Flujo en python

Programación Unidad 2: Python y algoritmia

Ph.D. Santiago Echeverri-Arteaga

La lógica de un programador:

Mi madre me dijo

Ve a la tienda y trae una botella de leche. Si hay huevos:

trae media docena.

Volví con 6 botellas de leche y mi madre me preguntó “¿Por qué has comprado 6 botellas de leche?”

Pues... porque había huevos

Índice

Índice

Ciclo por y mientras en python

Ciclo por en python

¿Para qué sirve el ciclo por?

Ciclo por en python

¿Para qué sirve el ciclo por?
Realiza determinado conjunto de órdenes mientras itera sobre una secuencia. Cuando llega al final de la secuencia se sale del ciclo.

Ciclo por en python

¿Para qué sirve el ciclo por?
Realiza determinado conjunto de órdenes mientras itera sobre una secuencia. Cuando llega al final de la secuencia se sale del ciclo.
Sintaxis

```
for VARIABLE_A_ITERAR in range(NUMERO):
    COMANDOS A REPETIR NUMERO VECES
```

Ejemplo:

```
for i in range(3):
    num = float(input('Ingrese un número: '))
    print ('El cuadrado de tu número es', num*num)
print('El ciclo ha concluido')

for i in range(3):
    print(i+1, '-- Hello')
```

Uso de Range

Statement	Values generated
`range(10)`	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
`range(1, 10)`	1, 2, 3, 4, 5, 6, 7, 8, 9
`range(3, 7)`	3, 4, 5, 6
`range(2, 15, 3)`	2, 5, 8, 11, 14
`range(9, 2, -1)`	9, 8, 7, 6, 5, 4, 3

Uso de Range

Statement	Values generated
`range(10)`	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
`range(1, 10)`	1, 2, 3, 4, 5, 6, 7, 8, 9
`range(3, 7)`	3, 4, 5, 6
`range(2, 15, 3)`	2, 5, 8, 11, 14
`range(9, 2, -1)`	9, 8, 7, 6, 5, 4, 3

Veamos un ejemplo

Uso de Range

Statement	Values generated
`range(10)`	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
`range(1, 10)`	1, 2, 3, 4, 5, 6, 7, 8, 9
`range(3, 7)`	3, 4, 5, 6
`range(2, 15, 3)`	2, 5, 8, 11, 14
`range(9, 2, -1)`	9, 8, 7, 6, 5, 4, 3

Veamos un ejemplo
pero antes... algo de spoiler

Operador repetición

para imprimir algun texto repetido basta con multiplicar el texto por un número en el comando print. Ejemplo:

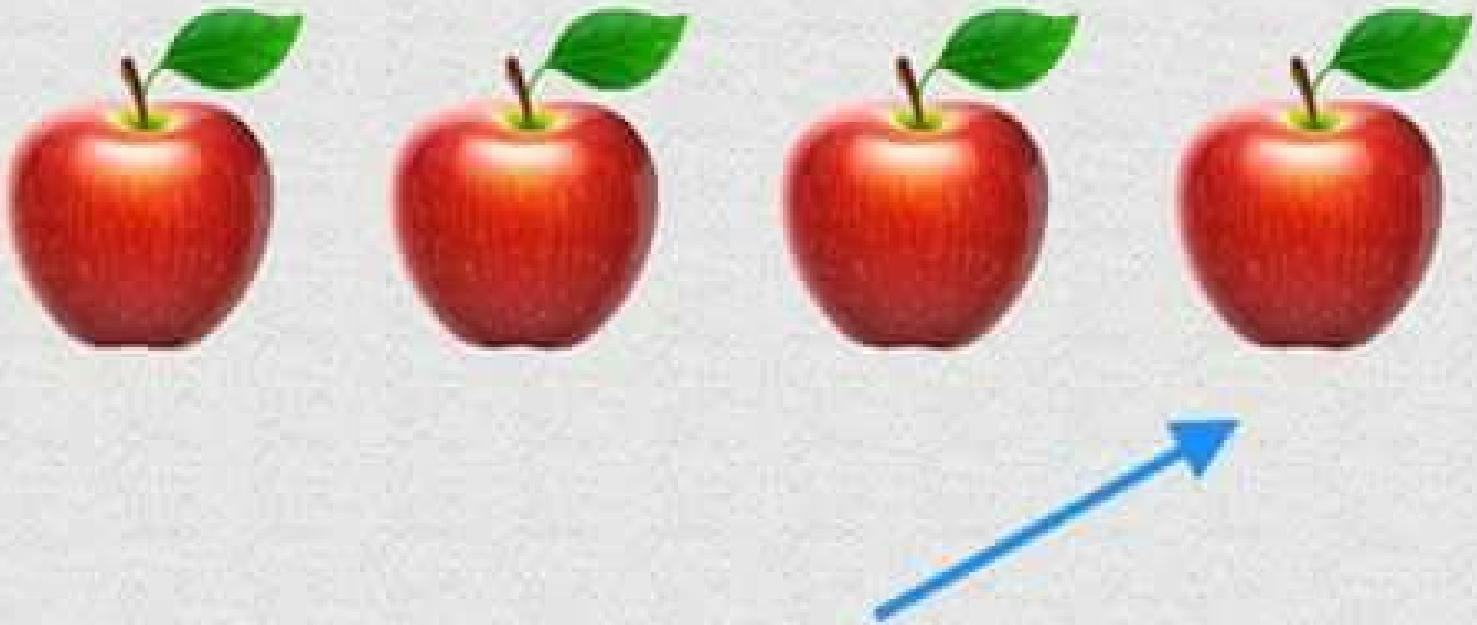
```
print('A'*6) # es equivalente a print('AAAAAA')
```

Siendo así, que generarían los siguientes códigos?

```
for i in range(4):
    print('*'*6)

for i in range(4):
    print('*'*(i+1))
```

YOU KNOW YOU'RE A PROGRAMMER WHEN..



THIS IS THE THIRD APPLE

for en ciertos valores particulares

```
for i in [1.2, 2.3, 3.85, -4.587, 5]:
    print(i)
```

Problema: Escribir un programa que calcule el dinero que tendrá una persona tras N años al poner X cantidad de dinero en un CDT con una tasa de interés del $p\,\%$.

Definición: Se debe escribir un programa que solicite el dinero que se va a poner en un CDT, la tasa de interés y el número de años que se dejará el dinero ahí. El programa debe imprimir en pantalla el dinero que se tendrá después de ese tiempo.

Análisis: El algoritmo tendría que pedir los tres datos al inicio del programa, luego hacer el cálculo y finalmente imprimir el resultado. Para hacer el calculo se puede tener en cuenta que el dinero que se tiene al final del primer año $X * (1 + \frac{p}{100})$ y ese sería el dinero sobre el que se calcule la ganancia de intereses para el segundo año y así sucesivamente.

Diseño: Tres input, un ciclo for y al final un print.

Implementación:

Implementación:

```
X = float(input("Ingrese la cantidad de dinero que
desea poner en el CDT: "))
N = int(input("¿Cuantos años dejará el dinero? "))
p = float(input("Ingrese el porcentaje de interés
pactado (solo el número): "))
for i in range(N):
    X = X*(1+p/100)
print("Después de {0} años, usted tenrá {1}"
.format(N,X))
```

Ciclo mientras en python

¿Para qué sirve el ciclo mientras?

Ciclo mientras en python

¿Para qué sirve el ciclo mientras?Repite una serie de órdenes mientras una condición (puede ser una condición compuesta) sea cierta. Es útil cuando **no se sabe exactamente cuantas veces se va a repetir el código**

Condicional (si) en python

Controlando el flujo por medio de Booleanos

“En 1840 el matemático George Boole demostró que las reglas clásicas de la lógica podían expresarse en forma puramente matemática utilizando solo los dos valores verdadero y falso. Un siglo después, Claude Shannon se dio cuenta de que el trabajo de Boole podía utilizarse para optimizar el diseño de conmutadores telefónicos electromecánicos. Su trabajo condujo directamente al uso de la lógica booleana para diseñar circuitos de computadora. En honor al trabajo de Boole, la mayoría de los lenguajes de programación modernos utilizan un tipo que lleva su nombre para realizar un seguimiento de lo que es verdad y lo que no.”

Relaciones y evaluación de corto circuito

¿Qué guarda en $x = 15 > 5$?

Relaciones y evaluación de corto circuito

¿Qué guarda en $x = 15 > 5$?
que dará $(2 < 3)$ **or** $(1/0)$ y que dará $(1/0)$ **or** $(2/3)$. ¿Por qué?

Relaciones y evaluación de corto circuito

¿Qué guarda en $x = 15 > 5$?

que dará $(2 < 3)$ **or** $(1/0)$ y que dará $(1/0)$ **or** $(2/3)$. ¿Por qué?

Junto a los operadores or, not y and los operadores relacionales son

Symbol	Operation
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
==	Equal to
!=	Not equal to

Relaciones y evaluación de corto circuito

¿Qué guarda en $x = 15 > 5$?

que dará $(2 < 3)$ **or** $(1/0)$ y que dará $(1/0)$ **or** $(2/3)$. ¿Por qué?

Junto a los operadores or, not y and los operadores relacionales son

Symbol	Operation
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
==	Equal to
!=	Not equal to

```
>>> 'A' < 'a'
True
>>> 'A' > 'z'
False
>>> 'abc' < 'abd'
True
>>> 'abc' < 'abcd'
True
```

Condicional si

Anteriormente habíamos visto los condicionales simples y bifurcada.
En cada una se tenía respectivamente

1. si CONDICIÓN:
       COMANDOS
2. si CONDICIÓN:
       COMANDOS 1
   sino:
       COMANDOS 2

ahora en python ellas son

1. if CONDICIÓN:
       COMANDOS
2. if CONDICIÓN:
       COMANDOS 1
   else:
       COMANDOS 2

elif

Adicionalmente tenemos el siguiente caso

1. if CONDICIÓN:
       COMANDOS 1
   elif CONDICION 2:
       COMANDOS 2
   else:
       COMANDOS 3

elif

Adicionalmente tenemos el siguiente caso

1. if CONDICIÓN:
   COMANDOS 1
   elif CONDICION 2:
   COMANDOS 2
   else:
   COMANDOS 3

```
# Este programa no funciona correctamente
edad = int(input("¿Cuántos años tiene? "))
if edad < 18:
    print("Es usted menor de edad")
elif edad < 0:
    print("No se puede tener una edad negativa")
else:
    print("Es usted mayor de edad")
```

Usando booleanos

Las dos siguientes implementaciones son exactamente iguales, pero la de la derecha le pasa **una variable en lugar de una condición** a un condicional, **ésta es interpretada como un booleano:**

```python
numero = int(input("Escriba un número: "))
if numero % 2 != 0:
    print(f"{numero} es impar")
else:
    print(f"{numero} es par")
```

```python
numero = int(input("Escriba un número: "))
if numero % 2:
    print(f"{numero} es impar")
else:
    print(f"{numero} es par")
```

Usando booleanos

```
print("Este programa mezcla dos colores.")
print("  r. Rojo      a. Azul")
primera = input("  Elija un color (r o a): ")
if primera == "r":
    print("  a. Azul      v. Verde")
    segunda = input("  Elija otro color (a o v): ")
    if segunda == "a":
        print("La mezcla de Rojo y Azul produce Magenta.")
    else:
        print("La mezcla Rojo y Verde produce Amarillo.")
else:
    print("  v. Verde    r. Rojo")
    segunda = input("  Elija otro color (v o r): ")
    if segunda == "v":
        print("La mezcla de Azul y Verde produce Cian.")
    else:
        print("La mezcla Azul y Rojo produce Magenta.")
print("¡Hasta la próxima!")
```

Contando

¿Qué hace el código?

```
count = 0
for i in range(1,101):
    if (i**2)%10==4:
    count = count + 1
print(count)
```

Vamos a practicar

Ejemplos

1. Escribir un programa que solicite el nivel de Ph y si está entre $0-4$ decir que es un ácido fuerte, $5-6$ ácido debil, 7 neutral, $8-9$ base debil y $10-14$ base fuerte
2. Escribe un programa de juegos de multiplicación para niños. El programa debe dar al jugador diez preguntas de multiplicación generadas al azar. Después de cada uno, el programa debe decirles si lo hicieron bien o mal y cuál es la respuesta correcta.
3. Escriba un juego que le permita al usuario jugar piedra papel o tijera cierto numero de rondas y al final diga el marcador.
4. Escriba un programa que le pida al usuario que ingrese un número e imprima todos los divisores de ese número

Ejercicios

Ejercicios i

Elija tres ejercicios a desarrollar y entreguelos la proxima clase

1. Escriba un programa que le permita al usuario pasar una temperatura de grados centígrados, farenheit o Kelvin a la escala térmica que el desee.
2. Escriba un programa que le pida al usuario una hora entre la 1 y las 12, le pida que ingrese am o pm y le pregunte cuántas horas en el futuro quiere ir. Imprima cuál será la hora dentro de tantas horas en el futuro, imprimiendo am o pm según corresponda. A continuación se muestra un ejemplo. (Se pueden ingresar número mucho mayores a 24)

Ejercicios ii

3. Un frasco de dulces de Halloween contiene una cantidad desconocida de dulces y si puedes adivinar exactamente cuántos dulces hay en el tazón, entonces ganas todos los dulces. La persona a cargo te dice: Si el caramelo se divide en partes iguales entre 5 personas, quedarían 2 dulces. Si se dividen en partes iguales entre 6 personas, la cantidad restante es de 3 piezas. Por último, pregunta por dividir los dulces en partes iguales entre 7 personas, y la cantidad que queda es de 2 piezas. Al mirar el cuenco, puede ver que hay menos de 200 piezas. Escribe un programa para determinar cuántas piezas hay en el cuenco.

Recuerde crear una función main() y emplear las buenas prácticas de la programación.