

Curso Python

Cristian Segura L.

2024-04-29

Índice general

Capítulo 1: Introducción a Python	3
Computadores y Lenguajes de Programación	3
Historia del Lenguaje de Programación Python	3
Capítulo 2: Instalación y Primeros Pasos	4
Instalación en Ubuntu 22.04	4
Instalar Python 3 desde la consola	4
Mi Primer Programa en Python3 ('Hello World!')	4
Capítulo 3: Tipos de Variables	5
3.1 Tipos Simples: Números Enteros (Integer)	5
3.1.A Número Enteros en Matemáticas vs Python**	5
3.1.B Operadores para Enteros	5
3.1 Tipos Simples: Booleanos (Boolean)	6
Capítulo 4: Datos complejos	7
Capítulo de Aplicación de Python3	8
SSH con Python con Paramiko	8
Referencias	8

Capítulo 1: Introducción a Python

Computadores y Lenguajes de Programación

Historia del Lenguaje de Programación Python

Capítulo 2: Instalación y Primeros Pasos

Instalación en Ubuntu 22.04

Instalar Python 3 desde la consola

PASO 1: Abrir la consola (shell)

PASO 2: Instalar Python 3 y PIP con el comando

```
$ sudo apt install python3 python3-pip
```

PASO 3: Corroborar la versión de Python instalada

```
$ python3 -version  
Python 3.10.12
```

Mi Primer Programa en Python3 ('Hello World!')

PASO 1: Crear directorio python3 dentro del directorio home del usuario que estemos usando. En este directorio realizaremos todas las pruebas que se muestran en este curso.

```
# Ir al directorio home  
$ cd ~  
# Corroboramos en dónde nos encontramos (típicamente en /home/usuario)  
$ pwd  
# Creamos el directorio `python3`  
$ mkdir python3
```

PASO 2: Dentro directorio python3 creamos el archivo hello.py

```
# Entramos al directorio python3  
$ cd ~/python3
```

```
# Creamos el archivo `hello.py`  
$ touch hello.py
```

PASO 3: Con algún editor de texto (gedit, nano, vim, etc.) abrimos el archivo hello.py y agregamos el siguiente código fuente Python:

```
print('Hello World!')
```

Guardamos el archivo

PASO 4: Dentro de la consola ejecutamos el programa Python que acabamos de crear con el comando

```
# Ejecutamos el script hello.py  
$ python3 hello.py
```

Capítulo 3: Tipos de Variables

Tipos simples:

- Tipos Numéricos: Enteros, Flotantes
- Tipo Boolean

3.1 Tipos Simples: Números Enteros (Integer)

3.1.A Número Enteros en Matemáticas vs Python**

Número Enteros en Matemáticas

- Números Naturales: $N = 1, 2, 3, 4, \dots$
- Números Enteros: $Z = \dots, -2, -1, 1, 2, 3, 4, \dots$
 - Enteros Positivos: $Z^+ = 1, 2, 3, \dots$
 - Enteros Negativos: $Z^- = -1, -2, -3, \dots$
 - Enteros Positivos incluyendo el Cero $Z_0^+ = 0, 1, 2, 3, \dots$
 - Enteros Positivos incluyendo el Cero $Z_0^- = 0, -1, -2, -3, \dots$

Números Enteros en Python

- Números positivos, Negativos y Cero
- Valores máximo positivo y mínimo negativo

Ejemplo 1

```
# Crear una variable de tipo Integer
>>> num=15
# Averiguando el tipo de una variable Integer
>>> print(type(num))
<class 'int'>
# Valor Máximo de Número Entero
>>> import sys
>>> sys.maxsize
9223372036854775807
# max integer = 9.223.372.036.854.775.807
# min integer = -sys.maxsize - 1
# -9223372036854775808
```

3.1.B Operadores para Enteros

Operadores para Enteros

- Suma (+)
- Resta (-)
- Multiplicación (*)
- División Entera (/)

Ejemplo 2

```
# Creamos un de numeros enteros
>>> numa = 15
>>> numb = 7
# Operación suma
>>> numa + numb
22
# Operación resta
>>> numa - numb
8
# Operación Producto
>>> numa * numb
105
# Operación División
>>> numa / numb
2.142857142857143
```

Ejemplo 3

```
# Cuidado con las divisiones
>>> numc = numa / numb
>>> numc
2.142857142857143
>>> type(numa)
<class 'int'>
>>> type(numb)
<class 'int'>
# Una división de números enteros en python puede producir un
# resultado de otro tipo de dato
>>> type(numc)
<class 'float'>
```

Ejemplo: Ecuación de segundo grado

Una ecuación de Segundo Grado de la forma:

$$ax^2 + bx + c = 0$$

Tiene las soluciones

$$x_1 = \frac{b^2 + \sqrt{\Delta}}{2a}$$
$$x_2 = \frac{b^2 - \sqrt{\Delta}}{2a}$$

donde Δ es el discriminante que se calcula

$$\Delta = b^2 - 4ac$$

El algoritmo de funcionamiento del programa Python es:

- Si el Discriminante Δ es mayor que cero se calculan las dos raíces de la ecuación.
- Si el Discriminante Δ es cero, se calcula una raíz
- Si el Discriminante Δ es menor que cero, no se calcula una raíz y se imprime un mensaje que dice: la ecuación no tiene soluciones reales

```
import math
```

```
# Ecuación Segundo Grado  $ax^2 + bx + c = 0$ 
```

```
a = 1
```

```
b = 4
```

```
c = 3
```

```
# discriminante de la ecuación
```

```
disc = b**2 - 4*a*c
```

```
# Solución 1 (x_1)
```

```
x_1 = (b**2 - math.sqrt(disc))/(2*a)
```

```
# Solución 2 (x_2)
```

```
x_2 = (b**2 + math.sqrt(disc))/(2*a)
```

Referencias

- <https://en.wikipedia.org/wiki/Integer>
- <https://docs.python.org/3/library/stdtypes.html>

3.1 Tipos Simples: Booleanos (Boolean)

Capítulo 4: Datos complejos

Capítulo de Aplicacion de Python3

SSH con Python con Paramiko

- Ejemplo de conexión SSH simple. Se realiza una conexión remota vía SSH al servidor especificado por la dirección IP. Una vez establecida la conexión SSH se ejecuta el comando df en la máquina remota.

```
import paramiko

command = "df"

# Update the next three lines with your
# server's information

host = "YOUR_IP_ADDRESS"
username = "YOUR_LIMITED_USER_ACCOUNT"
password = "YOUR_PASSWORD"

client = paramiko.client.SSHClient()
client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
client.connect(host, username=username, password=password)
_stdin, _stdout, _stderr = client.exec_command("df")
print(_stdout.read().decode())
client.close()
```

Referencias

- <https://www.linode.com/docs/guides/use-paramiko-python-to-ssh-into-a-server/>