Estructuras Iterativas

...

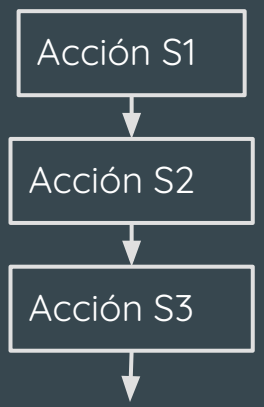
Programación
2024

Cómo haría para calcular la media de una lista de números ingresados por el usuario?

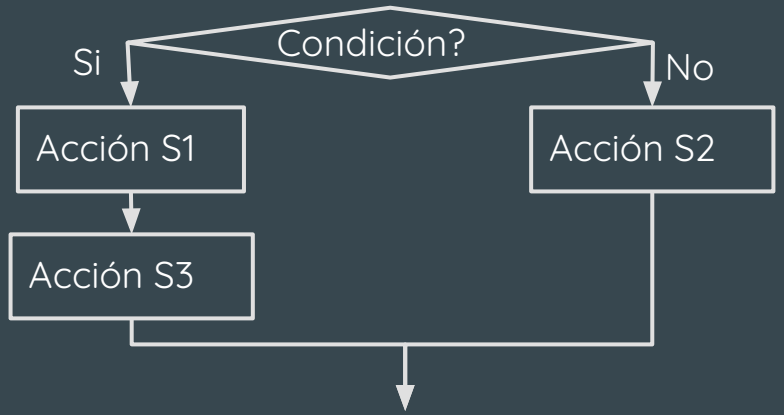
La lista termina cuando el usuario ingresa el número cero.

Estructuras de control

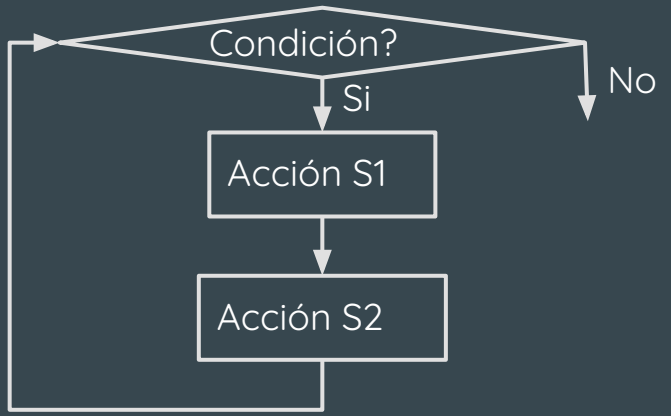
Cualquier algoritmo se puede construir utilizando combinaciones de tres estructuras de control:



Secuencial



Selectiva



Iterativa

Estructuras Iterativas

- Se denominan **bucles** (o ***loop***, en inglés) a una secuencia de instrucciones que se repite un cierto número de veces.
- Una **iteración** es cada ejecución (repetición) de esa secuencia de instrucciones.
- Todos los bucles se repiten hasta que se cumpla una **condición** determinada.

Estructuras Iterativas

- Existen 2 tipos de estructuras repetitivas en Python:
  - ***while***
  - ***for***
- En los bucles tipo ***while*** se conoce la condición de corte, pero no es necesario saber a priori el número de iteraciones.
- Al momento de declarar un bucle ***for*** es necesario definir el número de iteraciones que se van a necesitar.
- Al igual que en las estructuras selectivas, es necesario ***indentar los bloques de código***.

Bucle *while*

```
print (“Cuenta hacia atrás”)

inicio = 10
while inicio > 0:
  print (inicio)
  inicio -= 1
```

- Se realiza una acción ***mientras*** una condición sea verdadera.
- Dentro del bucle debe modificarse la variable incluida en la condición para poder alcanzar el final del bucle.

Bucles infinitos

- Es imprescindible identificar correctamente la condición de corte de un ***while***.
- Si la condición nunca toma el valor *False*, el bucle y el programa continuarán infinitamente.

```
# Ejemplo de bucle infinito

result = 0
while result >= 0:
    result += 1
print (result)
```

Bucle *for*

```
print (“Imprimiendo una palabra, letra por letra”)

for letra in “programacion”:
    print (letra)
```

- Realiza una acción ***por*** cada valor a ser evaluado.
- Todo lo que siga a la palabra reservada ***in*** se considera la lista de valores que se va a evaluar.
- Este grupo de valores **NO** puede modificarse.

Bucle *for*: Listas de valores

- Un bucle for solo funciona a partir de una lista prefijada de valores.
- Esos valores pueden ser:
  - Caracteres dentro de un string:

```
for letra in “progracion”
```

  - Números en un rango:

```
for numero in range(0, 10)
```

  - Elementos dentro de una lista <- siguiente clase!!

Sentencias de salto

- Es posible detener o saltar valores en un bucle.
- La sentencia ***continue*** hace que la iteración no termine y se continúe con la siguiente
- La función ***break*** hace que el bucle se termine
- Se aconseja evitar estas sentencias y definir mejor las condiciones de corte o los rangos de valores a utilizar.

Ejemplo de *continue*

```
for mes in range (1,13):
  for dia in range (1,32):
    if mes == 2 and dia > 28:
      continue
    if (mes == 4 or mes == 6 or mes == 9 or
        mes == 11) and dia > 30:
      continue
    print (dia, “/”, mes)
```

Estructuras anidadas

- Al igual que con las estructuras selectivas, es posible anidar bucles entre sí, junto con otras estructuras.
- La indentación va a definir qué bloque de código pertenece a cada estructura.

Ejemplo 1

- Calcular la media de una lista de números ingresados por el usuario. La lista termina cuando el usuario ingresa el número cero.

Ejemplo 1: pseudocódigo

- Entrada: lista de números (negativo termina).
- Salida: promedio

- suma = 0; cantidad = 0
- Pedir numero
- mientras numero != 0:
  - suma += numero
  - cantidad += 1
  - Pedir numero
- Devolver suma/cantidad

Ejemplo 1: código

```
suma = 0
cantidad = 0
numero = float(input())
while numero != 0:
  suma += numero
  cantidad += 1
  numero = float(input())

print (suma/cantidad)
```

Ejemplo 2

- Obtenga la raíz cuadrada de todos los números entre 0 y un número ingresado por el usuario.

Ejemplo 2: pseudocódigo

- Entrada: num
- Salida: raíz cuadrada de los números entre 0 y num

- Pedir número
- Para cada num entre o y número:
    - raiz = num**(0.5)
    - devolver raiz

Ejemplo 2: código

```
numero = int(input())
for n in range(0, numero+1):
  raiz = n**0.5
  print (n, raiz)
```

```
matias@matias-laptop:~$ python test.py
9
0 0.0
1 1.0
2 1.4142135623730951
3 1.7320508075688772
4 2.0
5 2.23606797749979
6 2.449489742783178
7 2.6457513110645907
8 2.8284271247461903
9 3.0
```

Ejemplo 3

- Contar cuántas vocales tiene una palabra ingresada por un usuario.

Ejemplo 3: pseudocódigo

- Entrada: palabra
- Salida: número de ‘a’, ‘e’, ‘i’, ‘o’ y ‘u’

- Pedir palabra
- contador = 0
- por cada letra en palabra:
  - si letra == ‘a’ or letra == ‘e’ or letra == ‘i’ or letra == ‘o’ or letra == ‘u’
    - contador += 1
- Devolver contador

Ejemplo 3: código

```
palabra = input(“Ingrese palabra: “)
contador = 0
for letra in palabra:
  if letra == “a” or letra == “e” or … # resto de vocales
    contador += 1
print (“El número de vocales es”, contador)
```