CC - AI

Programación para la Computación Científica - IA

Python for Programmers

Numpy II

Universidad Sergio Arboleda

Prof. John Corredor

```
#!/usr/bin/env/python

import sys

from datetime import datetime

import numpy as np
```

```
def numpysum(n):
    a = np.arange(n) ** 2
    b = np.arange(n) ** 3
    c = a + b
    return c
```

```
def pythonsum(n):
    a = range(n)
    b = range(n)
    c = []
    for i in range(len(a)):
        a[i] = i ** 2
        b[i] = i ** 3
        c.append(a[i] + b[i])
    return c
```

```
size = int(sys.argv[1])
start = datetime.now()
c = pythonsum(size)
delta = datetime.now() - start
print "The last 2 elements of the sum", c[-2:]
print "PythonSum elapsed time in microseconds", delta.microseconds
start = datetime.now()
c = numpysum(size)
delta = datetime.now() - start
print "The last 2 elements of the sum", c[-2:]
print "NumPySum elapsed time in microseconds", delta.microseconds
```

- np.full((3x3),True,dtype=bool)
- np.ones((3,3),dtype=bool)
- arr = np.array([0,1,2,3,4,5,6,7,8,9])
    - #> arr([1.3.5.7.9])
    - #> arr[arr%2 ==1]
- np.arange(10)
    - a.reshape(2, -1) # Setting to -1 automatically decides the number of cols
- a = np.array([2, 6, 1, 9, 10, 3, 27])
    - #> array([6, 9, 10])
        - b[(b>=5)&(b<=10)]

- arr = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
  - #> array([0, -1, 2, -1, 4, -1, 6, -1, 8, -1])
    - arr[arr % 2 == 1] = -1
- a = np.array([1,2,3])
  - #> array([1, 1, 1, 2, 2, 2, 3, 3, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3])
    - np.r_[np.repeat(a, 3), np.tile(a, 3)]
- a = np.arange(10).reshape(2,-1)   b = np.repeat(1, 10).reshape(2,-1)
  - #> array([[0, 1, 2, 3, 4, 1, 1, 1, 1, 1],
    #>   [5, 6, 7, 8, 9, 1, 1, 1, 1, 1]])
    - np.concatenate([a, b], axis=1)
    - np.hstack([a, b])
    - np.c_[a, b]

- a = np.array([1,2,3,2,3,4,3,4,5,6]) b = np.array([7,2,10,2,7,4,9,4,9,8])
    - #> array([2, 4])
        - np.intersect1d(a,b)
- a = np.array([1,2,3,4,5]) b = np.array([5,6,7,8,9])
    - #> array([1,2,3,4])
        - np.setdiff1d(a,b)
- a = np.array([1,2,3,2,3,4,3,4,5,6]) b = np.array([7,2,10,2,7,4,9,4,9,8])
    - #> (array([1, 3, 5, 7]),)
        - np.where(a == b)

-
- Taller 101-numpy-exercises-python
- Entregar en PDF

```
import numpy as np
import os
x = np.arange(10)
y = np.arange(11, 20)
print("Original arrays:")
print(x)
print(y)
np.savez('temp_arra.npz', x=x, y=y)
print("Load arrays from the 'temp_arra.npz' file:")
with np.load('temp_arra.npz') as data:
    x2 = data['x']
    y2 = data['y']
    print(x2)
    print(y2)
```

Challenge 01

1. Escriba un programa NumPy para crear una matriz de 3x3 con valores que oscilen entre 2 y 10.
2. Escriba un programa NumPy para invertir una matriz (el primer elemento se convierte en el último)
3. Escriba un programa NumPy para convertir los valores de grados centígrados en grados Fahrenheit. Los valores centígrados se almacenan en una matriz NumPy n=90 (random)

Challenge 02

1. Escriba un programa NumPy para encontrar el número de elementos de una matriz, la longitud de un elemento de la matriz en bytes y el total de bytes consumidos por los elementos.
   a. Salida:
      i. Tamaño de la matriz: 3
      ii. Longitud de un elemento de matriz en bytes: 8
      iii. Total de bytes consumidos por los elementos de la matriz: 24

References

- Kernighan, Brian W., and Dennis M. Ritchie. The C Programming Language. Vol. 2. Englewood Cliffs: prentice-Hall, 1988.
- Silberschatz, Abraham, Peter B. Galvin, and Greg Gagne. Operating System Concepts. Vol. 8. Wiley, 2013.
- https://planningtank.com/computer-applications/data-processing-cycle
- https://www.talend.com/resources/what-is-data-processing/
- https://peda.net/kenya/ass/subjects2/computer-studies/form-3/data-processing/dpc2
- https://www.academia.edu/38210518/What_is_Data_Processing
- https://www.studocu.com/en/document/polytechnic-university-of-the-philippines/information-and-communication-technology/lecture-notes/data-processing-lectures-in-data-processing/3167716/view
- http://download.nos.org/srsec330/330L2.pdf