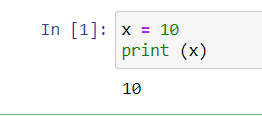
1 - VARIABLES & OPERATORS: [*LINK DOC COMPLETA*](https://drive.google.com/drive/folders/1ufSjqaebTQOoWpsSaei36hs9tOQMaOiJ)

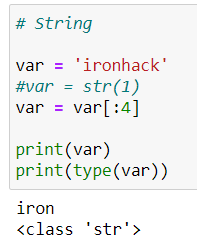
1.1 VARIABLES: son marcadores de posición (placeholders) para **almacenar valores de datos** para recuperarlos más tarde. Necesitan tener UN NOMBRE (éste debe de seguir unas normas):

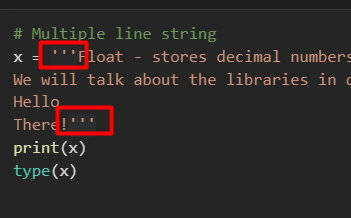
* El nombre de una variable debe comenzar con una letra o el carácter de subrayado
* Un nombre de variable **no puede comenzar con un número**
* El nombre de una variable **solo puede contener caracteres alfanuméricos** y **guiones bajos** (Az, 0-9 y \_)
* Los nombres de las variables **distinguen entre mayúsculas y minúsculas**

**→ print()** es devolver el resultado de una variable



1. **DataTypes**: Python is a dynamically typed language. That means the interpreter **infers the type** of an object **at runtime**.
   1. INTEGERS (enteros)(**int**): almacena valores enteros positivos o negativos.
   2. FLOAT (**tuple**): almacena números decimales. Interpreta cualquier punto como un decimal. Ej.: 20.2, 100.2403, -5.50
   3. STRING (**str**) (cadena de texto): almacena un grupo de caracteres encapsulados en una **comilla simple** o **comillas dobles**. Ej.: "hola", "Ironhack", "Profesor". Si utilizamos una línea de string larga (con múltiples líneas) usaremos **comillas triples**.

****



* 1. BOOLEAN (**bool**): almacena True o False (importante que el primer caracter empieza con mayus)
* **True** = 1 = cualquier string = cualquier número salvo el 0
* **False** = 0 = vacío = bool()
  + bvar = bool(87) -> True
  + bvar = bool(0) -> False
  + bvar = bool(“0”) -> True
  + bvar = bool(“kjndk”)\* -> True
  + bvar = bool('0') \* 8 -> True = 8 (en este caso el igual es a 8 pq estamos multiplicando true(1) por 8)

→ también hay un method → **bool(“x”)**, con esto creamos fácilmente STATEMENTS

\*\*\*\*

IMPORTANTE:

* cuando utilizas un **method no** hace falta **crear una nueva variable**, porque te lo aplica directamente sobre esa misma variable.

Ejemplo x = [1,1]

x.append(“hola”)

[1,1,”hola”]

Si hacemos y=x.append(“hola”)

y no existirá, no será un objeto. Si hacemos print(y) nos dará “None”, de que no hay nada dentro (y su type sería NoneType) → Estaríamos creando un **VACÍO**

* cuando haces una **función si que puedes crear una variable** y darle el return otra vez

\*\*\*\*

→ python\_data\_types\_structure

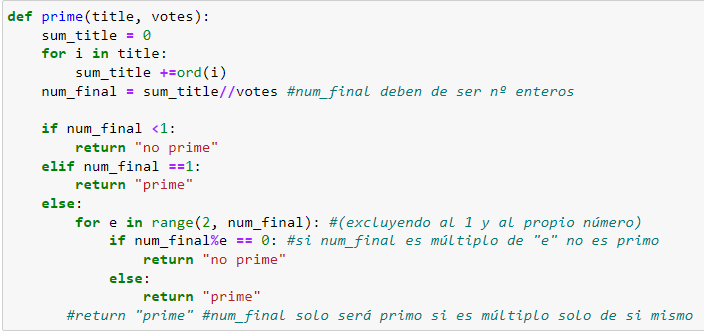
1.2 OPERATORS: son símbolos especiales en Python que se utilizan para realizar determinadas operaciones, como **operaciones matemáticas**, **operaciones lógicas** y **operaciones de comparación**. El valor o las variables sobre las que trabaja el operador se denominan OPERANDS.

* 1. ARITHMETIC OPERATORS: usadas para operaciones matemáticas:
* **`+`** : Addition (x + y)
* **`-`** : Subtraction (x - y)
* **`\***` : Multiplication (x \* y)
* **`/`** : Division (x / y) → CUIDADO que te lo devuelve FLOAT
* **`\*\*`** : Modulus (x \*\* y) → x elevado a y
* **`//`** : Floor division (x // y) → Redondea el resultado al valor entero más bajo en la recta numérica. lo llaman división entera. Te lo devuelve INT
* **“==”** : es igual a

x = 10

y = 5

* **`%`** : Modulus (x % y) → nos dice si un número es múltiplo de otro
  + if x%2 ==0 → x es múltiplo de 2 (es par)
  + if x%3 ==0 → x es múltiplo de 3
  + …..
  + números primos son aquellos SOLO divisibles por si mismo



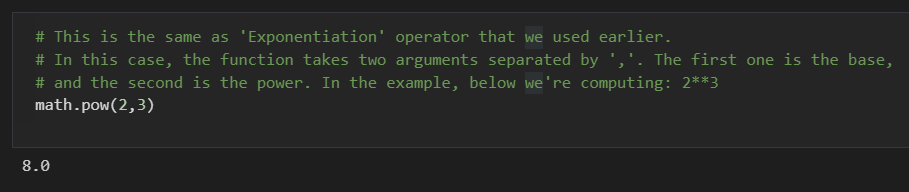
* 1. MATH LIBRARY: [link](https://docs.python.org/3/library/math.html) → para realizar operaciones matemáticas más complejas.
     1. **mach.floor**

import math # This is the way to have access to all the functions included in the `math` library

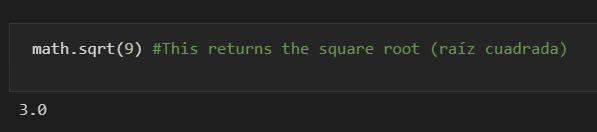
math.floor(2.9) # This is the same as 'floor division' operator that we used earlier

2

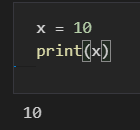
* + 1. **mach.pow**

****

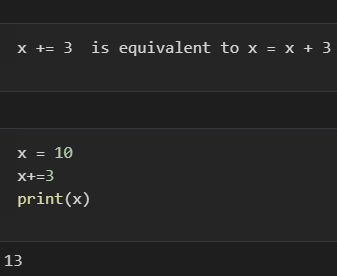
* + 1. **mach.sqrt**

****

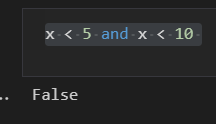
* 1. ASSIGNMENT OPERATORS: asignan valores a las variables



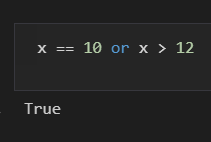
Algunos operadores de asignación especial asignan el valor y realizan una operación matemática simple al mismo tiempo



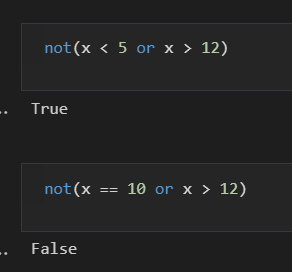
* 1. COMPARISON OPERATORS: son para hacer una comparación entre dos variables. Devuelven un valor booleano, True o False.
* **==** Equal x == y
* **!=** Not equal x != y
* **>** Greater than x > y
* **<** Less than x < y
* **>=** Greater than or equal to x >= y
* **<=**  Less than or equal to x <= y
  1. LOGICAL OPERATORS: se utilizan para combinar dos o más operaciones de comparación. Ejercicio **x = 10**
* **and**
* Devuelve **True** si se cumplen **TODAS las condiciones**.
* Sino **False**.



* **or**
* Devuelve **True** si se cumple **ALGUNA de las condiciones**.
* Sino **False** (si no se cumple ninguna también False)



* **not**
* REVIERTE el resultado, Devuelve **False** si el resultado es True



\* [LINK](https://www.tutorialspoint.com/python/python_basic_operators.htm) listado de operadores básicos. [LINK](https://www.programiz.com/python-programming/operators) más completo