

¿Qué lenguaje de programación aprender?

¿Qué lenguaje de programación aprender?, deberá cambiarse a ¿Qué quiero lograr?

1. Construir un sitio web
2. Construir una aplicación
3. Construir un videojuego
4. Etc

Especialidades en el campo de la tecnología

1. Frontend
2. Backened - *Python*
3. Internet de las cosas - *Python*
4. Inteligencia Artificial - *Python*
5. DevOps
6. Data Science - *Python*
7. Videojuegos
8. Desarrollo móvil

¿Por qué aprender Python?

Python es un lenguaje de programación creado por *Guido van Rossum* a principios de los años 90 cuyo nombre está inspirado en el grupo de cómicos ingleses "Monty Python".

Python es lenguaje de programación multiparadigma, interpretado, dinámico y **multiplataforma**.

Soporta:

1. programación imperativa
2. programación, orientación a objetos.
3. programación funcional.

Multiplataforma

El intérprete de Python está disponible en multitud de plataformas:

1. Unix
2. Solaris
3. Linux
4. DOS
5. Windows
6. OS/2
7. Mac OS
8. Etc

Por lo que si no utilizamos librerías específicas de cada plataforma, nuestro programa podrá correr en todos estos sistemas sin grandes cambios.

Algoritmo

Es una lista de pasos, procedimientos o acciones ordenados que nos permiten resolver un problema o alcanzar un resultado.

Características de los algoritmos:

1. Precisión (No ambiguo)
2. Determinismo
3. Finitud
4. Elementos de entrada
5. Un dominio y rango bien definidos

Consola

No necesitamos la interfaz gráfica de nuestra computadora para poder usarla. Nuestra mejor herramienta es la consola. La consola nos permite comunicarnos con el computador por medio de comandos y así realizar tareas sin la necesidad de utilizar el mouse o una interfaz grafica.

Comandos básicos para usar en la consola:

- Ctrl + L = Limpiar pantalla (clear)
- CD = Change Directory
- ... = Carpeta padre
- CD... = Cambiar de directorio a la carpeta padre
- Alt + 92 = \
- ls = list

- mkdir = Make directory
- touch = para crear archivos

Operadores Aritméticos Python

Para iniciar la consola interactiva de Python debemos escribir el comando `**py **` en Windows, pero en otros sistemas el comando es `python3`.

En la consola nos permite escribir operaciones matemáticas como `1 + 1` sin escribir nada más, pero en el editor de código debemos *imprimir* el resultado, de la siguiente manera:

```
print(5 + 5). Con esto obtendremos el resultado.
```

Operadores Aritméticos en Python

- Suma: `1 + 2`
- Resta: `3 - 4`
- Multiplicación: `3 * 4`
- División (con decimales): `5 / 5`
- División (sin decimales): `21 // 5`
- Módulo: `25 % 7`
- Potencia: `2 ** 2`

Raíz cuadrada:

```
import math
math.sqrt( 9 )
3.0
math.sqrt( 11.11 )
3.3331666624997918
math.sqrt( Decimal('6.25') )
2.5
```

Ejecutar todas estas operaciones en la consola de python

Jerarquía de operadores aritméticos en python

1. Paréntesis
2. Exponentes o raíces
3. Multiplicaciones o divisiones
4. Sumas y restas
5. Asignacion

Variables

Una variable es un lugar en memoria (una caja) en el que podemos guardar objetos (números, texto, etc). Esta variable posee un identificador o nombre con el cual podemos llamarla para guardar y recuperar información.

Asignación de variables

Se crea una variable asignándole un valor:

```
<identificador> = <valor>

# Ejemplo en consola python
>>> numero1 = 11
>>> numero2 = 7
>>> numero3 = numero1 + numero2
```

Reasignación de variables

Podemos en cualquier momento cambiar el valor de nuestra variable volviendo a asignar un valor al mismo identificador.

```
<identificador> = <nuevo_valor>
```

Reglas para identificadores en python

- No pueden empezar con un número.
- Deben estar en **minúsculas**.
- Separamos las palabras usando guion bajo: `_`.

Tipos de Datos en Python

- Tipo texto: `str`
- Tipo numericos: `int`, `float`, `complex`
- Secuenciales: `list`, `tuple`, `range`
- Mapeos: `dict`
- Conjuntos: `set`, `frozenset`
- Booleanos: `bool`
- Binary Types: `bytes`, `bytearray`, `memoryview`
- None Type: `NoneType`

Ejemplos de data type

```
# Tipos de variables en Python

# int (entero)
a = 28

# float (decimales o reales)
b = 1.5

# str (string o cadena de texto)
c = "Hello"
c = 'Hello'

# boolean (verdadero o falso)
d = True

# NoneType (Sin valor)
e = None

# str (5 y "5" no son lo mismo. La primera es un entero y la segunda una cadena de texto)
f = "5"
```

Datos primitivos python

Un **objeto** es una forma de modelar el mundo en programación. En los lenguajes de programación se caracterizan por tener métodos y atributos. En Python todo es un objeto.

Podemos encontrar cuatro tipos de datos que vienen definidos por defecto en Python, a estos tipos de datos los conocemos como primitivos. Tipos de datos primitivos en Python

```
Integers: números Enteros
Floats: números de punto flotante (decimales)
Strings: cadena de caracteres (texto)
Boolean: booleanos (Verdadero o Falso)
```

Algunos operadores aritméticos pueden funcionar para operar con otros tipos de datos. Por ejemplo: podemos sumar strings, lo que concatena el texto o multiplicar un entero por un string, lo que repetirá el _string _las veces que indique el entero.

¿Cómo saber el tipo de dato que estoy usando?

Usando el comando `type`

```
## Resultado de consola
>>> x = 5
>>> type(x)
<class 'int'>
```

Ejecutar lo siguiente con consola

```
# Sumar dos textos(string)
>>> nombre1 = "Edgar Erik"
>>> nombre2 = "Andrés Urbano"
>>> nombre1 + nombre2
'Edgar ErikAndrés Urbano'

# Multiplicar un texto
>>> nombre1 * 4
'Edgar ErikEdgar ErikEdgar ErikEdgar Erik'

# Formateando texto
>>> nombre1 + ", " + nombre2
'Edgar Erik, Andrés Urbano'
```

Convertir un dato a un tipo diferente

Cómo convertir un tipo de dato a otro en Python:

Sintaxis

- `int(var)` variable a entero
- `float(var)` variable a flotante
- `str(var)` variable a texto
- `bool(var)` variable a booleano
- `abs(var)` variable a valor absoluto

Ejemplo de convesión de datos en Python

Ejemplo 1

```
>>> number1 = input("Escribe un número: ")
Escribe un número: 4
>>> number2 = input("Escribe otro número: ")
Escribe un número: 5
>>> numero1 + numero 2
=> '45' <== Se concatenan
```

Ejemplo 2

```
>>> number1 = int(input("Escribe un numero: "))
Escribe un numero: 100
>>> number2 = int(input("Escribe otro numero: "))
Escribe otro numero: 300
>>> number1 + number2
=> 400
```

Ejemplo 3

```
>>> numero1 = 4.5
int(numero1)
=> 4 <== Trunca el flotante
```

Ejemplo 4

```
>>> numero1 = 4.5  
str(numero1)  
=> '4.5' <== Lo convierte a texto
```

Operadores lógicos y de comparación(relacionales) en Python

Operadores lógicos

- and (y)
- or (ó)
- not (no)

Operadores Relacionales

- == - (igual qué)
- != - (diferente de)
- > - (mayor que)
- < - (menor que)
- >= - (mayor o igual)
- <= - (menor o igual)

Ejemplo operadores relacionales


```
# Operadores Lógicos
>>> es_estudiante = True
>>> trabaja = False
>>> es_estudiante and trabaja
False
>>> es_estudiante or trabaja
True
>>> not es_estudiante
False

# Operadores Relacionales
```

Programa en Python

```
pesos = input("¿Cuántos pesos mexicanos tienes?: ")
pesos = float(pesos)

valor_dolar = 21.80
dolares = pesos / valor_dolar
dolares = round(dolares,2)

dolares = str(dolares)
print("Tienes $" + dolares + " dólares")
```