

Operadores

Los operadores son valga la redundancia operaciones que pueden hacerse en Python de forma automática utilizando los diferentes caracteres especiales existen varios tipo de operadores:

Aritméticos

suma

Para sumar al igual que en todos los lenguajes de programación se usa el carácter de suma +

```
suma = 2 + 2
print(suma)
print(2+2)
```

En la suma se puede hacer la suma de la misma variable mas otro numero usando el operador +=

```
numero1 = 20
print(numero1)
numero1+= 10
#se suma el valor de "numero1" con el 10 asignado y se guarda en la misma
variable
print(numero1)
```

Resta

Al igual que en la suma la resta se representa con el carácter de resta -

```
resta = 5 - 3
print(resta)
print(5-3)
```

al igual que en la suma se puede restar el valor de la variable con otro numero con el operador -=

```
resta = 10
print(resta)
resta -= 5
#se resta asi misma por 5 y se guarda en la misma variable
print(resta)
```

Multiplicación

En la multiplicación se usa el carácter de asterisco `*`

```
multiplicacion = 5 * 2
print(multiplicacion)
print(5*2)
```

Acá podríamos hablar de la potenciación pues para calcular la potencia de un numero elevado a otro con el operador `**`

```
potencia = 2**3
print(potencia)
```

igualmente podemos hacer que una variable se multiplique asi misma `*=` o se potencie asi misma `**=`

```
mult = 2
print(mult)
mult *= 4 #se multiplica asi misma por 4 y lo guarda en la misma variable
print(mult)
```

```
potencia = 5
print(potencia)
potencia**= 2 #se elevara el valor de "potencia" que es 5 a la 2 y se guardara
en la misma variable
print(potencia)
```

División

Para dividir se utiliza el carácter de pleca `/`

```
division = 4/2
print(division)
print(4/2)
```

Para que te un numero entero siendo esta una operación que da números decimales, simplemente se pone doble pleca //

```
division = 30//7
print(division)
print(30//7)
```

El modulo o el residuo se que es el restante de una división se utiliza el carácter de porcentaje %

```
modulo = 20%3
print(modulo)
```

para que una variable se auto divida así misma con los operadores /= //= %=

```
division = 12
division /=2 #se divide la variable "division" asi misma entre 2 y se guarda
en la misma variable
print(division)
division //= 5 #se divide asi mismo entre 5 dando solo numeros enteros
division %= 3 #se divide asi mismo entre 3 y devuelve el residuo o sobrante de
la division
```

Relacionales

Se emplean para relacionar 2 o mas valores

mayor a : > si el numero es mayor devuelve true de lo contrario devuelve False

```
numero = 5 > 4
print(numero)
```

menor a : < si el numero es menor devuelve true de lo contrario devuelve False

```
numero = 3 < 5
print(numero)
```

exactamente igual: `==`

```
numero = 5 == 5
print(numero)

#Tambien funciona con cadenas de texto

txto1 = "hola"
texto2 = "hola"
print(txto1 == texto2)
```

también esta la variante de igual o mayor que `>=` y igual o menor que: `<=`

```
numero1 = 20
numero2 = 10
numero3 = 20
print(numero1 >= numero2)
print(numero1 >= numero3)
print(numero2 <= numero3)
print(numero3 <= numero1)
```

Lógicos

los operadores lógicos sirven para tomar decisiones por medio de condiciones:

operador lógico `or` funciona si uno de dos valores es verdadero retorna `true` si ambos son falsos retorna `false`

```
a = true
b = false
print(a or b) #retorna true porque el valor "a" es verdadero
a = false
print(a or b) #retorna falso porque ambos son falsos
```

operador lógico `and` funciona si ambos valores son verdaderos devuelve `true` y si al menos 1 es falso devuelve `false`

```
a = true
b = true
print(a and b) #retorna true porque ambos son verdaderos
a = false
print(a and b) #retorna false porque al menos 1(siendo a) es falso
```

operador lógico `not` funciona si un valor es falso devuelve `true` si es verdadero devuelve `false`

```
a = false
print(not a) #devuelve true porque "a" es falso
a = true
print(not a) #devuelve false porque "a" es verdadero
```

Pertenencia

Sirven para saber si un valor esta en una variable

operador de pertenencia `in` y `not in`, el operador `"in"` nos retornara `true` si el valor esta en una variable, sino esta nos devolverá `false`, y viceversa con el operador `"not in"`

```
a = [1, 2, 4]
print(1 in a) #devolvera true pq 1 esta en la variable a
print(3 not in a) #devolvera true porque 3 no esta en la variable a

#tambien funciona con string

a = "Hola mundo"
print("mundo" in a) #devolvera true porque la cadena "mundo" esta en la variable a
```

Identidad

Un operador de identidad se emplea para comprobar si dos variables emplean la misma ubicación en memoria.

operadores de identidad: `is` y `is not`, el operador `"is"` devolverá `true` si dos variables tienen el mismo valor, y viceversa con el operador `"is not"`

```
```py
a = 3
b = 3
c = 4
print a is b # muestra True
print a is not b # muestra False
print a is not c # muestra True

x = 1
y = x
z = y
print z is 1 # muestra True
print z is x # muestra True

str1 = "FreeCodeCamp"
str2 = "FreeCodeCamp"

print str1 is str2 # muestra True
print "Code" is str2 # muestra False

a = [10,20,30]
b = [10,20,30]

print a is b # muestra False (ya que las listas son objetos mutables en Python
```