

# Clase 5-6

## Estructuras de control

Un código es una secuencia de instrucciones, que por norma general son ejecutadas una tras otra.

Podemos verlo como una receta de cocina, en la que tenemos unos pasos a seguir.

Empezamos por el principio, vamos realizando cada tarea, y una vez hemos llegado al final, hemos terminado

```
poner_agua_hervir()  
echar_arroz_hervir()  
cortar_pollo()  
freir_pollo()  
mezclar_pollo_arroz()
```

# Clase 5-6

## Estructuras de control

Las estructuras de control en Python permiten dirigir el flujo de ejecución de un programa. Estas estructuras son esenciales para tomar decisiones en el código, repetir operaciones y controlar el proceso de ejecución en función de distintas condiciones.

Python ofrece varias estructuras de control, cada una adaptada a necesidades específicas

- Estructuras de control condicionales
- Estructuras de control iterativas

# Clase 5-6

## Estructuras de control

### Indentación

Para hablar de estructuras de control de flujo en Python, es imprescindible primero, hablar de indentación.

¿Qué es la indentación?

En un lenguaje informático, la indentación es lo que la sangría al lenguaje humano escrito (a nivel formal).

No todos los lenguajes de programación, necesitan de una indentación, aunque sí, se estila implementarla, a fin de otorgar mayor legibilidad al código fuente. Pero en el caso de Python, la indentación es obligatoria, ya que de ella, dependerá su estructura.

Una indentación de 4 (cuatro) espacios en blanco, indicará que las instrucciones indentadas, forman parte de una misma estructura de control.

Una estructura de control, entonces, se define de la siguiente forma

```
inicio de la estructura de control:  
    expresiones
```

# Clase 5-6

## Lectura de datos por teclado

La función input()

input(prompt)

La función input(prompt) imprime en pantalla el mensaje 'prompt' y devuelve una línea de texto escrita por el usuario con el teclado.

Esta función se puede utilizar para detener la marcha del programa hasta que se pulse la tecla 'Enter' o la tecla 'Return'

```
1 nombre = input('Escribe tu nombre: ');  
2 print('Hola', nombre)
```

```
PS C:\Users\jose luis\Desktop\diploPython> & C:/is/Desktop/diploPython/vacio.py  
Escribe tu nombre: jose luis  
Hola jose luis  
PS C:\Users\jose luis\Desktop\diploPython> 
```

# Clase 5-6

## Estructuras de control

### Estructura secuencial

```

1  Algoritmo area_triangulo
2      Definir base, altura como numerico
3      Escribir "Ingrese la base ; "
4      Leer base
5      Escribir "Ingrese la altura ; "
6      Leer altura
7      area<-(base * altura) / 2
8      Escribir "Su area es ; " area
9      Escribir "      /\ "
10     Escribir "    /  \ "
11     Escribir "   /    \ "
12     Escribir "  /      \ "
13     Escribir " /_____\ "
14     Escribir " "
15  FinAlgoritmo
16

```

PSelnt - Ejecutando proceso AREA\_TRIANGULO

\*\*\* Ejecución Iniciada. \*\*\*

Ingrese la base ;

> 6

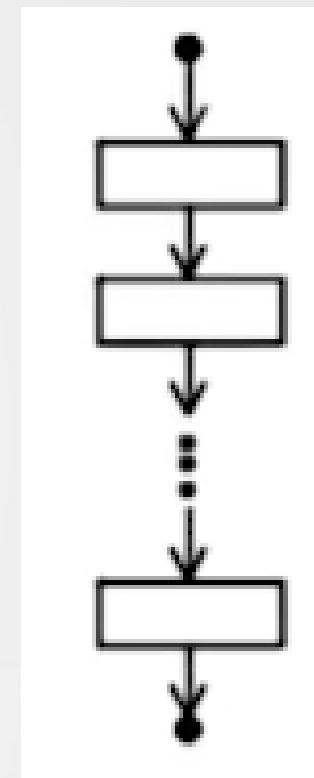
Ingrese la altura ;

> 4

Su area es ; 12



\*\*\* Ejecución Finalizada. \*\*\*



# Clase 5-6

## Estructuras de control

### Estructura secuencial

```
1  Algoritmo area_triangulo
2      Definir base, altura como numerico
3      Escribir "Ingrese la base ; "
4      Leer base
5      Escribir "Ingrese la altura ; "
6      Leer altura
7       $area \leftarrow (base * altura) / 2$ 
8      Escribir "Su area es ; " area
9      Escribir "      /\n"
10     Escribir "      /  \n"
11     Escribir "      /    \n"
12     Escribir "      /      \n"
13     Escribir "      /_____\n"
14     Escribir "      "
15  FinAlgoritmo
16
```

```
1  if __name__ == '__main__':
2      base = float()
3      altura = float()
4      print("Ingrese la base ; ")
5      base = float(input())
6      print("Ingrese la altura ; ")
7      altura = float(input())
8      area = (base*altura)/2
9      print("Su area es ; ",area)
10     print("      /\n")
11     print("      /  \n")
12     print("      /    \n")
13     print("      /      \n")
14     print("      /_____\n")
15     print("      ")
16
```

# Clase 5-6

## Estructuras de control

### Operadores relacionales y operadores lógicos

Símbolo	Significado	Ejemplo	Resultado
==	Igual que	5 == 7	False
!=	Distinto que	rojo != verde	True
<	Menor que	8 < 12	True
>	Mayor que	12 > 7	True
<=	Menor o igual que	12 <= 12	True
>=	Mayor o igual que	4 >= 5	False

Operador	Ejemplo	Explicación	Resultado
and	5 == 7 and 7 < 12	False and False	False
and	9 < 12 and 12 > 7	True and True	True
and	9 < 12 and 12 > 15	True and False	False
or	12 == 12 or 15 < 7	True or False	True
or	7 > 5 or 9 < 12	True or True	True
xor	4 == 4 xor 9 > 3	True o True	False
xor	4 == 4 xor 9 < 3	True o False	True

# Clase 5-6

## Estructuras de control

### if/if-else/if-elif

Ingrese la edad de una persona y determine si es mayor o menor de edad

```
# estructura condicional simple

EdadPersona=int(input("ingrese la edad de la persona"))
if EdadPersona<=0:
    print("la edad no es válida")
if EdadPersona>18:
    print("la persona es mayor de edad")
if EdadPersona<18:
    print("la edad de la persona es de un menor de edad")
```



# Clase 5-6

## Estructuras de control

### if/if-else/if-elif

Ingrese la edad de una persona y determine si es mayor o menor de edad

```
1  # estructura condicional doble anidada
2
3  EdadPersona=int(input("ingrese la edad de la persona"))
4  if EdadPersona>0:
5      if EdadPersona>18:
6          print("la persona es mayor de edad")
7      else:
8          print("la edad de la persona es de un menor de edad")
9  else:
10     print("la edad no es válida")
11
```

# Clase 5-6

## Estructuras de control

### if/if-else/if-elif

Ingrese la edad de una persona y determine si es mayor o menor de edad

```
1  # estructura condicional doble elif
2
3  EdadPersona=int(input("ingrese la edad de la persona"))
4  if EdadPersona<0:
5      print("la edad no es válida")
6  elif EdadPersona>18:
7      print("la persona es mayor de edad")
8  else:
9      print("la edad de la persona es de un menor de edad")
10
```

# Clase 5-6

## Estructuras de control

### Estructuras repetitivas (bucles for y while)

Realizar un programa que imprima los números del 1 al 100

```
1  
2 print("inicio del programa")  
3 for x in range(1,11):  
4     print(x)  
5 print("fin del programa")  
6
```

```
is/Desktop/diploPython/vacio.py"  
inicio del programa  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
fin del programa
```

# Clase 5-6

## Estructuras de control for

Realizar un programa que permita la carga de 5 valores por teclado y posteriormente, nos muestre la suma de los valores ingresados y su promedio

```
1
2  suma=0
3  for x in range(5):
4      num=int(input("Ingrese un valor:"))
5      suma=suma+num
6  promedio=suma/5
7  print("La suma es: ")
8  print(suma)
9  print("El promedio es: ")
10 print(promedio)
```

# Clase 5-6

## Estructuras de control

bucles for

Realizar un programa para crear un listado de frutas e imprimir su contenido

 Spyder (Python 3.12)

```
In [2]: frutas = ["manzana", "banana", "cereza", "mandarina", "naranja", "higo"]

In [3]: for fruta in frutas:
...:     print(fruta)
...:
manzana
banana
cereza
mandarina
naranja
higo
```

# Clase 5-6

## Estructuras de control

### Bucle while

```
In [7]: contador=0

In [8]: while contador<3:
...:     print("dentro del contador")
...:     contador=contador+1
...:
dentro del contador
dentro del contador
dentro del contador

In [9]: |
```

# Clase 5-6

## Estructuras de control

### Bucle while

Este bucle, se encarga de ejecutar una misma acción "mientras que" una determinada condición se cumpla.

```
In [4]: anio=2010  
  
In [5]: while anio<=2025:  
...:     print("Informes del Año", str(anio))  
...:     anio=anio+1  
...:
```

```
Informes del Año 2010  
Informes del Año 2011  
Informes del Año 2012  
Informes del Año 2013  
Informes del Año 2014  
Informes del Año 2015  
Informes del Año 2016  
Informes del Año 2017  
Informes del Año 2018  
Informes del Año 2019  
Informes del Año 2020  
Informes del Año 2021  
Informes del Año 2022  
Informes del Año 2023  
Informes del Año 2024  
Informes del Año 2025
```

# Clase 5-6

## Estructuras de control

### Bucle while

Escribir un programa que solicite la carga de un valor positivo y nos muestre desde 1 hasta el valor ingresado de uno en uno

```
In [9]: num = int(input("Ingrese un número positivo: "))
...:
...: if num > 0:
...:     for i in range(1, num + 1):
...:         print(i)
...: else:
...:     print("El número ingresado no es positivo.")
...:
```

Ingrese un número positivo: 8

1  
2  
3  
4  
5  
6  
7  
8

```
In [10]: numero = int(input("Ingrese un número positivo: "))
...:
...: while numero <= 0:
...:     print("Por favor, ingrese un número positivo.")
...:     numero = int(input("Ingrese un número positivo: "))
...:
...: contador = 1
...: while contador <= numero:
...:     print(contador)
...:     contador += 1
...:
```

Ingrese un número positivo: 5

1  
2  
3  
4  
5



# Clase 5-6

## Estructuras de control

### Bucle while

Desarrollar un programa que permita la carga de 5 valores por teclado y nos muestre posteriormente la suma de los valores ingresados y su promedio

```
1  contador = 0
2  suma = 0
3
4  # Bucle while para ingresar 10 valores
5  while contador < 5:
6      valor = float(input(f"Ingresa el valor {contador + 1}: "))
7      suma += valor
8      contador += 1
9
10 # Calcular el promedio
11 promedio = suma / 10
12
13 # Mostrar los resultados
14 print("\nSuma de los valores:", suma)
15 print("Promedio de los valores:", promedio)
16
17
```

# Tipos de datos/Métodos

# FIN