

Clase 10

Diego Armando Santillán Arriaga

18/01/19

1 Operaciones booleanas en Python

Para saber el tipo de una variable se utiliza: `type("nombre de la variable")`.

Para sumar una cantidad a una variable para fijar un incremento o decremento se utiliza: `"nombre de la variable" += "cantidad que se agregará a la variable"`

Cuando el cambio es de uno en uno se escribe `"variable" += i`. Esto funciona para cualquier operador (`*`, `+`, `-`, etc).

El resultado de comparar una variable que ya tiene un valor asignado con otro valor es `True` o `False`. Esto es una operación booleana.

Ejemplo:

```
>>> a = 10
>>> a == 20
False
>>> a == 10
True
```

Aparte de la comparación se pueden utilizar los siguientes operadores booleanos:

```
!= : negación
>>> a= 10
>>> a!=20
True
```

```
not ("condición que se va a negar"): tambien es una negación
>>> a=10
>>> not (a == 20)
True
```

```
<, >, >=, <= : comparación de magnitudes
>>> a=10
>>> a<=20
True
```

Evaluación booleana:

```
bool("variable")
```

Valora si algo es verdad o no. En el caso de cadenas de caracteres el resultado de bool es Falso solamente si la variable es la cadena vacía (" "). En el caso de valores numéricos el resultado de bool es Falso solamente si la variable es 0. En el caso de las listas (valores o cadenas entre corchetes) el resultado de bool es Falso solamente si la variable es la lista vacía ([]).

if bool(variable) == True: es lo mismo que if bool(variable):
if puede escribirse sin else pero no al revés.

2 Listas y operaciones entre listas

Las listas son valores, cadenas, listas contenidas dentro de corchetes:

```
>>> L=[2,3,4,4,6]  
>>> P=[2,3, 'Hola']
```

Para acceder los elementos dentro de una lista se utiliza: "nombre de la lista"["índice del elemento"].

Los elementos se numeran empezando desde el 0.

Para saber cuantos elementos tiene una lista se utiliza: **len("nombre de la lista")**

Una lista que es elemento de otra lista solo es contabilizada como un elemento.

Si uno de los elementos de la lista es otra lista se puede escribir len("nombre de la lista"["índice del elemento"]) para contabilizar sus elementos.

Para insertar un elemento al final de una lista se utiliza **"nombre de la lista".append("elemento")**.

Para insertar un elemento en una lista en la posición que queramos se utiliza **"nombre de la lista".insert(índice de la posición en donde quiero que esté, "elemento")**.

Para sacar un elemento de la lista se utiliza **"nombre de la lista".pop("índice del elemento que va a retirar")** y lo despliega.

La posición de un elemento de la lista es: $\text{posición} = \text{índice} + 1$. Esto también aplica para el número de elemento numerando la lista desde el 1.

Para agregar varios elementos a una lista en un solo comando se utiliza: **"nombre de la lista".extend(lista con los elementos que se van a agregar)**

Para asignar los elementos de la lista a un conjunto de variables se utiliza: **variable1, variable2, variable3, ... = "nombre de la lista"**. Esto toma un elemento de la lista y lo asigna a la variable correspondiente y así sucesivamente. No modifica la lista, solamente hace una asignación.

Las listas nos permiten almacenar varios elementos en una sola variable.

Para crear listas más fácilmente sin tener que introducir cada dato manualmente se escribe **"nombre de la lista" = range(límite inferior, límite superior, incremento)**. El primer elemento será el límite inferior, el segundo será el límite inferior más el incremento y así sucesivamente hasta el máximo número obtenido sumando el incremento que esté antes del límite superior (el intervalo considerado es [límite inferior, límite superior)).

3 Ciclos for

Podemos combinar un ciclo for con una lista de manera que se ejecute una instrucción a cada elemento de la lista. Al principio del ciclo for se introduce una variable que representará los elementos de la lista. Esta es la que se debe de usar al escribir la instrucción.

for "variable que representara elementos de la lista" in "nombre de la lista": "instrucción"

El for tomará cada variable de la lista y le aplicará la instrucción. Se puede utilizar un while para realizar la tarea de un for y viceversa solo que difieren en cierta medida.