

FUNDAMENTOS DE PROGRAMACIÓN EN PYTHON

[CC-BY-NC-SA] Leonardo Marco

FPP-PE-P3. Operadores y expresiones

ALUMNO: Alejandro Bermúdez

1) Realiza las siguientes operaciones aritméticas:

- `100+100`
- `11/3` (división decimal)
- `11//3` (división entera)
- `7 módulo 5`
- `5 módulo 7`

```
>>> print(100+100)

200

>>> print(11/3)

3.6666666666666665

>>> print(11//3)

3

>>> print(7%5)

2

>>> print(5%7)

5
```

2) Realiza las siguientes operaciones de asignación aumentada:

- `a igual a a+20`
- `a igual a a+b`
- `a igual a a++`

- a igual ++a

```
>>> a = 0

>>> b = 0

>>> a = a+20

>>> a

20

>>> a = a+b

>>> a

20

>>> a = a++

File "<stdin>", line 1

    a = a++
          ^
SyntaxError: invalid syntax

>>> a = ++a

>>> a

20
```

3) Realiza las siguientes comparaciones booleanas y piensa en el resultado obtenido:

- 2 es igual a 2
- 2 es distinto de 2
- 2 es menor a 2
- 2 es igual a "2"
- 2 es distinto a "2"
- 2 es menor a "2"
- [1,2] es igual a [1,2]
- [1,2] es menor a [1,3]
- [2,1] es menor a [1,999]

- 2 es igual a [2]
- [1,2] es igual a (1,2)
- [1,2] es menor a (1,2)
- [1, "a"] es igual a [1, 2]
- [1, "a"] es menor a [1, 2]
- [2, "a"] es menor a [1, 2]
- True es igual a 0
- True es igual a 1
- True es igual a 2
- False es menor a 1
- True es mayor a 1

```
>>> 2==2
```

```
True
```

```
>>> 2!=2
```

```
False
```

```
>>> 2<2
```

```
False
```

```
>>> 2=="2"
```

```
False
```

```
>>> 2!="2"
```

```
True
```

```
>>> 2<"2"
```

```
Traceback (most recent call last):
```

```
File "<stdin>", line 1, in <module>
```

```
TypeError: '<' not supported between instances of 'int' and 'str'
```

```
>>> [1,2] == [1,2]
```

```
True
```

```
>>> [1,2]<[1,3]
```

True

```
>>> [2,1]<[1,999]
```

False

```
>>> 2==[2]
```

False

```
>>> [1,2]==(1,2)
```

False

```
>>> [1,2]<(1,2)
```

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

TypeError: '<' not supported between instances of 'list' and 'tuple'

```
>>> [1,"a"]==[1,2]
```

False

```
>>> [1,"a"]<[1,2]
```

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

TypeError: '<' not supported between instances of 'str' and 'int'

```
>>> [2,"a"]<[1,2]
```

False

```
>>> True==0
```

False

```
>>> True==1
```

True

```
>>> True==2
```

```
False
```

```
>>> False<1
```

```
True
```

```
>>> True>1
```

```
False
```

```
>>>
```

4) Crea un script que pida al usuario 5 valores y los asigne a 5 variables `n1`, `n2`, `n3`, `n4` y `n5` (realiza las operaciones de asignación en una única instrucción). A continuación muestra:

- Si los 5 valores son iguales.
- Si los 5 valores están vacíos.
- Si el usuario a introducido los 5 valores ordenados.

```
# EJEMPLO DE INVOCACIÓN
```

```
python3 s.py
```

```
Valor de n1: 10
```

```
Valor de n2: 20
```

```
Valor de n3: 30
```

```
Valor de n4: 40
```

```
Valor de n5: 50
```

```
Los valores son iguales      : False
```

```
Los valores son vacíos      : False
```

```
Los valores están ordenados: True
```

```
n1, n2, n3, n4, n5 = int(input("Valor de n1: ")), int(input("Valor de
n2: ")), int(input("Valor de n3: ")), int(input("Valor de n4: ")),
int(input("Valor de n5: "))

print("Los valores son iguales: " + str(bool(n1==n2==n3==n4==n5)))

print("Los valores son iguales: " +
str(bool(" "==n1==n2==n3==n4==n5)))
```

```
print("Los valores están ordenados: " + str(bool(n1<n2<n3<n4<n5)))
```

Ejecución:

```
[alumno@vm1:~]$ python3 Documentos/operadores.py
```

Valor de n1: 10

Valor de n2: 20

Valor de n3: 30

Valor de n4: 40

Valor de n5: 50

Los valores son iguales: False

Los valores son iguales: False

Los valores están ordenados: True

5) Modifica el script anterior para que en lugar de obtener los 5 valores mediante la entrada estándar se obtengan por parámetros.

Script:

```
n1, n2, n3, n4, n5 = 10, 20, 30, 40, 50

print("Los valores son iguales: " + str(bool(n1==n2==n3==n4==n5)))

print("Los valores son iguales: " + str(bool(" "==n1==n2==n3==n4==n5)))

print("Los valores están ordenados: " + str(bool(n1<n2<n3<n4<n5)))
```

Ejecución:

```
[alumno@vm1:~]$ python3 Documentos/operadores.py
```

Los valores son iguales: False

Los valores son iguales: False

Los valores están ordenados: True

6) Crea un script que pida al usuario 2 valores, los asigne a las variables `v1` y `v2` y muestre:

- El valor booleano de `v1`.
- El valor booleano de `v2`.
- El resultado de hacer `and` de ambos valores.
- El resultado de hacer `or` de ambos valores.
- El valor booleano del `and` de ambos valores.
- El valor booleano del `or` de ambos valores.

EJEMPLO DE INVOCACIÓN

```
python3 s.py
```

```
Valor de v1      : aaa
```

```
Valor de v2      : bbb
```

```
Bool de v1       : True
```

```
Bool de v2       : True
```

```
v1 and v2        : bbb
```

```
v1 or v2         : aaa
```

```
Bool de v1 and v2 : True
```

```
Bool de v1 or v2  : True
```

Script:

```
v1, v2 = input("Valor de v1: "), input("Valor de v2: ")

print(bool(v1))

print(bool(v2))

print(v1 and v2)

print(v1 or v2)

print(bool(v1 and v2))
```

```
print(bool(v1 or v2))
```

Ejecución:

```
[alumno@vm1:~]$ python3 Documentos/operadores6.py
```

Valor de v1: aaa

Valor de v2: bbb

True

True

bbb

aaa

True

True

7) Crea un script que cree una lista con 10 valores numéricos aleatorios entre el 1 y el 50.

- A continuación solicitará un número al usuario y comprobará si el número introducido está en el array.
- Ejecuta el script varias veces hasta obtener un valor `True`.

Para obtener un número aleatorio en Python:

```
import random
```

```
n = random.randint(1, 10)      # número entero entre 1 y 10 (ambos  
incluidos)
```

EJEMPLO DE INVOCACIÓN

```
python3 s.py
```

```
Introduzca un número: 20
```

```
[18, 43, 24, 20, 17, 21, 26, 34, 40]
```

```
Coincidencia: True
```

Script:


```
import random

lista = [random.randint(1, 50) for _ in range(10)]

numero = int(input("Introduce un número: "))

print(lista)

print("Coincidencia: ", str(bool(numero in lista)))
```

Ejecución:

```
[alumno@vm1:~]$ python3 Documentos/operadores7.py
```

Introduce un número: 26

[25, 27, 9, 9, 36, 11, 26, 46, 10, 18]

Coincidencia: True