Python

.alto nivel

.multipropósito

.autopep8 instalado (te corrige el código como cree que es la manera más legible)

.set de reglas

El código que escribimos en Python, es pasado al compilador de la maquina, luego pasa por el byte code, luego va al PVM (Python virtual machine), y luego va al lenguaje de la maquina

De esta manera, no importa nuestro sistema operativo o componentes, ya que se ejecuta en todas las versiones y se puede ejecutar directamente en el navegador (igual que java)

Variables

.python asigna un espacio en la memoria física del computador, para cuando queramos acceder hacer referencia en la parte de la memoria que acabamos de asignarle a dicha variable

Tipos:

alumnos = 5000 Integral(es un entero)

puntaje = 9.9 float(tiene decimal)

Strings

Lenght : permite obtener la longitud que tiene un string en particular. Ej en código:

print(len(nombre\_curso))

Para acceder a un índice en particular se usan los paréntesis cuadrados. Ej

print(nombre\_curso[2])

Para cortar el string se usa una sintaxis similar. Ej

print(nombre\_curso[0:8])

el 0 indica de donde se empieza a recortar el string, y después del punto(8), hasta donde queremos hacerlo. Si después de los puntos no le poneos ningún valor, Python ejecutara hasta el final del string. Si por el contrario, solo le pasamos el valor de la derecha(8), se tomara por defecto el 0

Formatear Strings

Para concatenar, se hace de la siguiente manera

nombre = "Marcelo"

apellido = "Gonzalez"

nombre\_completo = f"{nombre} {apellido}"

print(nombre\_completo)

esto devuelve en terminal: Marcelo Gonzalez

Podemos pasarle cualquier expresión que queramos al array; Ej.:

nombre = "Marcelo"

apellido = "Gonzalez"

nombre\_completo = f"{nombre[0]} {2 + 5}"

print(nombre\_completo)

esto devuelve en terminal: M 7

Métodos

Un método es una función que se encuentra dentro de un objeto. Ej.:

animal = "Gatito Hermoso"

print(animal.upper())

upper transforma todo en letras mayúsculas

Podemos concatenar los métodos. Ej.:

print(animal.strip().capitalize()) //strip elimina los espacios en blanco de la izquierda

Tambien de puede hacer esto para remover espacios en blanco en izquierda y derecha

print(animal.rstrip())

print(animal.lstrip())

BACKSLASH: ALT GR + una tecla ubicada a dos posiciones a la izquierda de la de retroceso

Números y Funciones

Python no trae muchas funciones nativas para trabajar con números, pero viene con un modulo de manera nativa que podemos importar para trabajar con números

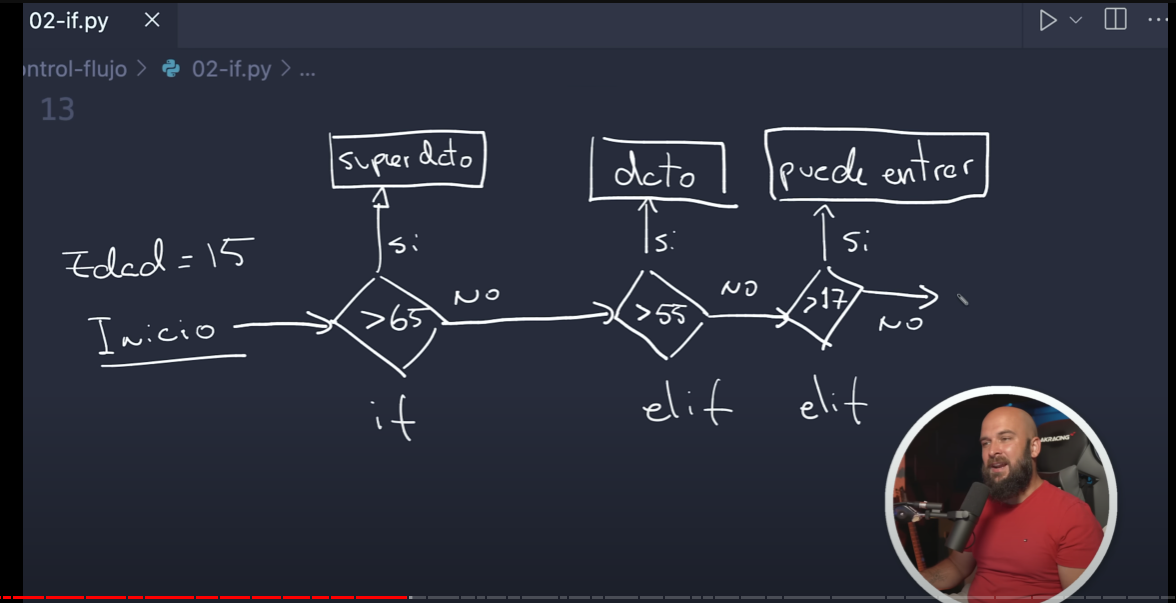
Ej: import math

Comparadores Lógicos

Devuelven false o true. Ej:

Print(1 > 2) = false

If de manera gráfica:



Operadores lógicos

And = Evalúa dos condiciones, si las dos son true, devuelve true. Al contrario, si una de las dos es false, devuelve false toda la operación

Or = Evalúa las dos condiciones, si uno es true, toda la operación da true. Si ambos dan false, devuelve false

Not = Niega el resultado de la operación (cambia los valores). Ej. ;

gas = False

encendido = False

if not gas or encendido:

    print("Puedes avanzar")

Aquí el not hace que las condiciones cambien de false a true, lo que da como resultado de toda

La operación True(El código ejecuta “P uedes avanzar”)

Operador de corto circuito

Va a depender del operador con el cual trabajemos

Si es que nosotros estamos trabajando con nuestro operador lógico and, quiere decir que absolutamente todas las operaciones que está usando and, tienen que ser true. Pero lo que ocurre, es que la primera si es falsa, el and de la derecha no lo va a evaluar debido al primer false. Significaría una orden de cómputo importante para el servidor. El código se ejecuta de izquierda a derecha siempre

En el caso de or, vasta que con el valor de la izquierda sea true para que toda la operación sea true

Si el de la izquierda es false, si va a evaluarse el valor de la izquierda

Loop For

El for cumple múltiples funciones, aunque se utiliza principalmente para iterar una lista de elemementos.

Dentro del in (dentro del for) se coloco la palabra reservada range, que es un iterable dentro de Python

Tambien se pueden iterar los strings. Ej.:

for char in "Ultimate Python":

    print(char)

While

Un loop que se ejecuta siempre y cuando se cumpla una condición

Loop Anidado

Podemos ejecutar un for, dentro de otro for. Ej.:

for j in range(3): (a esto se le conoce como outer for/loop)

    for k in range(2): (y esto inner for/loop)

        print(f"{j}, {k}")

Funciones

