

PrimerosPasos

December 9, 2020

1 Primeros pasos en Pyhon

1.1 Conociendo el lenguaje

1.1.1 Tipos de datos

```
[2]: # Trabajando con tipos de datos
x = 29
y = 15.9
z = True
c = 2 + 5j
s = "Soy un string"
sx = 'Tambien soy un string'

# Imprime las variables
print(x)
print(y)
print(z)
print(c)
print(s)
print(sx)
```

```
29
15.9
True
(2+5j)
Soy un string
Tambien soy un string
```

1.1.2 Operadores

```
[8]: # Operadores aritméticos
x = 2 + 8
y = 25.4 + 30.4
z = 5 // 2 # Operador de división considerando solo la parte entera
w = 5 / 2 # Operador de división tradicional
```

```

r = 12 % 5 # Resto de la división
p = z ** 5 # Potencia
m = z * 5 # Multiplicación

# Imprime los valores
print(x)
print(y)
print(z)
print(w)
print(r)
print(p)
print(m)

```

```

10
55.8
2
2.5
2
32
10

```

```

[6]: # Operadores de string (cadenas de caracteres)
x = "cadena"
y = 'libre'

# Operador de concatenación
z = x + y
# Operador de repetición
w = y*4

# Imprime las variables
print(z)
print(x,y) # Notar la concatenación implícita de las cadenas e insertando el ↵
            ↪espacio de separación
print(w)

```

```

cadenalibre
cadena libre
librelibrelibrelibre

```

```

[7]: # Operadores relacionales
x = 29
y = 15
print(x == y) # Operador de igualdad
print(x > y) # De forma análoga existe el <
print(x != y) # Operador de desigualdad
print(x >= y) # De forma análoga existe el <=

```

False
True
True
True

```
[9]: # Operadores lógicos
print(x > y or x > 20)
print(x > y and y > 20)
print(not x != y)
```

False
False
False

```
[14]: # Operadores de asignación
x = 15
print("x = ", x)
x += 2
print("x += 2 :", x)
x -= 6
print("x -= 6 :", x)
x *= 3
print("x *= 3 :", x)
x **= 2
print("x **= 2 :", x)
x /= 2
print("x /= 2 :", x)
x /= 3
print("x /= 3 :", x)
x %= 2
print("x %= 2 :", x)
```

x = 15
x += 2 : 17
x -= 6 : 11
x *= 3 : 33
x **= 2 : 1089
x /= 2 : 544
x /= 3 : 181.33333333333334
x %= 2 : 1.3333333333333334

```
[16]: # Operadores a nivel de bits
print(5 & 3) # Representa el operador AND
print(5 | 3) # Representa el operador OR
print(5 ^ 3) # Representa el operador XOR
print(5 << 2) # n << x : representa el movimiento de x bits a la izquierda
print(64 >> 2) # n >> x : representa el movimiento de x bits a la derecha
```

1
7
6
20
16

1.1.3 Explicación de resultados

Lo primero es buscar la representación en binario de ambos números 5 (101) y 3 (011)

- Usando la tabla de verdad del AND $101 \text{ AND } 011 = 001$ que, en el sistema decimal, es el 1
- Usando la tabla de verdad del OR $101 \text{ OR } 011 = 111$ que, en el sistema decimal, es el 7
- Usando la tabla de verdad del XOR $101 \text{ XOR } 011 = 110$ que, en el sistema decimal, es el 6
- $101 \ll 2$ implica mover 2 bits a la izquierda y queda 10100 que, en el sistema decimal, es 20
 $= (16+0+4+0+0)$
- $100000 \gg 2$ implica mover 2 bits a la derecha y queda 10000 que, en el sistema decimal, es 16

1.2 Funciones asociadas a los tipos de datos

```
[28]: x = 29
print(type(x))
print("Quiero concatenar usando el operador + " + str(x))
s = "25"
print(int(s))
s = "25.4"
print(float(s))
print(bool("1"))
print(bool("0"))
print(bool(0))
print(bool("cualquier string siempre será verdadero"))
print(complex(1,6))
```

```
<class 'int'>
Quiero concatenar usando el operador + 29
25
25.4
True
True
False
True
(1+6j)
```