Python para IA y NumPy

TAE IA
Formación de Talento Altamente Especializado

Definición

- Lenguaje de programación de **alto nivel**, que permite a los usuarios centrarse en la lógica del programa **sin preocuparse** por detalles de bajo nivel como la asignación y liberación de memoria, la arquitectura específica del procesador, etc.

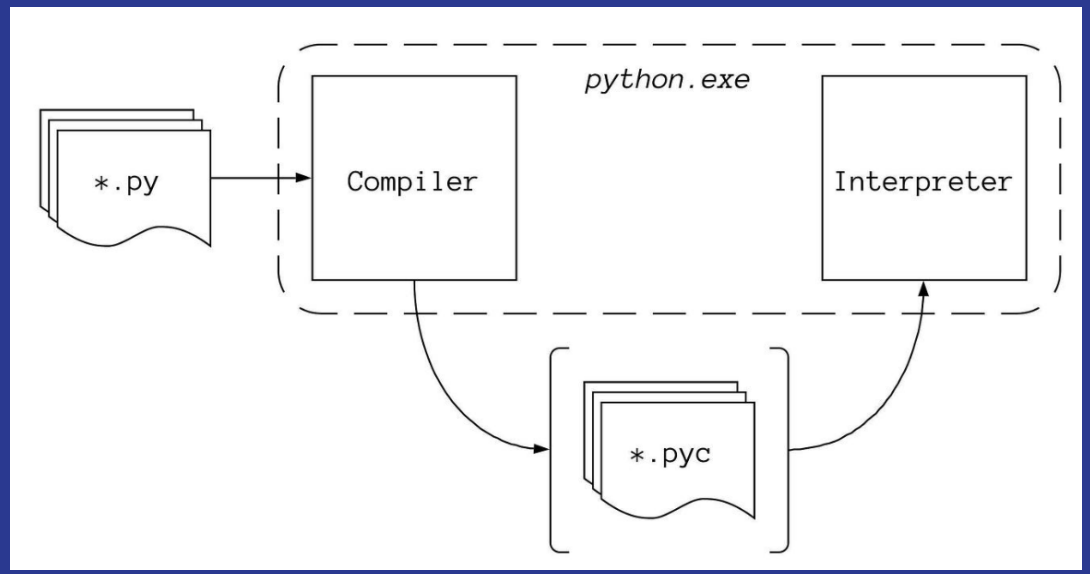
- Aunque permite la utilización de diferentes paradigmas de programación, **no impone** un único paradigma estricto, brindando flexibilidad en la forma de desarrollar el software.

python™

TAE IA
Formación de Talento Altamente Especializado

Ejecución

Python es un lenguaje **interpretado**, esto significa que tiene un intérprete el cual ejecuta en tiempo real cada línea de código sin necesidad de un proceso previo de transformación del mismo a formato binario, como por ejemplo lenguajes como C y C++.



TAE IA
Formación de Talento Altamente Especializado

Variables en Python

TAE IA
Formación de Talento Altamente Especializado

Definición

- Una variable es un espacio de almacenamiento que se asocia con un nombre simbólico (identificador) que se utiliza para almacenar datos.
- El valor de una variable puede cambiar durante la ejecución del programa.
- Se crea una variable simplemente asignándole un valor usando el operador **=**

```
x = 33
nombre = "Miguel"
```

Las variables **no necesitan** ser declaradas con un tipo específico; Python determina el tipo dinámicamente basado en el valor asignado.

TAE IA
Formación de Talento Altamente Especializado

Las variables en Python pueden almacenar datos de diferentes tipos, como números (int, float), cadenas de texto (str), listas (list), diccionarios (dict), entre otros.

Ámbito de una variable

Ámbito Local: Se refiere a las variables que están dentro de una función y solo son accesibles dentro de esa función.

Ámbito Global: Se refiere a las variables definidas fuera de cualquier función, accesibles desde cualquier parte del código.

TAE IA
Formación de Talento Altamente Especializado

Constantes en Python

TAE IA
Formación de Talento Altamente Especializado

Definición

- Una constante es un valor que **no debe cambiar** durante la ejecución del programa.
  - Aunque Python no tiene un mecanismo de constantes como tal, se utiliza una convención para definirlas.
- Las constantes en Python se definen utilizando nombres en mayúsculas para indicar que su valor no debe cambiar.

Aunque esta convención no impide que el valor cambie, es una práctica recomendada para indicar la intención de que el valor no debe modificarse.


```
PI = 3.14159
MIN_RANGO_MARINA_EC = "Alferez de Fragata"
```

TAE IA
Formación de Talento Altamente Especializado

Las constantes se utilizan para valores que se mantienen **constantes** a lo largo del programa, como configuraciones, parámetros fijos, y valores matemáticos

```python
# Variables
edad = 33
nombre_usuario = "Federica"

# Constantes
PI = 3.14159
MAX_RANGO_MARINA_EC = "Almirante"

# Uso de variables
edad += 1  # Incrementa la edad

# Uso de constantes
radio = 9
area_esfera = 4 * PI * (radio ** 2)  # Calcula el área de una esfera
volumen_esfera = (4 * PI * (radio ** 3)) / 3  # Calcula el volumen de una esfera
```

TAE IA
Formación de Talento Altamente Especializado

Estructuras de datos

TAE IA
Formación de Talento Altamente Especializado

Definición

- Son formas de **organizar y almacenar datos** de manera que se puedan **acceder y manipular eficientemente.**
- Permiten gestionar y trabajar con colecciones de datos de manera efectiva.
- Python ofrece una variedad de estructuras de datos integradas que se pueden usar para diferentes propósitos.

TAE IA
Formación de Talento Altamente Especializado

Listas

- Son colecciones **ordenadas de elementos** que pueden ser de **diferentes tipos** (enteros, cadenas, objetos, etc.).
- Las listas son **mutables**, lo que significa que **puedes cambiar, agregar o eliminar elementos después de que se haya creado la lista.**

```
lista = [1, 2, 3, "texto", 4.5]
```

TAE IA
Formación de Talento Altamente Especializado

Tuplas

- Son similares a las listas, pero son **inmutables**, lo que significa que **no se pueden modificar después de su creación.**
- Son útiles cuando quieres asegurarte de que **una secuencia de valores no cambie.**

```
tupla = (1, 2, 3, "texto", 4.5)
```

TAE IA
Formación de Talento Altamente Especializado

Conjuntos

- Son colecciones **desordenadas de elementos únicos**.
- Los conjuntos **no permiten duplicados.**
- Son útiles para **eliminar duplicados** de una colección de datos y para realizar **operaciones matemáticas como la unión y la intersección.**

```
conjunto = {1, 2, 3, 4}
```

TAE IA
Formación de Talento Altamente Especializado

Diccionario

- Son colecciones **desordenadas de pares clave-valor**, donde cada **clave única está asociada a un valor.**
- Son muy útiles cuando necesitas una r**elación de mapeo**, como una lista de nombres y sus respectivas edades.

```
diccionario = {"nombre": "Juan", "edad": 30, "ciudad": "México"}
```

TAE IA
Formación de Talento Altamente Especializado

Estructuras condicionales

TAE IA
Formación de Talento Altamente Especializado

Definición

- Son bloques de código que permiten ejecutar diferentes instrucciones o bloques de código **en función de si se cumple o no una condición específica**.
- Estas estructuras son fundamentales en la programación porque permiten **tomar decisiones** dentro del flujo de ejecución del programa.

TAE IA
Formación de Talento Altamente Especializado

Principales estructuras condicionales en Python

if: Evalúa una condición, y si esta es verdadera, ejecuta el bloque de código asociado.

```
if condicion:
    # Ejecuta codigo
```

elif: Evalúa una condición, y si esta es verdadera, ejecuta el bloque de código asociado.


```
if condicion1:
    # Código a ejecutar si condicion1 es verdadera
elif condicion2:
    # Código a ejecutar si condicion1 es falsa y condition2 es verdadera
```

TAE IA
Formación de Talento Altamente Especializado

Principales estructuras condicionales en Python

else: Se utiliza para ejecutar un bloque de código cuando ninguna de las condiciones anteriores (if o elif) es verdadera. **Es opcional y se coloca al final de la estructura condicional.**

```
if condicion1:
    # Código a ejecutar si condicion1 es verdadera
elif condicion2:
    # Código a ejecutar si condicion1 es falsa y condition2 es verdadera
else:
    # Código a ejecutar si todas las condiciones anteriores son falsas
```

TAE IA
Formación de Talento Altamente Especializado

Estructuras iterativas

TAE IA
Formación de Talento Altamente Especializado

Definición

- Son bloques de código que permiten **repetir la ejecución de un conjunto de instrucciones varias veces, basándose en una condición o una secuencia de elementos**.
- Estas estructuras son fundamentales para realizar **tareas repetitivas** de **manera eficiente** en un programa.

TAE IA
Formación de Talento Altamente Especializado

Principales estructuras iterativas en Python

for: Se utiliza para iterar **sobre una secuencia (como una lista, tupla, string, o rango)**. En cada iteración, **se toma un elemento de la secuencia y se ejecuta el bloque de código asociado.**

```
for elemento in secuencia:
    # Código a ejecutar en cada iteración
```

while: Se utiliza para repetir un bloque de código **mientras una condición sea verdadera**. A diferencia de for, **el número de iteraciones no está predeterminado; depende de cuándo se vuelva falsa la condición.**


```
while condicion:
    # Código a ejecutar mientras la condición sea verdadera
```

TAE IA
Formación de Talento Altamente Especializado

Control de flujo en estructuras iterativas

break: Termina la iteración actual y sale del bucle, **independientemente de si la condición del bucle se ha cumplido.**

```
for i in range(10):
    if i == 5:
        break
    print(i)
```

```
Output
0
1
2
3
4

=== Code Execution Successful ===
```

TAE IA
Formación de Talento Altamente Especializado

Control de flujo en estructuras iterativas

continue: Salta la iteración actual y pasa a la siguiente, sin terminar el bucle completo

```
for i in range(5):
    if i == 2:
        continue
    print(i)
```

```
Output
0
1
3
4

=== Code Execution Successful ===
```

TAE IA
Formación de Talento Altamente Especializado

Se puede utilizar Python desde varias fuentes:

- Instalar Python directamente en tu PC
- Instalar un entorno virtual para utilizar Python en él
- Instalar un contenedor virtual como Docker para usar Python en él
- Instalar Anaconda Navigator e internamente utilizar una de sus aplicaciones independientes, como Jupyter Notebook
- Utilizar Google Colaboratory en la nube de Google Drive

Google colab

TAE IA
Formación de Talento Altamente Especializado

Empecemos a utilizar Python

¡Vamos a probar!

TAE IA
Formación de Talento Altamente Especializado

Funciones en Python

TAE IA
Formación de Talento Altamente Especializado

Definición

- En Python, una función es un **bloque de código reutilizable** que realiza **una tarea específica**.
- Las funciones permiten **agrupar un conjunto de instrucciones bajo un mismo nombre, facilitando la reutilización y organización del código.**

TAE IA
Formación de Talento Altamente Especializado

¿Cómo se crea y utiliza una función?

- Para definir una función en Python, se utiliza la palabra clave **def**, seguida del **nombre de la función, paréntesis y dos puntos.**
- Dentro de los paréntesis, se pueden incluir **parámetros** que la función puede aceptar, estos pueden ser **obligatorios y opcionales**.
- El código de la función se escribe dentro de un bloque indentado.
- Es una buena práctica devolver el resultado de la función utilizando la palabra reservada **return.**
- Opcionalmente se puede colocar una flecha con el tipo de dato que devuelve la función.

TAE IA
Formación de Talento Altamente Especializado

```python
def sumar(primer_numero, segundo_numero = 2)-> int:
  resultado = primer_numero + segundo_numero
  return resultado
```

- Para llamar a la función creada simplemente se utiliza el nombre de la misma asignada a una variable la cual recuperará el valor devuelto por ella.
- En el ejemplo ilustrado la función recibe primer_numero como valor obligatorio mientras que el segundo_numero puede ser enviado como no, de no ser enviado se asignará el valor por defecto igual a 2.


```python
prueba_uno = sumar(17,5)
print(prueba_uno)
```

```
22
```

```python
prueba_dos = sumar(17)
print(prueba_dos)
```

```
19
```

TAE IA
Formación de Talento Altamente Especializado

Recordando el concepto de ámbitos

En Python, **el ámbito local** se refiere a las variables definidas dentro de una función, las cuales solo son accesibles dentro de esa función. Por ejemplo, una variable definida dentro de una función no puede ser usada fuera de ella.

El ámbito global, por otro lado, se refiere a las variables definidas fuera de cualquier función, las cuales pueden ser accedidas y modificadas desde cualquier parte del código. **Si necesitas modificar una variable global dentro de una función, debes usar la palabra clave global para indicar que estás refiriéndote a la variable global y no creando una nueva variable local.**

TAE IA
Formación de Talento Altamente Especializado

```python
variable_global = 100  # Variable global

def funcion_local():
    variable_local = 50  # Variable local
    print("Variable local dentro de la función:", variable_local)

def funcion_global():
    global variable_global
    variable_global = 200  # Modificando la variable global
    print("Variable global dentro de la función:", variable_global)

funcion_local()
print(variable_local)
```


```
ERROR!
Variable local dentro de la función: 50
Traceback (most recent call last):
  File "<main.py>", line 13, in <module>
NameError: name 'variable_local' is not defined

=== Code Exited With Errors ===
```

TAE IA
Formación de Talento Altamente Especializado

```python
variable_global = 100  # Variable global

def funcion_local():
    variable_local = 50  # Variable local
    print("Variable local dentro de la función:", variable_local)

def funcion_global():
    global variable_global
    variable_global = 200  # Modificando la variable global
    print("Variable global dentro de la función:", variable_global)

funcion_global()
print("Variable global después de modificarla:", variable_global)
```


```
Variable global dentro de la función: 200
Variable global después de modificarla: 200

=== Code Execution Successful ===
```

TAE IA
Formación de Talento Altamente Especializado

Código práctico sobre funciones en Python

¡Vamos a probar!

TAE IA
Formación de Talento Altamente Especializado

Módulos en Python

TAE IA
Formación de Talento Altamente Especializado

Definición

- En Python, un módulo es un **archivo** que contiene declaraciones y funciones.
- Los módulos permiten **organizar el código en archivos separados**, facilitando la reutilización y la gestión del código en proyectos grandes.
- Cada archivo Python **.py** es un módulo

TAE IA
Formación de Talento Altamente Especializado

Código práctico sobre módulos en Python

Básicamente se trata de una extensión más avanzada referente al uso de constantes, variables y funciones ¡Vamos a probar!

TAE IA
Formación de Talento Altamente Especializado

ACTIVIDAD

1. Cree un programa que convierta una determinada cantidad de dinero de euros a dólares estadounidenses (USD) y viceversa
2. Cree un programa que calcule el producto punto (escalar) entre dos vectores
3. Cree un módulo que realice la multiplicación de dos matrices y úselo

TAE IA
Formación de Talento Altamente Especializado

Introducción a NumPy para computación numérica

TAE IA
Formación de Talento Altamente Especializado

Definición

- Es una librería de Python que proporciona soporte para **arrays multidimensionales y matrices**, **junto con una colección de funciones matemáticas para realizar cálculos eficientes con estos arrays**.

- Se ha vuelto muy popular en la ciencia de datos, el aprendizaje automático y muchas otras aplicaciones de análisis numérico y científico.

NumPy

TAE IA
Formación de Talento Altamente Especializado

Gracias por su atención
¿Dudas?

TAE IA
Formación de Talento Altamente Especializado