

¿Qué es “for” en el contexto de python?

El *for* es un **tipo de bucle o ciclo que se utiliza para iterar sobre un objeto iterable** (listas, tuplas, diccionarios, strings, etc) y ejecutar un bloque de código de forma repetitiva de una manera sencilla y eficiente. Para este tipo de bucle, el número de iteraciones ya está definido de previamente, a diferencia del *ciclo while*.

La sintaxis básica del ciclo for en Python es la siguiente:

For **elemento** in **iterable**:

código a ejecutar para cada elemento

Donde:

- **Elemento** es la variable que toma el valor de cada elemento en la secuencia.
- **Iterable** es un objeto iterable.

En cada iteración del ciclo for, la variable elemento toma el valor del siguiente elemento en la secuencia y el código indentado debajo de la declaración for se ejecuta para ese elemento. Este proceso se repite para cada elemento en la secuencia hasta que se han agotado todos los elementos.

¿Qué es “while” en el contexto de python?

El bucle *while* evalúa una condición y luego ejecuta un bloque de código si la condición es verdadera. El bloque de código se ejecuta repetidamente hasta que la condición llega ser o es falsa.

La sintaxis básica es:

```
contador = 0
while contador < 10:
    # Ejecuta el bloque de código aquí
    # Siempre que el contador sea inferior a 10
```

La condición es cualquier expresión cuyo resultado solo puede ser verdadero o falso. Para realizar la condición se pueden incluir operadores de comparación de variables y también se pueden usar los conectores lógicos, exactamente igual con el *if*.

Un ejemplo de este bucle:

```
dia = 0
semana = ['Lunes', 'Martes', 'Miércoles', 'Jueves', 'Viernes', 'Sabado', 'Domingo']
while dia < 7:
    print("Hoy es " + semana[dia])
    dia += 1
```

Resultado:

```
Hoy es Lunes
Hoy es Martes
Hoy es Miércoles
Hoy es Jueves
Hoy es Viernes
Hoy es Sabado
Hoy es Domingo
```

Comportamiento de `isdecimal()`

`isdecimal()` es una función en Python que devuelve *True* si todos los caracteres en la cadena son caracteres decimales y hay al menos un carácter, *False* en el caso contrario. Los caracteres decimales son aquellos que se pueden usar para formar números de base 10.

Sintaxis:

`string_name.isdecimal()`

Aquí **`string_name`** es la *string* cuyos caracteres se van a comprobar

Un ejemplo de esta función:

```
s = "12345"  
print(s.isdecimal())  
True  
# contains alphabets  
s = "12geeks34"  
False  
print(s.isdecimal())  
# contains numbers and spaces  
s = "12 34"  
print(s.isdecimal())  
False
```