

Actividades Introducción a Python

Actividad de clase 3: Usamos la función para truncar números de la librería "math"

Se solicita al alumno la creación de un programa Python que calcule la versión truncada (sin decimales) de un número decimal que se ha solicitado previamente al usuario por teclado. Ha de hacerlo usando la correspondiente librería de Python.

```
import math
numero_decimal = float(input("Introduce un número decimal: "))
truncado = math.trunc(numero_decimal);
print("El número truncado es es: " , truncado);
```

```
===== RESTART: /home/usuario/Actividad3.py :
Introduce un número decimal: 34.058
El número truncado es es: 34
```

Actividad de clase 4: tabla de multiplicar

Escribe un programa que solicite al usuario un número entero y muestre la tabla de multiplicar correspondiente a ese número, desde el 1 hasta el 10.

```
numero = int(input("Introduce un número entero: "))

# Mostrar la tabla de multiplicar desde el 1 hasta el 10
print("Tabla de multiplicar del" , numero)
for i in range(1, 11):
    resultado = numero * i
    print(numero, "x", i, "=", resultado)
```

```
===== RESTART: /home/usuario/Actividad4.py
Introduce un número entero: 2
Tabla de multiplicar del 2
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20
|
```

Actividad de clase 5: mostrar un números pares

Escribe un programa que solicite al usuario un número entero positivo y luego muestre por pantalla todos los números pares desde 1 hasta ese número. Se ha de usar el bucle for con range.

```
numero = int(input("Introduce un número entero positivo: "))  
  
print("Números pares desde 1 hasta" , numero)  
for i in range(2, numero + 1, 2):  
    print(i)
```

```
===== RESTART: /home/usuario/Actividad5.py  
Introduce un número entero positivo: 10  
Números pares desde 1 hasta 10  
2  
4  
6  
8  
10
```

Actividad de clase 6: ¿qué número es mayor?

Escribe un programa que solicite al usuario dos números e imprima si el primero es mayor o es menor que el segundo.

```
num1 = float(input("Ingresa el primer número: "))  
num2 = float(input("Ingresa el segundo número: "))  
  
if num1 > num2:  
    print("El número" , num1 , "es mayor que" , num2)  
elif num1 < num2:  
    print("El número" , num1 , "es menor que", num2)  
else:  
    print("El número" , num1 , "es igual que", num2)
```

```
===== RESTART: /h  
Ingresa el primer número: 4  
Ingresa el segundo número: 6  
El número 4.0 es menor que 6.0  
|
```

Actividad de clase 7: día de la semana

Escribe un programa que solicite al usuario un número del 1 al 7 e imprima el día de la semana correspondiente (1 para lunes, 2 para martes, etc.).

```
num = int(input("Ingresa un número del 1 al 7: "))

dias_semana = {
    1: "Lunes",
    2: "Martes",
    3: "Miércoles",
    4: "Jueves",
    5: "Viernes",
    6: "Sábado",
    7: "Domingo"
}

if 1 <= num <= 7:
    print("El día correspondiente es:" , dias_semana[num])
else:
    print("Número inválido. Debes ingresar un número entre 1 y 7.")
,
```

```
Ingresa un número del 1 al 7: 3
El día correspondiente es: Miércoles
```

Actividad de clase 8: números pares con el operador módulo

Escribe un programa con un bucle for que recorra todos los números entre el 1 y el número introducido por el usuario con el teclado. El programa debe mostrar por pantalla los números pares y no mostrar los impares. Para ello usará el operador módulo de Python, que se representa por el símbolo %, calcula el residuo de la división entre dos números. Para averiguar si un número es par, podemos utilizar este operador. Si el número es divisible por 2 sin dejar residuo, significa que es par. En otras palabras, si el número dividido entre 2 tiene un residuo igual a 0, entonces el número es par. Este es un enfoque común para determinar si un número es par utilizando el operador módulo en Python.

```
num = int(input("Introduce un número: "))

for i in range(1, num + 1):
    if i % 2 == 0:
        print(i)
```

```
===== RESTART: /home/usuario
Introduce un número: 3
2
```

Actividad de clase 9: Función para calcular el área de un rectángulo

Escribe un programa en Python que pregunte al usuario el tamaño de los dos lados distintos de un rectángulo y muestre el área del rectángulo por pantalla. El cálculo del área hace hacerse en una función. La función no muestra el área por pantalla, sólo la calcula.

```
def calcular_area(largo, ancho):  
    return largo * ancho  
  
largo = float(input("Introduce el tamaño del lado largo del rectángulo: "))  
ancho = float(input("Introduce el tamaño del lado ancho del rectángulo: "))  
  
area = calcular_area(largo, ancho)  
  
print("El área del rectángulo es:" , area)
```

```
===== RESTART: /home/usuario/Actividad9.py  
Introduce el tamaño del lado largo del rectángulo: 30  
Introduce el tamaño del lado ancho del rectángulo: 10  
El área del rectángulo es: 300.0  
|
```

Actividad de clase 10: Función que devuelve el mayor de dos números

Escribe un programa en Python que pregunte al usuario dos números cualesquiera y muestre el mayor de ambos por pantalla. El cálculo del número mayor debe hacerse en una función. La función no muestra el mayor número por pantalla, sólo la calcula.

```
def mayor_de_dos(num1, num2):  
    if num1 > num2:  
        return num1  
    else:  
        return num2  
  
numero1 = float(input("Introduce el primer número: "))  
numero2 = float(input("Introduce el segundo número: "))  
  
mayor = mayor_de_dos(numero1, numero2)  
  
print("El mayor de los dos números es:", mayor)
```

```
===== RESTART: /home/usuario/A  
Introduce el primer número: 20  
Introduce el segundo número: 5  
El mayor de los dos números es: 20.0  
|
```

Actividad de clase 11: Ejercicio de colecciones en Python (I)

Crea una lista que tenga los siguientes valores:

3,6,9,12,15,18,21,24,27,30

Para ello usa un bucle que vaya desde 1 a 10.

```
valores = []  
  
for i in range(1, 10):  
    multiplicador = i * 3  
  
    valores.append(multiplicador)  
  
print(valores)
```

Actividad de clase 12: Ejercicio de colecciones en Python (II)

Crea una lista de 10 elementos. Cada elemento será un número al azar entre el 1 y el 50.

Usa la librería random y la función randint que pertenece a dicha librería.

```
import random  
  
# lista vacía para guardar los números aleatorios  
numeros_aleatorios = []  
  
for i in range(10):  
    # Generar un número aleatorio entre 1 y 50 usando randint  
    numero = random.randint(1, 50) # random.randint(1, 50) genera  
    #un número aleatorio entre 1 y 50 inclusive  
    numeros_aleatorios.append(numero)  
  
print(numeros_aleatorios)
```

```
===== RESTART: /home/usi  
[41, 30, 8, 25, 3, 31, 13, 26, 23, 43]  
|
```

Actividad de clase 13: Ejercicio de colecciones en Python (III)

Crea una lista de 10 elementos. Cada elemento será un número al azar entre el 1 y el 50.

Usa la librería random y la función randint que pertenece a dicha librería. Después, pregunta al usuario que introduzca un número. A continuación recorre la lista y busca el número que ha introducido el usuario. Si lo encuentra, saca por pantalla el mensaje: "bingo!"

```
import random
lista_numeros = [] #Crea lista vacia para almacenar los núm aleatorios

for i in range(10): # range(10) genera una secuencia de 0 a 9
    #(10 iteraciones en total)

    numero_azar = random.randint(1, 50)
    # 'randint(1, 50)' genera un número entero aleatorio entre 1 y 50
    lista_numeros.append(numero_azar)

numero_usuario = int(input("Introduce un número en tre 1 y 50: "))

if numero_usuario in lista_numeros:
    print("¡Bingo!")
else:
    print("Lo siento, el número no está en la lista.")
```

```
===== RESTART: /home/usuar:
Introduce un número en tre 1 y 50: 33
Lo siento, el número no está en la lista.
```

```
Introduce un número en tre 1 y 50: 20
¡Bingo!
|
```