

Métodos de minería de datos en
Python

Programación básica en Python

Contenido

1

Manipulación de cadenas

2

Flujos condicionales

3

Flujos cíclicos



Ejercicio de repaso clase anterior

Tipos de datos



- El lenguaje provee **representaciones** para **conjuntos de valores**
- Les llamamos **tipos de datos**

Tipos básicos o primitivos

- Tipos numéricos: **int**, **float**
- Tipos de texto: **str**
- Tipos lógicos (booleanos): **bool**

En Acción!!!



En Acción!!!



obtener datos de entrada

variable = `input(texto)`

```
nombre = input("¿Cuál es tu nombre?")
saludo = "Hola,"
pregunta = "¿Qué tal la clase?"
print(saludo, nombre, pregunta)
```

¿Cuál es tu nombre?

```
nombre = input("¿Cuál es tu nombre?")
saludo = "Hola,"
pregunta = "¿Qué tal la clase?"
print(saludo, nombre, pregunta)
```

... ¿Cuál es tu nombre?

```
nombre = input("¿Cuál es tu nombre?")
saludo = "Hola,"
pregunta = "¿Qué tal la clase?"
print(saludo, nombre, pregunta)
```

¿Cuál es tu nombre?Gabriela
Hola, Gabriela ¿Qué tal la clase?

En Acción!!!

```
nombre = input("¿Cuál es tu nombre?")
saludo = "Hola,"
saludo2 = ". Espero estés disfrutando la clase"
print(saludo, nombre, saludo2)

lec = input("¿Cuántas semanas llevas de clase?: ")
total = 16
faltan = total - lec
print("Te faltan ", faltan, "semanadas para vacaciones ¡Ánimo!")
```

En Acción!!!

```
nombre = input("¿Cuál es tu nombre?")
saludo = "Hola,"
saludo2 = ". Espero estés disfrutando la clase"
print(saludo, nombre, saludo2)
```

```
lec = input("¿Cuántas semanas llevas de clase?: ")

¿Cuál es tu nombre?Juan
Hola, Juan . Espero estés disfrutando la clase
¿Cuántas semanas llevas de clase1
```

```
nombre = input("¿Cuál es tu nombre?")
saludo = "Hola,"
saludo2 = ". Espero estés disfrutando la clase"
print(saludo, nombre, saludo2)
```

```
lec = input("¿Cuántas semanas llevas de clase?: ")
total = 16
faltan = total - lec
print("Te faltan ", faltan, "semanadas para vacaciones ¡Ánimo!")
```

```
¿Cuál es tu nombre?Juan
Hola, Juan . Espero estés disfrutando la clase
¿Cuántas semanas llevas de clase1
```

```
TypeError                                Traceback (most recent call last)
<ipython-input-7-2e98e3958cd0> in <module>()
      6 lec = input("¿Cuántas semanas llevas de clase")
      7 total = 16
----> 8 faltan = total - lec
      9 print("Te faltan ", faltan, "semanadas para vacaciones ¡Ánimo!")
```

```
TypeError: unsupported operand type(s) for -: 'int' and 'str'
```


En Acción!!!

```
nombre = input("¿Cuál es tu nombre?")
saludo = "Hola,"
saludo2 = ". Espero estés disfrutando la clase"
print(saludo, nombre, saludo2)

lec = input("¿Cuántas semanas llevas de clase?: ")
total = 16
faltan = total - lec
print("Te faltan ", faltan, "semanadas para vacaciones ¡Ánimo!")

¿Cuál es tu nombre?Juan
Hola, Juan . Espero estés disfrutando la clase
¿Cuántas semanas llevas de clase?1
-----
TypeError                                Traceback (most recent call last)
<ipython-input-7-2e98e3958cd0> in <module>()
      6 lec = input("¿Cuántas semanas llevas de clase")
      7 total = 16
----> 8 faltan = total - lec
      9 print("Te faltan ", faltan, "semanadas para vacaciones ¡Ánimo!")

TypeError: unsupported operand type(s) for -: 'int' and 'str'
```

```
print(type(nombre))
```

```
<class 'str'>
```

```
print(type(lec))
```

```
<class 'str'>
```

En Acción!!!

```
nombre = input("¿Cuál es tu nombre?")
saludo = "Hola,"
saludo2 = ". Espero estés disfrutando la clase"
print(saludo, nombre, saludo2)

lec = int(input("¿Cuántas semanas llevas de clase?: "))
total = 16
faltan = total - lec
print("Te faltan ", faltan, "semanadas para vacaciones ¡Ánimo!")
```

```
¿Cuál es tu nombre?Juan
Hola, Juan . Espero estés disfrutando la clase
¿Cuántas semanas llevas de clase?: 1
Te faltan 15 semanasadas para vacaciones ¡Ánimo!
```



Manipulación de cadenas

Strings

- Piense en una **secuencia** de caracteres sensitivos
- Puede establecer **comparaciones** con `==`, `>`, `<=`, etc.
- `len()` es una función usada para obtener el largo de la cadena
`s = "abs"`
`len(s)` → evalúa 3
- Las barras cuadradas permiten indexar valores que están en ciertas posiciones:
index: 0 1 2 → La indexación siempre empieza en el caracter 0
index -3 -2 -1 → El último carácter tiene asignado el índice -1

`s[0]`

`s[1]`

`s[2]`

`s[3]`

`s[-1]`

`s[-2]`

`s[-3]`

Strings

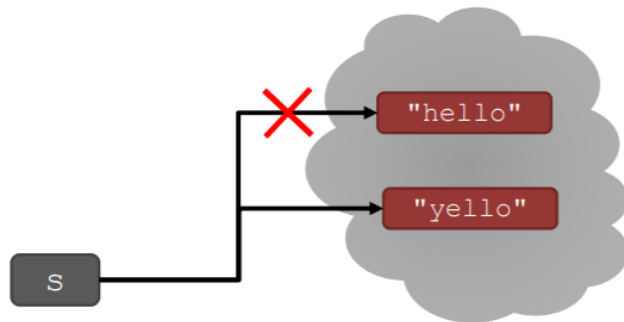
- Se puede hacer slice de una cadena usando la sintaxis [start:stop:step] (step = 1 por defecto)
- Dependiendo de lo que se desee obtener, se puede omitir el índice y dejar solo los dos puntos

s = "abcdefgh"

s[3:6]	def
s[3:6:2]	df
s[::]	abcdefgh
s[::-1]	hgfedcba
s[4:1:-2]	ec

Strings

- Las cadenas de texto son **inmutables**, no pueden ser modificadas



```
s = "hello"  
s[0] = "y" # genera error  
s = "y"+s[1:] # permitido, genera un nuevo string  
print(s)
```

Strings

- Las cadenas de texto también pueden contener espacios e incluso salto de líneas

```
print("Hoy no están que se duermen ¿Cierto, chicxs?")
```

Hoy no están que se duermen ¿Cierto, chicxs?

```
print("Hoy no están que se duermen \n ¿Cierto, chicxs?")
```

Hoy no están que se duermen
¿Cierto, chicxs?

Strings

Ejercicios:



"Línea"



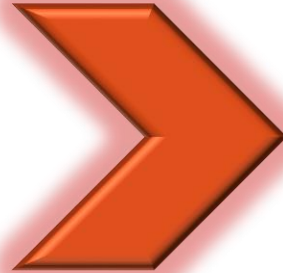
"Línea 1 \n
Línea 2"



"SALUDO"[2]



"SALUDO"[3:1:-1]



'Línea'



'Línea 1
Línea 2'



'L'



'UL'

RETO

Escriba un programa tipo *cheerleader*, el cual le permita generar una porra para la universidad, utilizando al menos dos reglas para manipulación de strings. La salida debe ser más o menos la siguiente:

```
Dame un letra!!!!: U
U!!!!
Dame un letra!!!!: S
S!!!!
Dame un letra!!!!: T
T!!!!
Dame un letra!!!!: A
A!!!!
Queeeeeeeeé diceeeee?
USTA !!!
USTA !!!
```

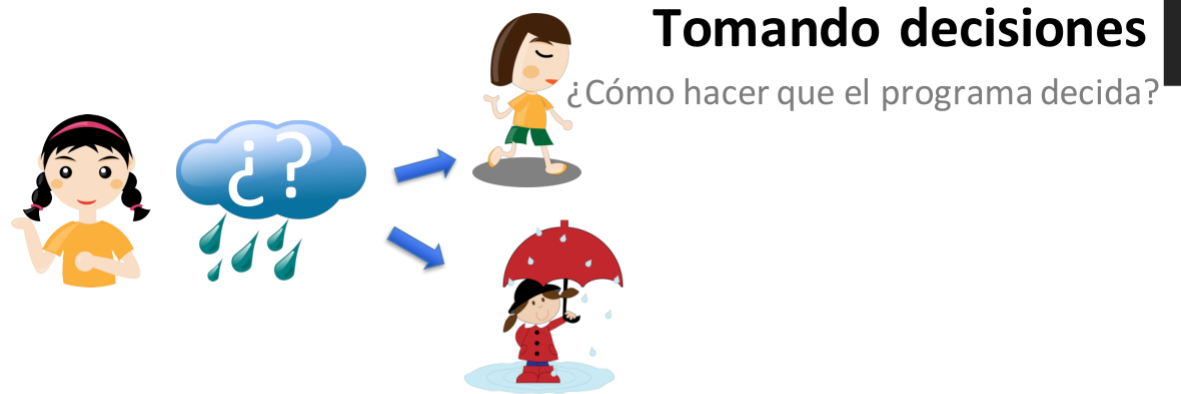
PISTA INICIAL:

```
size = int(input("¿Cuántas letras tiene la palabra?: "))
times = int(input("Nivel de Entusiasmo (1-10): "))
```



Flujos condicionales

Flujos Condicionales

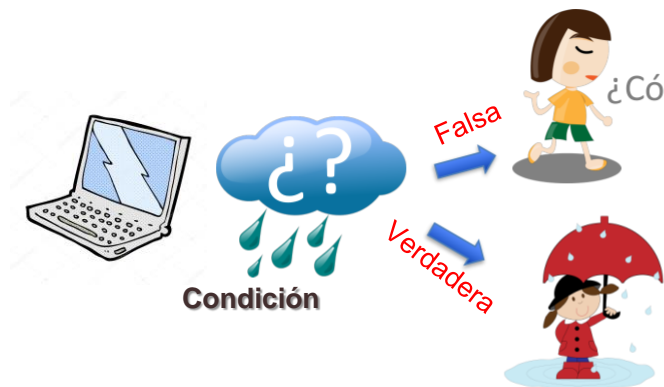


Objetivo: Lograr que el programa decida en base a una condición

Flujos Condicionales

Tomando decisiones

¿Cómo hacer que el programa decida?




- Si la **condición es verdadera**, la persona realizará una acción.
- Si la **condición es falsa**, realizará otra acción


Objetivo: Lograr que el programa decida en base a una condición


Flujos Condicionales


- Se aplican a **int** o **float**
 < <= > >= != ==
- Siempre entregan un tipo **bool**


 7 < 4.5


 False

 8 >= 4

 True


 8 != 6


 True

 6 == 7

 False


- Se aplican a **bool**
 not **and** **or**
- Siempre entregan un tipo **bool**


 **not** 7 < 4.5


 True


 3>5 **and** 2<6

 False

 3>5 **or** 2<6

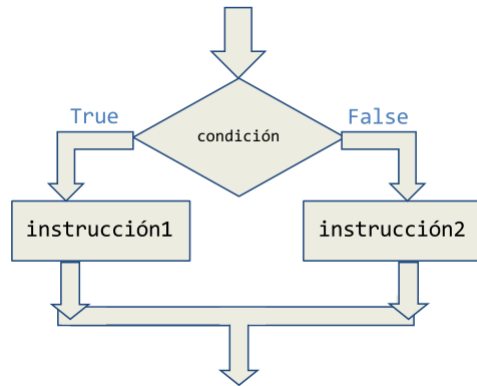
 True

 5>3 **and** 2<6

 True



Flujos Condicionales



instrucción condicional

```
if condición:
    instrucción1
else:
    instrucción2
```

dentro de if

dentro de else

indentación

Flujos Condicionales - if



```
llueve = True
if llueve == True:
    print("Llevaré paraguas")
else:
    print("No llevaré paraguas")
print("Ahora saldré a la calle")
```



Llevaré paraguas
Ahora saldré a la calle

Flujos Condicionales - else

```
llueve = False
if llueve == True:
    print("Llevaré paraguas")
else:
    print("No llevaré paraguas")
print("Ahora saldré a la calle")
```

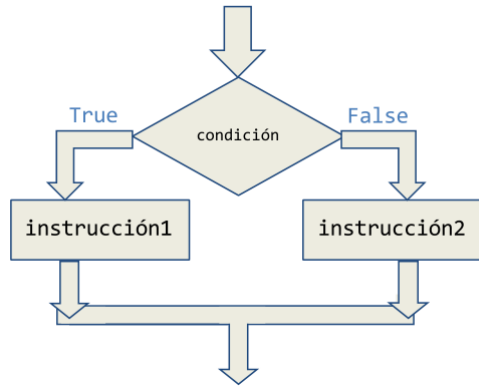


No llevaré paraguas
Ahora saldré a la calle

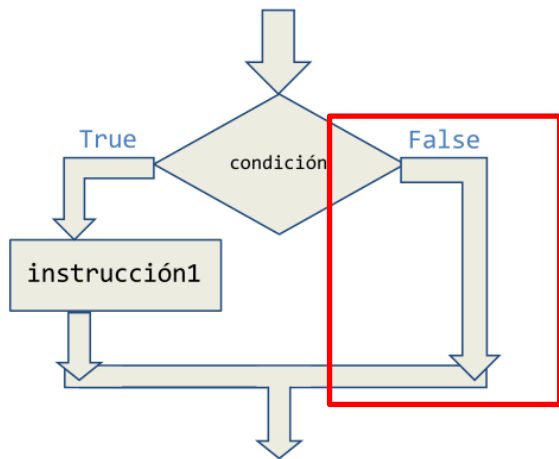
Flujos Condicionales - else

```
llueve = True
temperatura = 12
if llueve == True and temperatura < 20:
    print("Llevaré paraguas y abrigo")
else:
    print("No necesito paraguas o abrigo")
```

Flujos Condicionales – if sin else



Flujos Condicionales

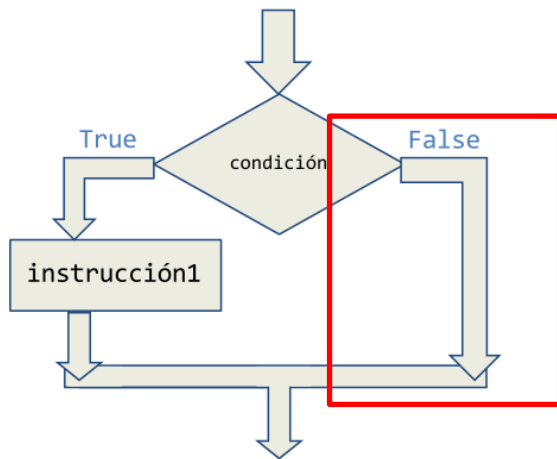


```
charco = True
print("Comienza la caminata!")
if charco == True:
    print("A saltar!")
print("Fin de la caminata")
```



```
Comienza la caminata!
A saltar!
Fin de la caminata
```

Flujos Condicionales



```
charco = False
print("Comienza la caminata!")
if charco == True:
    print("A saltar!")
print("Fin de la caminata")
```



Comienza la caminata!
Fin de la caminata

Flujos Condicionales

else sin if

```
charco = False  
else:  
    print("No saltaré")  
print("Sigo caminando")
```

Flujos Condicionales

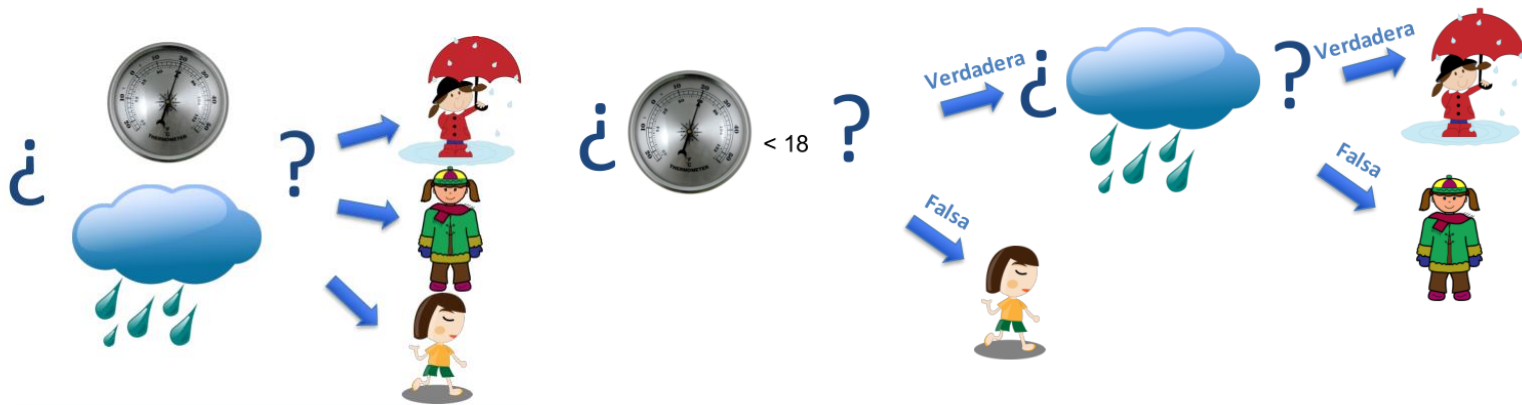
else sin if

```
charco = False
else:
    print("No saltaré")
print("Sigo caminando")
```

```
charco = False
if charco == True:
    print("A saltar!")
print("Sigo caminando")
else:
    print("No saltaré")
```

Flujos Condicionales

DECISIONES MÁS COMPLEJAS



Necesidad: El programa debe tomar una acción a partir de varias posibilidades

Flujos Condicionales

DECISIONES MÁS COMPLEJAS

```
llueve = True
temperatura = int(input("Ingresa temp°"))
if temperatura < 18:
    if llueve == True:
        print("Llevaré paraguas y abrigo")
    else:
        print("Solo llevaré abrigo")
else:
    print("No necesito paraguas ni abrigo")
```

Necesidad: El programa debe tomar una acción a partir de varias posibilidades

Flujos Condicionales

DECISIONES MÁS COMPLEJAS

```
llueve = True
temperatura = int(input("Ingresa tempº"))
if temperatura < 18:
    if llueve == True:
        print("Llevaré paraguas y abrigo")
    else:
        print("Solo llevaré abrigo")
else:
    print("No necesito paraguas ni abrigo")
```

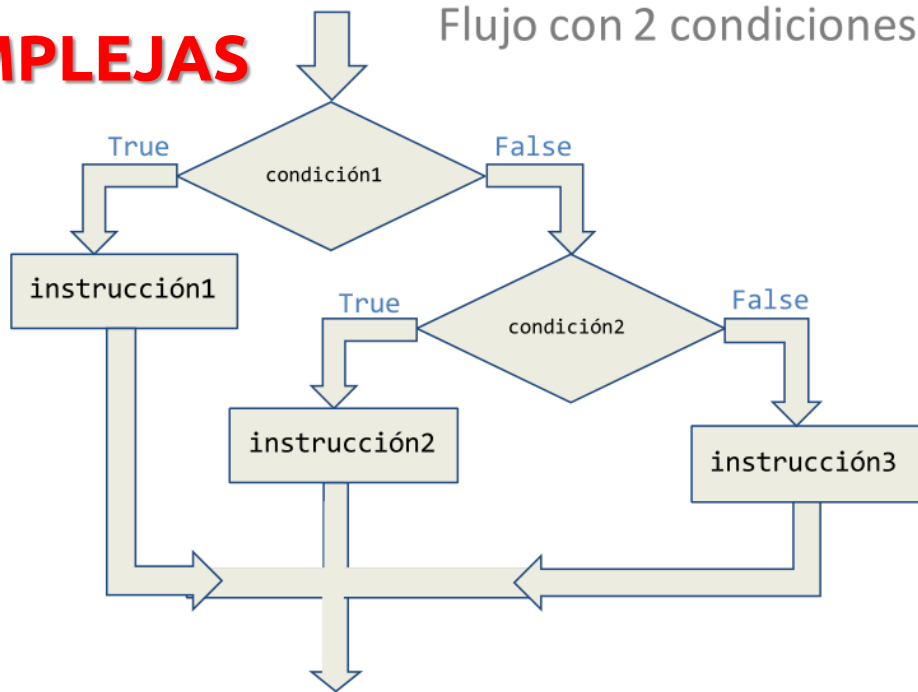
```
llueve = True
temperatura = int(input("Ingresa tempº"))
if temperatura < 18:
    if llueve == True:
        print("Llevaré paraguas y abrigo")
    else:
        print("Solo llevaré abrigo")
else:
    print("No necesito paraguas ni abrigo")
```

Necesidad: El programa debe tomar una acción a partir de varias posibilidades

Flujos Condicionales

DECISIONES MÁS COMPLEJAS

```
if condición1:  
    instrucción1  
elif  
    condición2:  
    instrucción2  
else:  
    instrucción3
```



Flujos Condicionales

ELIF: ESTRUCTURA GENERAL

```
if condición1:  
    instrucción1  
elif condición2:  
    instrucción2  
elif condición3:  
    instrucción3  
...  
else:  
    instrucciónN
```

```
if condición1:  
    instrucción1  
elif condición2:  
    instrucción2  
elif condición3:  
    instrucción3
```

Flujos Condicionales

ELIF: EJEMPLO

```
llueve = True
temperatura = int(input("Ingresa tempº"))
if temperatura < 18 and llueve == True:
    print("Llevaré paraguas y abrigo")
elif temperatura < 18 and llueve == False:
    print("Solo llevaré abrigo")
else:
    print("No llevaré paraguas ni abrigo")
```

Flujos Condicionales

ELIF: EJEMPLO

```
llueve = True
temperatura = int(input("Ingresa tempº"))
if temperatura >= 18:
    print("No llevaré paraguas ni abrigo")
elif llueve == True:
    print("Llevaré paraguas y abrigo")
else:
    print("Solo llevaré abrigo")
```



Flujos cíclicos

Flujos Ciclicos - while

- Cómo hacer que el programa realice un ciclo?

Fahrenheit
32
42
52
62
72



fº	cº
32	0
42	5
52	11
62	16
72	22

```
print("fº    cº")  
print("32    0")  
print("42    5")  
print("52    11")  
print("62    16")  
print("72    22")
```

Flujos Ciclicos - while

- Cómo hacer que el programa realice un ciclo?

Fahrenheit
32
42
52
62
72



°F	°C
32	0
42	5
52	11
62	16
72	22

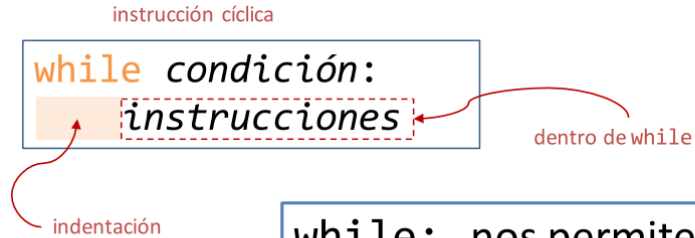
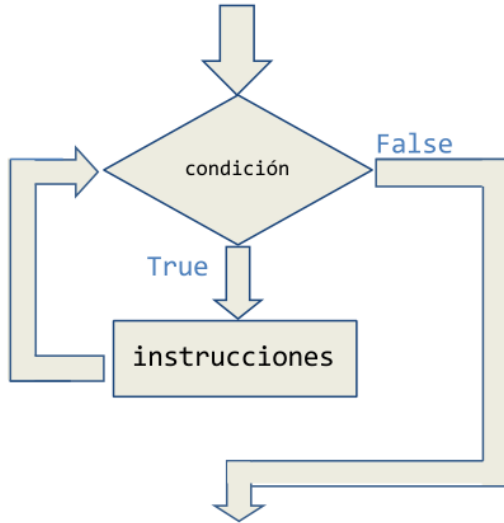


$$C = (F - 32) * \frac{5}{9}$$

Necesidad: Lograr que el programa realice un ciclo repitiendo un conjunto de instrucciones

Flujos Ciclicos - While

Flujo cíclico



while: nos permite repetir tantas veces como queramos un conjunto de instrucciones

Flujos Ciclicos - while

Pseudocódigo

```
temp = 32
print("fº      cº")
while temp sea menor a 73:
    imprimir temp y su conversión a Cº
    sumar 1 a temp
```

fº	cº
32	0
33	0
34	1
...	
71	21
72	22

Flujos Ciclicos - while

- Código

```
temp = 32
print("fº      cº")
while temp < 73:
    print(temp, "      ", int((temp-32)*5/9))
    temp = temp + 1
```

fº	cº
32	0
33	0
34	1
...	
71	21
72	22

Flujos Ciclicos - While

Código

```
temp = 32
print("fº      cº")
while temp < 73:
    print(temp, "      ", int((temp-32)*5/9))
```

fº	cº
32	0
...	
32	0
...	
32	0
...	

Flujos Ciclicos - for

Fahrenheit	Celsius
0	-17
1	-17
...	11
19	-7
20	-6

$$C = (F - 32) * \frac{5}{9}$$



fº	cº
0	-17
1	-17
2	-16
3	-16
4	-15
5	-15
6	-14
7	-13
8	-13
9	-12

Objetivo: Lograr que el programa repita un conjunto de instrucciones un número definido de veces

Flujos Ciclicos - for

Con WHILE:

```
temp = 0
print("fº      cº")
while temp < 21:
    print(temp, " ", int((temp-32)*5/9))
    temp = temp + 1
```

fº	cº
0	-17
...	
20	-6

```
n = 0
while n < 5:
    print(n)
    n = n+1
```

Con FOR:

```
print("fº      cº")
for temp in range(21):
    print(temp, " ", int((temp-32)*5/9))
```

fº	cº
0	-17
...	
20	-6

```
for n in range(5):
    print(n)
```

Flujos Ciclicos - for

instrucción cíclica for

```
for variable in secuencia:
```

```
    instrucciones
```

dentro de for

indentación

for: nos permite recorrer elemento por elemento una secuencia.
Podemos generar secuencias con range.

```
for variable in range(i,j,k):  
    instrucciones
```

- Cada vez que inicia un loop, **variable** toma un valor
- La primera vez, toma el valor más pequeño del rango
- La siguiente vez, toma el valor previo+1
- Y así sucesivamente...

Flujos Ciclicos - for

range(fin): Crea una secuencia de números que parte de 0 y llega hasta fin-1.

```
for i in range(7):  
    print(i)
```

range(k, fin): Crea una secuencia de números que parte de k y llega hasta fin-1.

```
for i in range(8,14):  
    print(i)
```

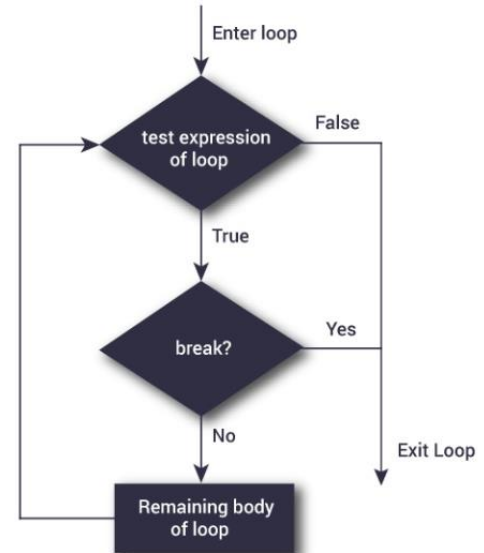
range(k, fin, h): Crea una secuencia de números que parte de k y llega hasta fin-1, avanzando de a h números.

```
for i in range(4,101,3):  
    print(i)
```


Flujos Ciclicos – break statement

```
mysum = 0
for i in range(5, 11, 2):
    mysum += i
    if mysum == 5:
        break
    mysum += 1
print(mysum)
```

¿Qué pasa en este programa?



Flujos Ciclicos – for vs. while

for

- Número conocido de iteraciones
- Se puede cortar la ejecución mediante un **break**
- Usa un contador
- Se puede reescribir un for usando un **while**

while

- Número ilimitado de iteraciones
- Se puede cortar la ejecución mediante un **break**
- Puede usar un contador pero debe ser inicializado antes del loop e incrementado dentro del loop
- **NO se puede reescribir un while usando un for**

¡Gracias!

¿Preguntas?

