PYTHON

**String**

secuencia de caracteres definidos entre ‘

(‘hola mundo’)

“hola\n mundo” -> Salto de línea

“””hola

Mundo””” -> Salto de línea

NUMEROPI = 3.1416 -> Constante

**Imprimir en pantalla**

Print(‘hola mundo’)

**Comentarios de código**

# Soy un comentario

‘’’

Texto que no se va a interpretar

‘’’

**Variable**

Nombre = ‘Diego’

Altura = ‘176cm’

Print(‘nombre’) -> Estará imprimiendo “Diego”

**Concatenación**

Unión de varias variables

Print(f”(nombre) – (altura)”) -> Estará imprimiendo Diego – 176cm

Print(nombre + “ – “ + altura)

**Entrada de datos**

Edad = input (‘Qué edad tienes?: ’) -> Pregunta y te lo guarda en ‘edad’

Print(‘Tu edad es de’ edad) -> Tu edad es de (lo que hayas puesto en ‘edad’)

**Condiciones**

Altura = int(input(‘Cuánto mides?: ’)) (int para covertir str en número)

If altura >= 175

Print(‘eres una persona alta’)

Else

Print(‘Eres bajito!!`)

Si tu altura es igual o mayor de 175, te arrojará el primer mensaje. Si es inferior, el segundo

**Funciones**

Def mostrarAltura():

Altura = int(input(‘Cuánto mides?: ’)) (int para covertir str en número)

If altura >= 175

Print(‘eres una persona alta’)

Else

Print(‘Eres bajito!!`)

“mostrarAltura()” te permitirá reutilizar el código de abajo sin necesidad de escribirlo de nuevo

Ejemplo 2:

Var\_altura = int(input(‘Cuánto mides?: ’)) (int para covertir str en número)

Def mostrarAltura(altura):

If altura >= 175

Print(‘eres una persona alta’)

Else

Print(‘Eres bajito!!`)

mostrarAltura(var\_altura)

mostrarAltura(var\_altura)

mostrarAltura(var\_altura)

En este caso, no te pregunta 3 veces Cúanto mides, puesto que lo sacas de la función “mostrarAltura()”, pero te da la respuesta 3 veces

**Listas**

Guardar una colección de datos dentro de una misma variable

Personas = [“Víctor”, “Paco”, “Pepe”]

Print(personas[0]) -> Estaría imprimiendo el primero

For persona in personas

Print(‘-‘ -+- persona)

**Tipos de datos:**

****

Int: número sin decimales, ya sea positivo o negativo.

float se usan para números no muy grandes con decimales

double para números muy grandes con decimales.

char es un carácter Unicode, y solemos escribirlo entre comillas simples (‘ ‘).

String: cadena de caracteres

Boolean: Representa si una condición se cumple o no se cumple. Suelen tener dos valores identificados, true (verdadero) o false (falso)

**Scope, ámbito**

Texto

Descripción generada automáticamente

Al estar fuera de la función, no imprime “mensaje”

Texto

Descripción generada automáticamente

Al estar dentro de la función, imprime “mensaje”

Texto

Descripción generada automáticamente

Imprime, porque aunque está fuera de la función, se ha convertido en variable global, y están disponibles en cualquier punto del programa.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Exposiciones trabajos alumnos:**



Listas [] = Son mutables (se pueden manipular)

Tuplas () = Son inmutables

Set {} = Son mutables, pero no es una estructura ordenada. No se puede acceder con [1]. No admite repetidos.

Diccionarios {} = Se agrupan relación clave – valor. Ejemplos “Nombre”:”Diego”

Meter letras en un string:

list\_letters = ""

    for i in string:

            list\_letters += i

Dar la vuelta a un string

letters.lower()[::-1])

Ejemplo de recursividad (calcular factorial):

def factorial (x):

    if x == 1:

        return 1

    return x \* factorial(x-1)

print(factorial(5))