

Conociendo Python

Conociendo Python	1
¿Qué aprenderás?	2
Introducción	2
Antes de empezar: ¿Qué se requiere para este módulo?	3
¿Qué es Python?	4
Interpretado	4
Dinámico o dinámicamente tipado	4
Multiplataforma	4
Mi primer código en Python	5
Ejecución guiada 1: Mi primer código en Python	6
Salida de datos: print()	7
Comentarios	7
Comentarios en múltiples líneas	8
Python como una calculadora	8
Limitantes	9



¡Comencemos!

¿Qué aprenderás?

- Reconocer los distintos entornos de ejecución del lenguaje Python.
- Reconocer conceptos básicos del lenguaje Python (sintaxis y estructura).

Introducción

Cuando hablamos de programar, nos imaginamos un hacker sentado frente a un computador escribiendo código en una pantalla negra con letras verdes, pero en realidad, este es un concepto mucho más amplio.

En términos generales, consiste en crear un programa o software. En específico, un programa contiene instrucciones y etapas que se irán ejecutando con distintas instrucciones que dirijan al programa a llevar a cabo su cometido. Este proceso de creación es complejo, involucrando varias etapas y, quizá sea por esta complejidad, que también se le llama a este proceso "**Desarrollar**" y a los programadores "**Desarrolladores**".

Durante este módulo, se enseñan los aspectos más básicos y fundamentales para comenzar a programar. Estos conocimientos son aplicables para programar en distintos lenguajes de programación, pero en este caso **nos enfocaremos específicamente en Python**, luego se introducen contenidos más específicos sobre la programación utilizando Python.

En este capítulo, revisaremos los aspectos más importantes para poder ejecutar un código mediante el Lenguaje de programación Python. Esto será la base para las actividades y desafíos que tenemos preparados para ti.

Antes de empezar: ¿Qué se requiere para este módulo?

Para este y el resto de los módulos, se trabajará utilizando **Python versión 3.7.1 o superior** mediante **Anaconda**, y un editor de texto (incluido en el mismo Anaconda).

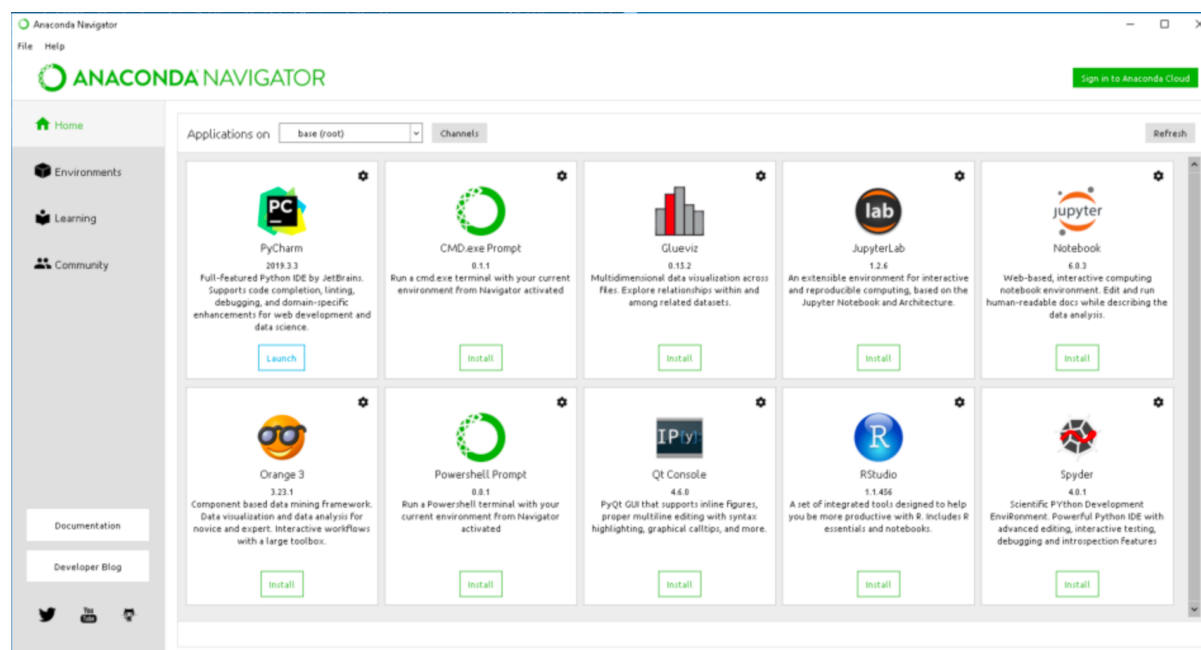


Imagen 1. Anaconda Navigator en Windows
Fuente: Desafío Latam

Anaconda está disponible para **Windows, macOS y Linux**, por lo que se requiere cualquiera de estos sistemas operativos. Sus instrucciones de instalación se pueden encontrar en la información complementaria disponible para la unidad.

¿Qué es Python?

Python es un lenguaje de programación interpretado, cuya filosofía hace hincapié en la legibilidad de su código. Se trata de un lenguaje de programación multiparadigma, ya que soporta parcialmente la **orientación a objetos**, **programación imperativa** y, en menor medida, **programación funcional**. Es un lenguaje **interpretado**, **dinámico** y **multiplataforma**, y es administrado por la Python Software Foundation y posee una licencia de código abierto.

Fue creado en 1991 por Guido van Rossum, y a pesar de ser un lenguaje con casi 30 años, tiene una fuerte presencia en la industria, principalmente en las áreas de **desarrollo web** y **Ciencia de Datos**. Para conocer detalles más específicos del lenguaje puedes leer el archivo de **información complementaria**.

A continuación, te presentamos algunas definiciones importantes para introducirnos de mejor manera al lenguaje Python:

Interpretado

Quiere decir que a diferencia de otros lenguajes, Python **no se compila**, es decir, todo su código se traduce a lenguaje que puede ser interpretado por el computador (normalmente lenguaje de máquinas). Python sólo se traduce a medida que es necesario, lo que lo hace un versátil, pero no tan veloz como los lenguajes compilados como Java, C o C++.

Dinámico o dinámicamente tipado

Es un lenguaje que permite variables que pueden tomar valores de distinto tipo.

Multiplataforma

Se refiere a que el lenguaje puede montarse en distintas plataformas como Windows, Mac, Linux, e incluso existen versiones de Python para microcontroladores como Arduino.

A continuación, te presentamos un esquema que resume las características recién mencionadas:

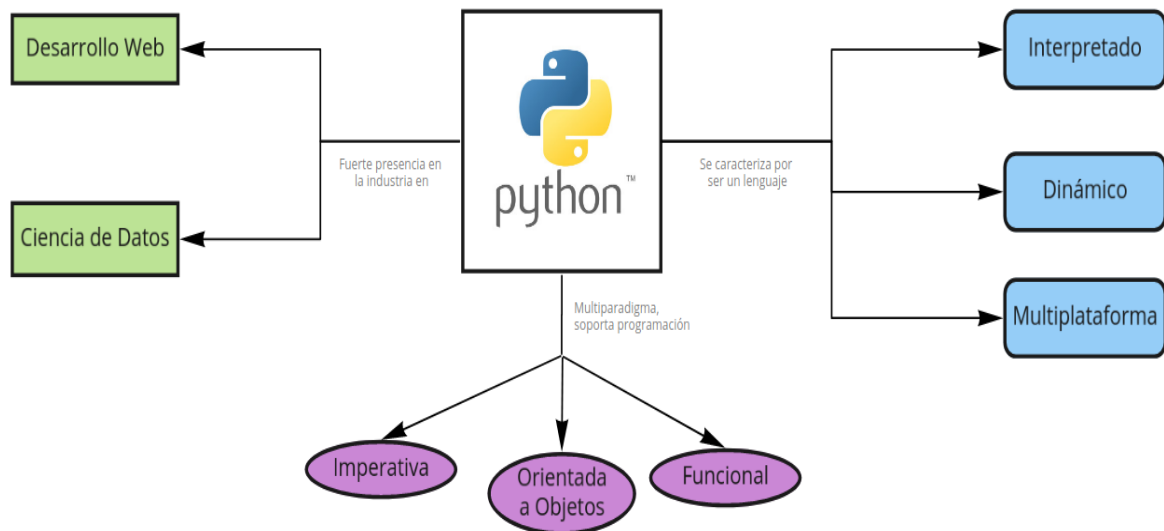


Imagen 2. Principales características de Python
Fuente: Desafío Latam

Mi primer código en Python

En todo lenguaje de programación existe el “**hola mundo**”. Esto se refiere a un código muy básico que nos permitirá mostrar el funcionamiento del lenguaje y algunos aspectos del lenguaje.

Crearemos entonces nuestro primer programa con el fin de entender cómo crear y ejecutar scripts en Python:

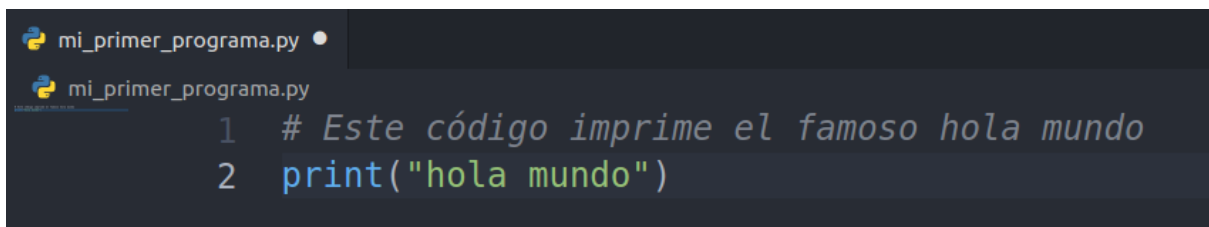
```
# Este código imprime el famoso hola mundo  
print("hola mundo")
```

Ejecución guiada 1: Mi primer código en Python

Es momento de entrar en práctica, para esto, te invitamos a revisar el siguiente ejercicio guiado.

Para escribir nuestro primer programa en el editor de texto, vamos a seguir los siguientes pasos:

1. Abrir nuestro editor de texto.
2. Escribir una instrucción sencilla. Ej. `print("hola mundo")`
3. Guardar el archivo con extensión `.py`.

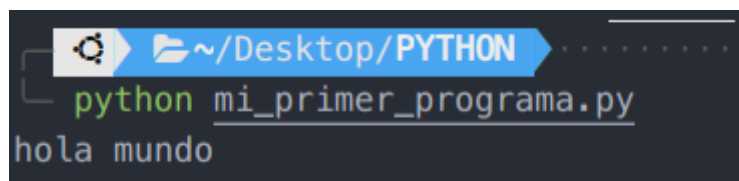


```
mi_primer_programa.py
mi_primer_programa.py
1 # Este código imprime el famoso hola mundo
2 print("hola mundo")
```

Imagen 3. mi_primer_programa.py.

Fuente: Desafío Latam.

A continuación, debemos abrir un terminal en la misma ubicación donde se encuentra el archivo y ejecutar `python mi_primer_programa.py`.



```
~/Desktop/PYTHON
python mi_primer_programa.py
hola mundo
```

Imagen 4. Sin respuesta.

Fuente: Desafío Latam.



NOTA: Para el caso particular de este script se creó una carpeta llamada PYTHON en el Escritorio.

Salida de datos: print()

print es una función para mostrar valores de variables o resultados de operaciones en la pantalla. Podemos entender inicialmente una función como un comando que Python ejecutará (definiremos formalmente las funciones en la tercera unidad de este módulo).

`"hola mundo"` es el argumento de la función `print`, y Python mostrará dicho valor en pantalla.



NOTA: Asegúrate de utilizar `print` para ver en pantalla solo los resultados que te interesan.

Comentarios

Todo texto a continuación de un signo `#` es un **comentario**. Estas son líneas de código que son ignoradas. Por ejemplo, este código imprime el famoso `hola mundo` que será una manera de explicar o documentar el código `print("hola mundo")`.

Observa el siguiente ejemplo:

```
# Esta línea es un comentario
# Los comentarios son ignorados
# Pueden ser una línea nueva
print(2 + 2) # 0 puede acompañar una línea de código existente
# print(2 + 3)
# Si comentamos un código válido en Python, no se ejecutará
```

4

En este caso, tenemos varios comentarios que explican el funcionamiento de éstos. En este código estamos sumando dos más dos, e incluso se puede ver un código válido de Python justo en la línea debajo, pero dado que está comentado, no se mostrará en pantalla su resultado.

Comentarios en múltiples líneas

Consiste en envolver todo el comentario entre tres comillas dobles al inicio y al final. Esta forma es comúnmente utilizada para **documentar funciones**.

```
"""  
Comentario multilínea:  
Python  
lo  
ignoraré  
"""
```

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

```
hola
```

Python como una calculadora

La funcionalidad más simple que se le puede dar a Python es la de una calculadora. Por defecto, **Python incluye operaciones básicas** como las sumas, restas, multiplicaciones, divisiones, entre otros.

Algunos ejemplos:

Suma:

```
print(2 + 2)
```

```
4
```

Resta:

```
print(6 - 3)
```

```
3
```

Multiplicación:

```
print(7 * 8)
```


56

División:

```
print(90 / 10)
```

10



NOTA: `print()` es sumamente importante para visualizar el resultado de la operación realizada; en caso de no utilizarlo los resultados no aparecerán en la pantalla.

Limitantes

Si bien Python se caracteriza por ser un lenguaje de programación de sintaxis simple, esto no significa que no tenga reglas para escribir el código. Por ejemplo, **Python no permite sumar letras y números**.

Cada vez que Python no sea capaz de ejecutar una instrucción que no está permitida, se mostrará algo así:

```
In [2]: "gato" + 2
-----
TypeError                                 Traceback (most recent call last)
<ipython-input-2-37334b11ca62> in <module>()
----> 1 "gato" + 2

TypeError: must be str, not int
In [3]:
```

Imagen 5. TypeError.
Fuente: Desafío Latam.

Para evitar pasar malos ratos, es muy importante entender las reglas que rigen el lenguaje Python. El error que acabamos de ver ocurre porque no entendemos uno de los primeros aspectos a estudiar: **los tipos de datos**.