

**INFORME SOBRE PYTHON, CONOCER SU DEFINICION HASTA QUE ES
UNA LISTA Y SUS METODOS DE EMPLEACION**

Alejandro Alfaro

1 A, Nocturna, Programación y Algoritmos

Carrera de Desarrollo de Software

Contenido

Lenguaje de programación Python.....	4
¿Qué es Python?	4
Características	4
Su utilidad en la carrera:	4
Tipos de datos en Python.....	4
Números	4
Cadenas de texto.....	5
Booleanos.....	5
Listas	5
Tuplas.....	5
Conjuntos	5
Diccionarios	5
Su utilidad en la carrera	6
Declarar una variable.....	6
Características	6
Su utilidad en la carrera	6
Variables locales y globales.....	6
Local	6
Global.....	7

Características	7
Su utilidad en la carrera	7
Funciones y que tipos existen	7
¿Qué es una función?.....	7
Tipos de Funciones	8
Funciones integradas (built-in functions).....	8
Funciones definidas por el usuario	8
Funciones anónimas (lambda functions).....	8
Métodos	8
Características	9
Su utilidad en la carrera	9
Listas	9
¿Qué es una lista?	9
Métodos de las listas	9
Características	10
Su utilidad en la carrera	10
Conclusiones	10

Lenguaje de programación Python

¿Qué es Python?

Python es un lenguaje de programación interpretado, lo que quiere decir que su ejecución es inmediata no necesita ser compilado y después ejecutado, lo que permite un desarrollo fácil y una depuración rápida. Es muy popular actualmente gracias a su fácil sintaxis y comprensión del código siendo utilizado como método de enseñanza en las carreras como Ing. De sistemas o Desarrollo de software, además de ser un lenguaje de propósito general lo que nos permite usarlo para el desarrollo de diferentes proyectos.

Características

Es un lenguaje interpretado, fácil, rápido, dinámico con el usuario además de tener una utilidad variada con un ecosistema completo.

Su utilidad en la carrera:

Python al ser un lenguaje fácil, de una comprensión rápida y con capacidades para desarrollar diferentes proyectos ya sea en cualquier ecosistema Windows, Linux o Mac, logrando con esto un buen desempeño este lenguaje será la prioridad en el transcurso de nuestra carrera, no solo de este semestre.

Tipos de datos en Python

En Py existen diferentes tipos de datos ya sea para operar, almacenar, editar o usar en el transcurso del programa, donde a continuación listaremos algunos tipos de datos importantes, cabe recalcar que estos datos son más complejos dependiendo del programa:

Las características de cada tipo de dato se encuentran en las descripciones de los mismo.

Números

Enteros (int): representa números enteros sin decimal, por ejemplo, 2, -105, 123,0.

Flotantes (float): representa números decimales o de punto flotante, por ejemplo, 8.36, -0.54.

Cadenas de texto

String (str): representa una secuencia de letras encerrados entre comillas simples o dobles, por ejemplo, "Hola", 'Mundo', "123".

Booleanos

bool: representa un valor de verdad, que puede ser True (verdadero) o False (falso). Se utiliza para la lógica y las expresiones condicionales.

Listas

list: representa una colección ordenada y mutable de elementos. Puede contener elementos de diferentes tipos y se define entre corchetes [], separando los elementos por comas.

Tuplas

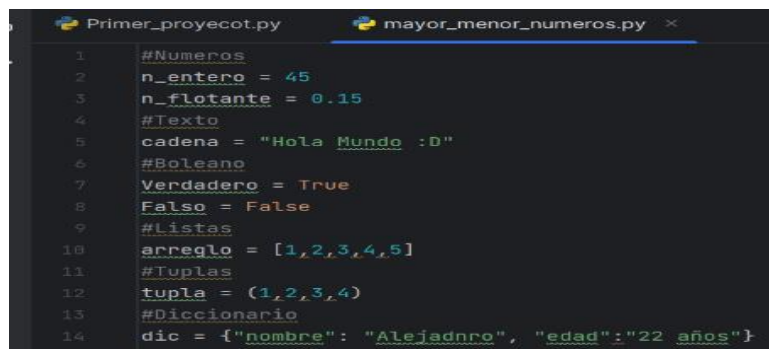
tuple: es similar a una lista, pero es inmutable, lo que significa que no se pueden modificar una vez creadas. Se define entre paréntesis () y los elementos se separan por comas.

Conjuntos

set: representa una colección no ordenada y sin elementos duplicados. Se define entre llaves {} o utilizando la función set ().

Diccionarios

dict: representa una colección de pares clave-valor. Cada elemento se compone de una clave y su correspondiente valor, y se definen entre llaves {}. Los pares clave-valor se separan por comas y se separan la clave y el valor con dos puntos (:).



```
1 #Numeros
2 n_entero = 45
3 n_flotante = 0.15
4 #Texto
5 cadena = "Hola Mundo :D"
6 #Booleano
7 Verdadero = True
8 Falso = False
9 #Listas
10 arreglo = [1,2,3,4,5]
11 #Tuplas
12 tupla = (1,2,3,4)
13 #Diccionario
14 dic = {"nombre": "Alejandrro", "edad": "22 años"}
```

Su utilidad en la carrera

Conocer estos tipos de datos son muy importantes ya que son lo básico de Py y sin ellos no podremos definir bien nuestro programa y por conclusión este fallara.

Declarar una variable

En Py la declaración de una variable o su definición es sencilla ya que solo se usa el símbolo igual (=) el cual es utilizado en la asignación de un valor a una variable, esto es simple gracias a que Py ya con el parámetro del dato que le demos este comprenderá si es entero, booleano, flotante, etc.

```
mayor_menor_numeros.py x
1 # Asignación de un valor entero a una variable llamada "edad"
2 edad = 25
3
4 # Asignación de una cadena de texto a una variable llamada "nombre"
5 nombre = "Juan"
6
7 # Asignación de un valor decimal a una variable llamada "precio"
8 precio = 9.99
9
10 # Asignación de un valor booleano a una variable llamada "activo"
11 activo = True
```

Características

El ser un lenguaje de fácil uso, este comprenderá que tipo de variable vamos a operar con solo asignarle un dato cualquiera.

Su utilidad en la carrera

Una vez que comprendamos que tipo de datos tenemos es momento de comprender como usarlos y como asignarlos a una variable por lo que este tema es muy importante.

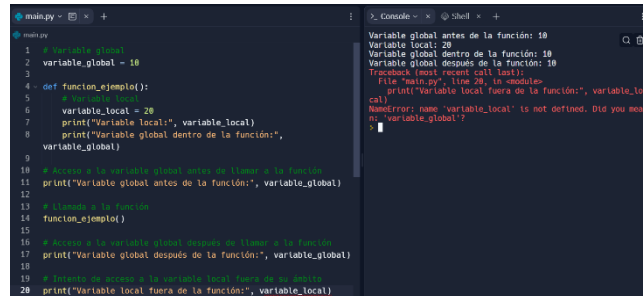
Variables locales y globales

Local

Es una variable que se declara dentro de una función o bloque de código específico por lo que solo se puede acceder a esa variable solo dentro de ese bloque de código esto limita su alcance al bloque de código donde fue creada.

Global

Es una variable que se declara fuera de cualquier función o bloque de código y que esta disponible en cualquier momento del programa permitiendo que su alcance sea superior.



```
main.py | Console | Shell  
1 # Variable global  
2 variable_global = 10  
3  
4 def funcion_ejemplo():  
5     # Variable local  
6     variable_local = 20  
7     print("Variable local:", variable_local)  
8     print("Variable global dentro de la función:",  
9         variable_global)  
10  
11 # Antes de la función antes de llamar a la función  
12 print("Variable global antes de la función:", variable_global)  
13  
14 # Llamada a la función  
15 funcion_ejemplo()  
16  
17 # Después a la variable global después de llamar a la función  
18 print("Variable global después de la función:", variable_global)  
19  
20 # Después de volver a la variable local fuera de la función  
21 print("Variable local fuera de la función:", variable_local)
```

```
Variable global antes de la función: 10  
Variable local: 20  
Variable global dentro de la función: 10  
Variable global después de la función: 10  
Traceback (most recent call last):  
  File "main.py", line 28, in <module>  
    print("Variable local fuera de la función:", variable_lo  
    cal)  
NameError: name 'variable_local' is not defined. Did you mea  
n 'variable_global'?
```

Características

Las variables locales solo se encuentran disponibles dentro de la función donde fueron declaradas y su alcance es limitado.

Las variables globales se definen fuera de una función y están disponibles en cualquier momento del programa teniendo un alcance mayor.

Su utilidad en la carrera

Es muy importante en la carrera ya que este tipo de variables nos permitirá crear software de un nivel mas profesional.

Funciones y que tipos existen

¿Qué es una función?

En Python, una función es un bloque de código reutilizable que se puede llamar para realizar una tarea específica. Proporciona una forma de agrupar instrucciones relacionadas y ejecutarlas cuando sea necesario. Una función puede aceptar argumentos (valores de entrada), realizar ciertas operaciones y, opcionalmente, devolver un resultado.

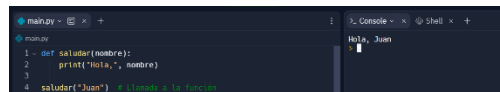
Tipos de Funciones

Funciones integradas (built-in functions)

Son funciones predefinidas en Python que están disponibles para su uso sin necesidad de importar ningún módulo adicional. Algunos ejemplos comunes son `print()`, `len()`, `input()`.

Funciones definidas por el usuario

Son funciones que se definen por el programador para realizar tareas específicas. Se crean utilizando la palabra clave `def`, seguida del nombre de la función, paréntesis para los argumentos y dos puntos (:). Pueden tener argumentos opcionales y pueden devolver un valor utilizando la palabra clave `return`. Aquí tienes un ejemplo:



```
main.py - x +
1 - def saludar(nombre):
2   print("Hola", nombre)
3
4 saludar("Juan") # Llama a la función
```

Console - x Shell x +
Hola, Juan

Funciones anónimas (lambda functions)

Son funciones pequeñas y sin nombre que se definen utilizando la palabra clave `lambda`. Estas funciones pueden tomar múltiples argumentos, realizar una operación y devolver un valor sin necesidad de una declaración `return`. Se utilizan principalmente en situaciones donde se requiere una función rápida y simple. Aquí tienes un ejemplo:

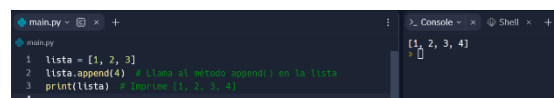


```
main.py
1 multiplicar = lambda x, y: x * y
2 resultado = multiplicar(5, 3)
3 print(resultado) # Imprime 15
```

15

Métodos

Son funciones que están asociadas a un objeto específico y se llaman en ese objeto utilizando la sintaxis `objeto.Método()`. Los métodos son comunes en Python para realizar operaciones en objetos y realizar acciones específicas. Un ejemplo común es el método `append()` de las listas para agregar elementos.



```
main.py - x +
1 lista = [1, 2, 3]
2 lista.append(4) # Llama al método append() en la lista
3 print(lista) # Imprime [1, 2, 3, 4]
```

Console - x Shell x +
[1, 2, 3, 4]

Características

Cada tipo tiene su propia utilidad y se utiliza en diferentes situaciones según las necesidades del programa. Las cuales les dota de ciertas utilidades en ciertos momentos.

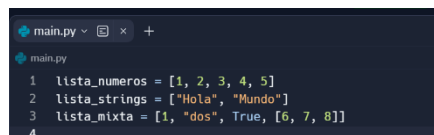
Su utilidad en la carrera

Conocer sobre funciones es muy importante ya que con ellas nuestro código será mas comprensible, más rápido de ejecutar y que sea mas profesional.

Listas

¿Qué es una lista?

En Python, una lista es una estructura de datos que permite almacenar y organizar una colección ordenada de elementos. Puedes pensar en ella como una secuencia mutable de elementos, donde cada elemento tiene una posición o índice único. Las listas en Python se definen utilizando corchetes `[]`, y los elementos se separan por comas. Pueden contener elementos de diferentes tipos, como números, cadenas de texto, booleanos u otras listas.



```
main.py x +
main.py
1 lista_numeros = [1, 2, 3, 4, 5]
2 lista_strings = ["Hola", "Mundo"]
3 lista_mixta = [1, "dos", True, [6, 7, 8]]
4
```

Métodos de las listas

Una vez que tienes una lista, puedes aplicar una variedad de métodos para manipular y acceder a sus elementos.

Append (elemento): agrega un elemento al final de la lista.

Insert (índice, elemento): inserta un elemento en la posición especificada.

Extend (iterable): agrega los elementos de otro iterable (como otra lista) al final de la lista actual.

Remove (elemento): elimina la primera aparición del elemento especificado en la lista.

Pop ([índice]): elimina y devuelve el elemento en la posición especificada. Si no se proporciona un índice, se elimina y devuelve el último elemento.

Index (elemento): devuelve el índice de la primera aparición del elemento en la lista.

Count (elemento): devuelve el número de veces que el elemento aparece en la lista

```
1 frutas = ["manzana", "naranja", "plátano"]
2
3 frutas.append("kiwi")
4 print(frutas)
5
6 frutas.insert(1, "pera")
7 print(frutas)
8
9 frutas.remove("naranja")
10 print(frutas)
11 print(frutas.index("plátano"))
12
13 print(len(frutas))
```

```
['manzana', 'naranja', 'plátano', 'kiwi']
['manzana', 'pera', 'naranja', 'plátano', 'kiwi']
4
['manzana', 'pera', 'plátano', 'kiwi']
```

Características

Las características principales de las listas es que con ellas podemos realizar ciertas operaciones que parecen complejas como contraseñas o claves de cifrado.

Su utilidad en la carrera

Las listas son muy importantes en el ámbito de Py y conocer sus métodos para saberlas utilizar es muy importante ya que esto nos permite crear y desarrollar nuevos proyectos mas complejos.

Conclusiones

Python es un lenguaje de programación ampliamente utilizado y versátil que se destaca por su simplicidad y legibilidad, cuenta con diferentes tipos de datos, como enteros, flotantes, cadenas de texto, booleanos, listas, entre otros, que permiten almacenar y manipular información de diferentes formas. Las variables en Python se pueden declarar simplemente asignándoles un valor y no es necesario especificar su tipo, ya que Python es un lenguaje de tipado dinámico. Las variables pueden ser globales o locales. Las variables globales se declaran fuera de cualquier función y están disponibles en todo el programa, mientras que las variables locales se declaran dentro de una función y solo son accesibles dentro de esa función. a los objetos.