

Python para el análisis de datos- Lectura 2

Ing. Pedro Rotta

Universidad de Piura - Vida Universitaria

Enero-2022

¿Por qué python es un lenguaje de programación?

Python es un lenguaje de programación porque se pueden desarrollar programas que se corren sobre un servidor u ordenador.

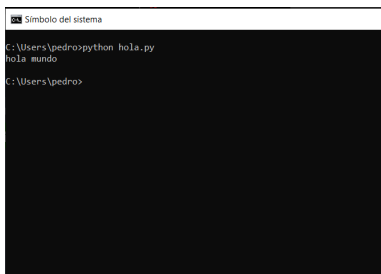
Hay dos clases de programas : De terminal y de interfaz gráfica.

Programas de terminal

Los programas de terminal como su nombre lo indica, corren sobre la consola del programa. Estos programas no tienen una interfaz de usuario, por lo que son usados netamente para investigación o en casos y automatización interna.

Programas de terminal

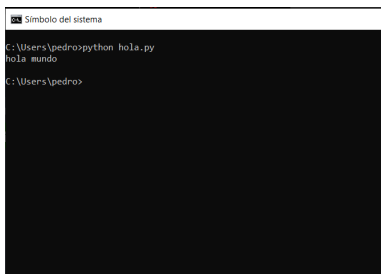
Los programas de terminal como su nombre lo indica, corren sobre la consola del programa. Estos programas no tienen una interfaz de usuario, por lo que son usados netamente para investigación o en casos y automatización interna.



```
Símbolo del sistema
C:\Users\pedro>python hola.py
hola mundo
C:\Users\pedro>
```

Programas de terminal

Los programas de terminal como su nombre lo indica, corren sobre la consola del programa. Estos programas no tienen una interfaz de usuario, por lo que son usados netamente para investigación o en casos y automatización interna.



```
Símbolo del sistema
C:\Users\pedro>python hola.py
hola mundo
C:\Users\pedro>
```

Programas de interfaz gráfica

Los programas de interfaz son los más conocidos por los usuarios y son los programas que se corren en escritorio o en web. Son usados como MVPs o como un producto de automatización más complejo.

¿Qué es python?

Python es un lenguaje de programación **orientado en objetos** que permite crear programas multipropósito. En este curso veremos el uso de python enfocado al área de análisis de datos. En dónde exploraremos varios ejemplos enfocados a distintas ramas científicas: Economía, Ingeniería, entre otras.

Qué es un objeto

Un *objeto* o *clase* en python es un elemento de programación que tiene **propiedades** y **métodos**. En las lecturas usaremos clases u objetos indistintamente para definir lo mismo.

Si comparamos con el mundo real por ejemplo, las personas somos objetos, los árboles son objetos, en fin todo lo que nos rodea puede ser llamado "**objeto**".

Qué es un objeto

Un *objeto* o *clase* en python es un elemento de programación que tiene **propiedades** y **métodos**. En las lecturas usaremos clases u objetos indistintamente para definir lo mismo.

Si comparamos con el mundo real por ejemplo, las personas somos objetos, los árboles son objetos, en fin todo lo que nos rodea puede ser llamado "**objeto**".

Los objetos más simples en python son 3: Flotantes, cadenas de texto y objetos booleanos.

Sintáxis y semántica

Si comparamos con el idioma, un lenguaje de programación tiene una sintáxis que se debe respetar y una semántica que se debe respetar. Muchos de los errores al principio son de sintáxis, por lo que es una parte esencial de un lenguaje de programación.

La **sintáxis** hace referencia a cómo se escriben los *comandos* dentro del lenguaje de programación, es por eso que al correr :

```
prints (" hola mundo" )
```

saldrá un error de sintaxis.

Sintaxis y semántica

La semántica tiene que ver con la lógica que debe existir en los comandos de Python. Algunas reglas semánticas son las siguientes:

- ▶ No se pueden multiplicar cadenas de texto
- ▶ No se puede sumar una cadena de texto y un número
- ▶ Puede multiplicar una cadena de texto por un número y el resultado será la cadena de texto repetida ese número de veces

Cadenas de Texto

Hablaremos de las propiedades básicas de las cadenas de texto. Sirven en primer lugar para comunicarse con el usuario de un programa.

En el primer programa que desarrollaremos de python en este curso vamos a enviar un mensaje a la consola al correr el programa. Este programa es el más sencillo de hacer en Python, pero cubre algunas cosas básicas que hay que tener en cuenta. (Ver colab.)

```
print(" hola  mundo" )
```

Cadenas de Texto

Un comando importante que permite introducir cadenas de texto como una **variable** es *input*.

Cadenas de Texto

Un comando importante que permite introducir cadenas de texto como una **variable** es *input*.

Una variable es un objeto de valor único. Por ejemplo una persona con nombre como Pedro, es una variable porque toma un valor específico.

Cadenas de Texto

Un comando importante que permite introducir cadenas de texto como una **variable** es *input*.

Una variable es un objeto de valor único. Por ejemplo una persona con nombre como Pedro, es una variable porque toma un valor específico.

En el código siguiente (Ver Colab.) Vemos como creamos una variable que se llama **email**.

Cadenas de Texto

Un comando importante que permite introducir cadenas de texto como una **variable** es *input*.

Una variable es un objeto de valor único. Por ejemplo una persona con nombre como Pedro, es una variable porque toma un valor específico.

En el código siguiente (Ver Colab.) Vemos como creamos una variable que se llama **email**.

```
email = input(" Ingresa tu email: ")  
print('Tu email es ', email)
```

Mira que tanto input como print son comandos que se establecen con paréntesis **()** . Dentro de los paréntesis se escriben cadenas de texto como mensaje. Estos mensajes a su vez son variables creadas inmediatamente dentro del comando input y print respectivamente.

Es bueno reconocer que las cadenas de texto se pueden escribir con comillas simples o comillas dobles. Sin importar como se escriban, ambas sintáxis dan el mismo resultado.

Números

Los números son las clases más comunes de python. Pueden ser de 2 diferentes tipos (*types*) : Enteros y flotantes.

- ▶ Enteros : Son usados para usar números enteros

Números

Los números son las clases más comunes de python. Pueden ser de 2 diferentes tipos (*types*) : Enteros y flotantes.

- ▶ Enteros : Son usados para usar números enteros
- ▶ Flotantes : Son usados para representar números racionales o irracionales.

Números

Los números son las clases más comunes de python. Pueden ser de 2 diferentes tipos (*types*) : Enteros y flotantes.

- ▶ Enteros : Son usados para usar números enteros
- ▶ Flotantes : Son usados para representar números racionales o irracionales.

Es importante reconocer que el comando `input` no permite crear variables de números por sí mismo, pues todo lo acepta como cadenas de texto. Para lograr ingresar un numero es necesario agregar el *type* del número antes del `input`.

Operaciones con Números

Las 4 operaciones básicas son permitidas por cualquier tipo de número. Se permiten hacer estas operaciones entre enteros y flotantes, dando como resultado un flotante o un entero según corresponda.

Cuando no esté seguro del tipo de número que es su variable llame el comando `type()` para poder reconocer el tipo y así evitar algún error.

En los siguientes ejemplos de Colab aprenderemos más acerca de los números