

Contando objetos con recursividad de pila

El siguiente código cuenta los dígitos de un número entero positivo:

```
def cuenta_digitos(numero):
    resultado = 0
    if type(numero) is int:
        if numero == 0:
            resultado = 1
        else:
            resultado = cuenta_digitos_aux(numero)
    return resultado

def cuenta_digitos_aux(numero):
    if numero != 0:
        return 1 + cuenta_digitos_aux(numero // 10)
    else:
        return 0
```

La función principal es *cuenta_digitos(numero)*. La primera validación que realiza es verificar si el argumento es un número entero: si no lo es, se retorna 0 (pues si no es número no tiene dígitos); si lo es, se verifica que este sea o no 0 (1 dígito).

Si el argumento es válido y no es 0 se llama a *cuenta_digitos_aux(numero)*. Ésta función utiliza división entera entre 10 para ir eliminando dígitos del número hasta convertirlo en 0. El ciclo recursivo se detiene cuando el número inicial ha sido reducido a 0.

Si, por ejemplo, llamamos a la función para contar los dígitos del número 200 el funcionamiento del código sería el siguiente:

1. 200 sí es de tipo *int*.
2. Y es diferente de 0, por lo tanto se invoca a la función auxiliar.
3. De entrada conocemos que $200 \neq 0$, por lo tanto se invoca la recursión a la función, esta vez utilizando división entera. Se almacena en pila 1 para luego sumar el resultado de la próxima recursión.
4. Se conoce que $20 \neq 0$, por lo tanto se invoca la recursión nuevamente y se suma un 1 en pila.
5. Se tiene que $2 \neq 0$, por lo tanto se invoca la recursión nuevamente y se suma un 1 en pila.
6. Ahora se cumple la igualdad $0 = 0$, se retorna 0 para ser sumado a la cantidad almacenada en pila.
7. Llenando hacia atrás en la pila se obtiene $0 + 1 + 1 + 1 = 3$. Por lo tanto 200 tiene 3 dígitos.

Un método similar se puede utilizar para obtener el número de elementos de una lista. Una lista es una colección ordenada de objetos, por ejemplo:

- [2, 3, 9] es una colección de enteros.
- ["hola", "mundo"] es una colección de strings.
- ["números", 100, 3.4] es una colección mixta de objetos.

Nota importante: Los strings son listas de caracteres por lo tanto muchas de las funciones aplicadas a las listas también son compatibles con los strings. Cuando usted declara una variable que sea igual a "hola", Python silenciosamente está creando la lista ['h', 'o', 'l', 'a'] en memoria. Estos principios son útiles para cursos más avanzados de la carrera.

Entonces para obtener la cantidad de elementos en una lista podemos utilizar el siguiente código:

```
def nelementos_lista(lista):
    resultado = 0
    if type(lista) is list:
        if lista != []:
            resultado = nelementos_lista_aux(lista)
    return resultado
```

```
def nelementos_lista_aux(lista):  
    if lista != []:  
        return 1 + nelementos_lista_aux(lista[1:])  
    else:  
        return 0
```