

Práctica 1

1. Objetivo

Conocer el proceso general de la simulación y familiarizarse con el lenguaje Python.

2. Simulación

La simulación consiste en imitar el funcionamiento de un proceso o sistema real, utilizando computadoras, para generar un conjunto de resultados "artificiales" que sirven para hacer inferencias o predicciones sobre el comportamiento del proceso o sistema real.

3. Python

Python es un lenguaje de programación multiparadigma, multiplataforma, interpretado, comúnmente utilizado para el análisis de grandes cantidades de datos.

Anaconda es una "suite" de herramientas para el desarrollo de proyectos de Ciencia de Datos y Aprendizaje Automático con Python y R.

Una vez instalado Anaconda (en Linux), puede acceder a sus diferentes herramientas ejecutando el siguiente comando en la terminal:

```
$ anaconda-navigator
```

También puede acceder directamente a JupyterLab o Spyder respectivamente con los comandos:

```
$ jupyter-lab
```

```
$ spyder
```

Es recomendable ejecutar dichos comandos en el directorio donde desee que queden sus archivos.

JupyterLab es una interfaz web y Spyder es un IDE, ambos para escribir e interpretar código Python. Otra forma de escribir e interpretar código Python es de manera interactiva en la línea de comandos después de ejecutar el comando "python" o haciendo un script en un archivo con extensión .py y escribiendo en la terminal "python nombreArchivo.py".

4. Ejercicio

Ejecutar el siguiente código en un JupyterNotebook (Shift+Enter).

```
print("Hola Mundo!")  
4+10*5
```

```
a=5  
print(a)  
a=10+7.5  
print(a)  
a = "Cadena de caracteres"  
print(a)
```

```
a,A = 10, (20+3j)  
print(a,A)  
type(A)
```

```
type(True)
```

```
lista = [0,1,"dos",3,"cuatro cinco"]  
lista[2] = 2  
print(lista)
```

```
tupla = (0,1,"dos","tres",[10,11,12])  
tupla[4][1] ='x'  
print(tupla)
```

```
list(range(4))
```

```
print(10/3)  
print(10//3)  
print(10**3)
```

```

l1 = [0,1,2,3]
l2 = [4,5,6,7]
print(l1+l2)
print(2*l1)
print(l1[2:4])

```

```

x = int(input("x= "))
if x>10:
    print("x mayor que 10")
elif x>0:
    print("x mayor que 0 y menor o igual que 10")
else:
    print("x menor o igual que 0")

```

```

y = int(input("y="))
r = "par" if y%2 == 0 else "impar"
print(y, " es ", r)

```

```

contactos = ["Ana", "Pedro", "Blanca", "Juan"]
contacto = "Pedro"
for x in contactos:
    if x==contacto:
        print("Tienes un contacto llamado ", contacto)
        break
else:
    print("No tienes un contacto llamado ", contacto)

```

```

l = list(range(5))
print(l)
total = 0;
i = 0;
while i < len(l):
    total += l[i]
    i+=1
else:
    print("Suma terminada con éxito")

print("Total = ", total)

```

```

-----

def esPrimo(numero):
    if numero <=1:
        return False
    for i in range(2,numero):
        if numero%i == 0:
            return False
    return True

```

```

esPrimo(10)

```

```

-----

import time
import numpy
def sumaListasNormal(l):
    X = list(range(l))
    Y = list(range(l))
    t1 = time.time()
    Z = [0]*l
    for i in range(l):
        Z[i] = X[i] + Y[i]
    return time.time()-t1

```

```

def sumaListasNumpy(l):
    X = numpy.arange(l)
    Y = numpy.arange(l)
    t1 = time.time()
    Z = X+Y
    return time.time()-t1

```

```

l = 1000
print("tiempo con sumaListasNormal = ", sumaListasNormal(l))
print("tiempo con sumaListasNumpy = ", sumaListasNumpy(l))

```

```

-----

import numpy as np
X = np.ones((2,2))
print(1+2.5*X)

```

```

-----

Y = [[0]*2, [0]*2]
X = [[1]*2, [1]*2]
print(Y)
print(X)

```

```

for i in range(2):
    for j in range(2):
        Y[i][j] = 1+2.5*X[i][j]
print(Y)

```

```

import numpy as np
X = np.arange(4).reshape(2,2)
Y = np.arange(4).reshape(2,2)
Z = X+Y
print(Z)
print(Z*Z)

```

```

import numpy as np
import random
for i in range(10):
    num = np.random.uniform(0.0,1.0)
    if num < 0.5:
        print("cara")
    else:
        print("cruz")

```

```

import matplotlib.pyplot as plt

plt.xlabel("Número de juegos")
plt.ylabel("Número promedio de lanzamientos")
plt.plot([10,20,40,80,160,320,640,1280],[4,19,8,16,10,13,12,12])
plt.show()

```

5. Problema

Considere un juego de lanzamiento de monedas con las siguientes reglas:

1. En cada juego se lanza una moneda no alterada en repetidas ocasiones, hasta que la diferencia entre el número de caras y cruces que han aparecido sea tres.
2. Si desea participar debe pagar un peso cada vez que se lanza la moneda y no puede abandonar el juego hasta que éste termine.
3. Al final de cada juego usted recibe 8 pesos independientemente del número de lanzamientos.

Simular el juego utilizando el lenguaje de programación Python, ejecutarlo varias veces y obtener:

- (i) El número promedio de lanzamientos por juego.
- (ii) El promedio de ganancias o pérdidas por juego.

Para cada punto (i) y (ii) hacer una gráfica con los resultados para diferente número de juegos (10, 20, 40, 80, 160, 320, 640, 1280, 2560, 3200,...)

¿Cuántas ejecuciones de la simulación (número de juegos) considera usted suficientes para decir que los resultados son aceptables y por qué?