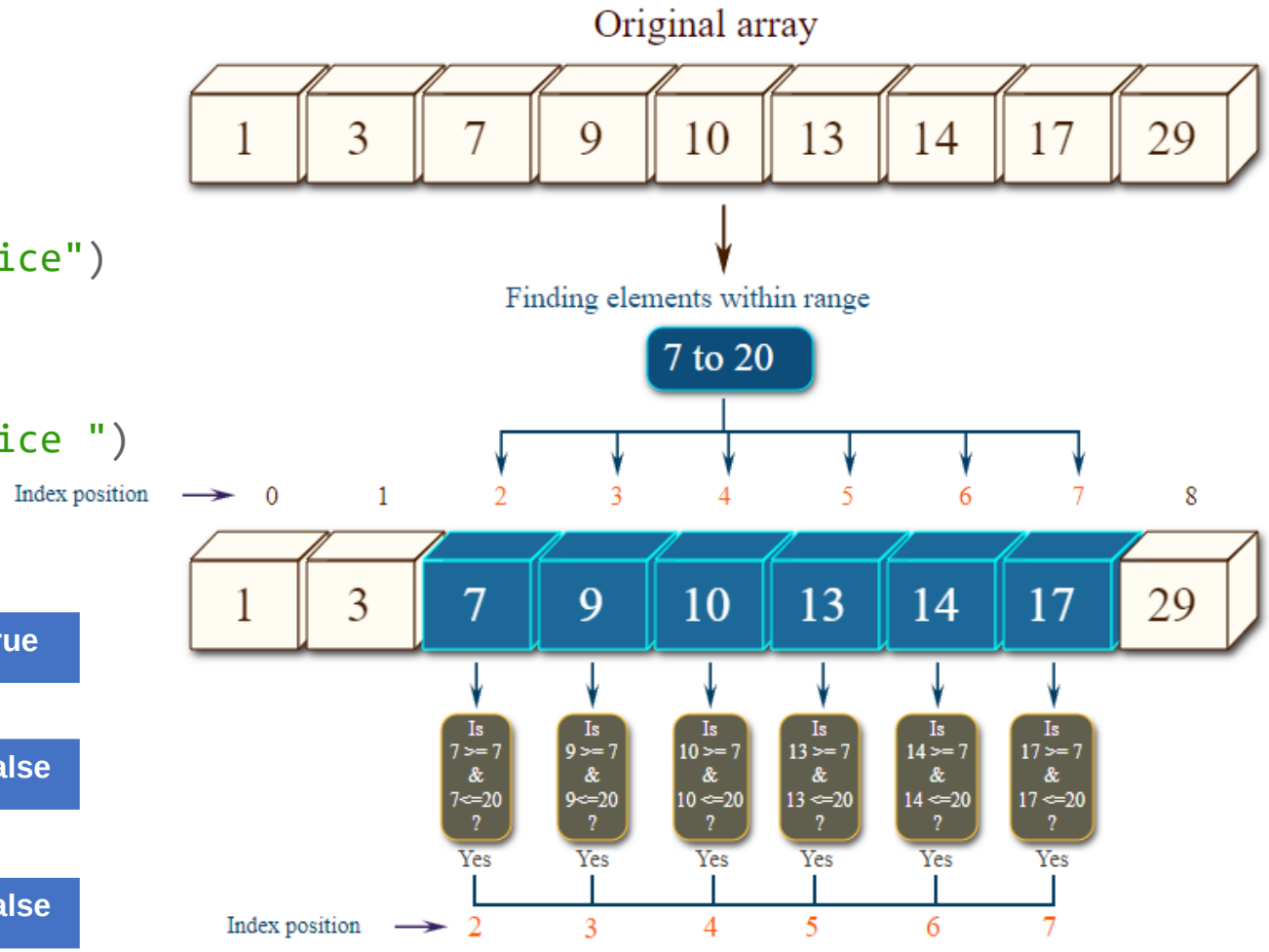
False	False	True	True	True	True	True	True	True

(a<=20)

True	True	True	True	True	True	True	True	False

&

False	False	True	True	True	True	True	True	False



https://www.w3resource.com/

Institución Universitaria
Reacreditada en Alta Calidad

¡Gracias!

Somos Innovación Tecnológica con Sentido Humano

Alcaldía de Medellín