

# DIPLOMADO DE PROGRAMACIÓN CON PYTHON V3

TEMA 1 SINTAXIS & JUPYTER  
NOTEBOOKS

**UNIBE**

EDUCACIÓN  
CONTINUA



Eliezer Figueroa  
MCT, MOS, MCSA, MCSE, ITIL, SCF  
3 Septiembre, 2019

## COMPONENTES DEL TEMA

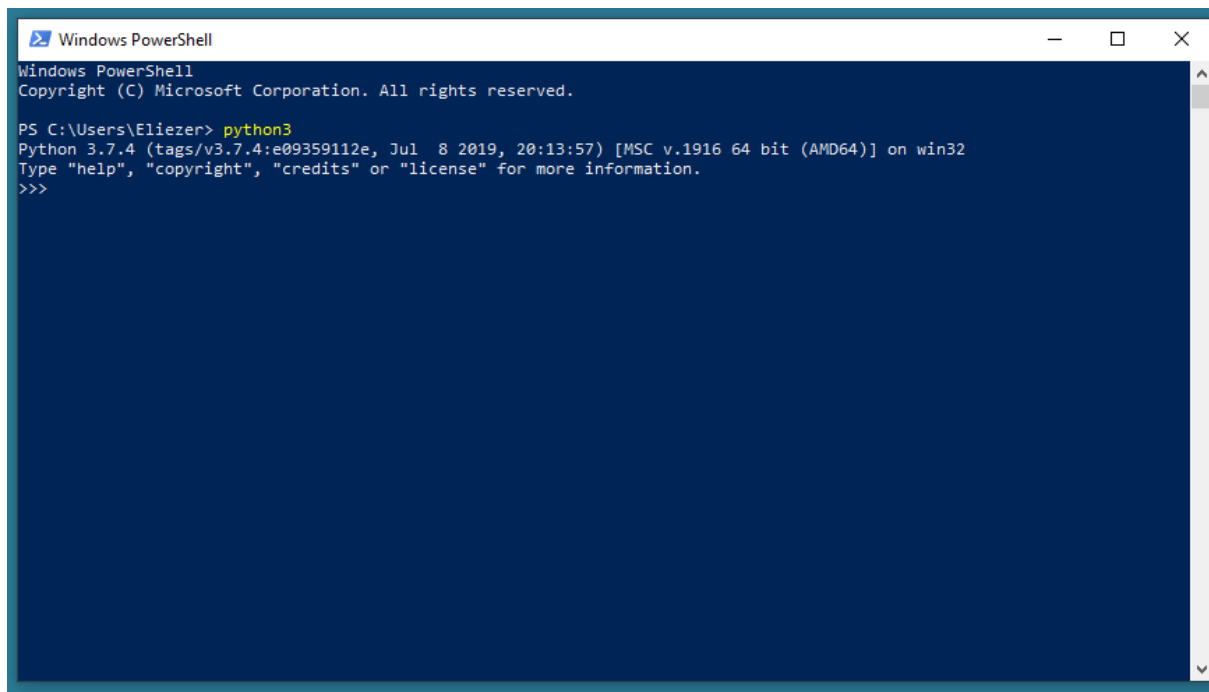
---

- ✓ INTRODUCCIÓN
- ✓ EL INTÉRPRETE DE PYTHON
- ✓ JUPYTER NOTEBOOK
- ✓ EJECUTAR SCRIPTS DE PYTHON
- ✓ SINTAXIS BÁSICA



- Python es un claro y poderoso lenguaje de programación orientado a objetos, comparable a Perl, Ruby, Scheme o Java.
- Utiliza una sintaxis elegante, que facilita la lectura de los programas que escribe.
- Es un lenguaje fácil de usar que simplifica el funcionamiento de su programa.
- Python es ideal para el desarrollo de prototipos y otras tareas de programación ad-hoc, sin comprometer la mantenibilidad.
- Viene con una gran biblioteca estándar que admite muchas tareas de programación comunes, como conectarse a servidores web, buscar texto con expresiones regulares, leer y modificar archivos.
- Actualmente hay dos versiones principales de Python disponibles: Python Versión 2 y Python Versión 3.

- El intérprete de Python generalmente se instala como `/usr/local/bin/python3.7`
- En la ruta de búsqueda de tu Shell de Unix es posible iniciarlo escribiendo el comando: `python3.7`
- En las máquinas Windows en las que haya instalado desde Microsoft Store, el comando `python3.7` estará disponible.

A screenshot of a Windows PowerShell window. The title bar reads "Windows PowerShell". The window content shows the following text:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\Eliezer> python3
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:13:57) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

- Google Colab está construido encima de Jupyter Notebook
- Python 2 y Python 3 son compatibles
- Todas las bibliotecas principales de Python como TensorFlow, Scikit-learn, Matplotlib, entre muchas otras, están preinstaladas y listas para ser importadas.

The screenshot shows a web browser window with a tab titled 'Python 3 Examples.ipynb - Colab'. The address bar shows the URL: <https://colab.research.google.com/drive/1kVeraWWYGxjoY95TcGzyxuhyCo>. The notebook interface includes a menu bar with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. Below the menu bar are buttons for '+ CODE' and '+ TEXT'. The notebook content is divided into two sections:

**Print Number As Currency**

```
[ ] amount=1234
    print('${:,.2f}'.format(amount))
```

Output: \$1,234.00

**Print Funtion Syntax Example**

```
[ ] print('year is ',2019,' month is ', 3)
```

Output: year is 2019 month is 3

- Para ejecutar los scripts de Python con el comando python, debe abrir una línea de comandos y escribir la palabra python, o python3 si tiene ambas versiones, seguidas de la ruta de acceso a su script.
- Puede usar el editor de texto de su preferencia para crear el script.

Shell

```
$ python3 hello.py  
Hello World!
```

- Un comentario es un fragmento de texto dentro de un programa que NO se ejecuta como parte del programa.
- Los comentarios se pueden utilizar para proporcionar información adicional que ayude a comprender el código.

```
# Comment on a single line  
  
user = "JDoe"    # End of line comment after code
```

- La función **print()** se usa para enviar texto, números u otra información imprimible a la consola.

```
print("Hello World!")
```

```
print(100)
```

```
pi = 3.14159
```

```
print(pi)
```

```
import math
```

```
print(math.sqrt(16))
```



- La función input() permite la entrada de datos por parte del usuario.

## Example

Ask for the user's name and print it:

```
print('Enter your name:')  
x = input()  
print('Hello, ' + x)
```

## Example

Use the prompt parameter to write a message before the input:

```
x = input('Enter your name:')  
print('Hello, ' + x)
```

- Es una secuencia de caracteres (letras, números, espacios en blanco o puntuación) entre comillas.
- Una cadena de Python se puede crear utilizando comillas dobles simple.
- Si una cadena debe dividirse en varias líneas, el carácter de barra invertida \ se puede usar para indicar que la cadena continúa en la siguiente línea.

```
user = "User Full Name"

long_line = "This string is broken up\
into multiple lines"
```

- Se utilizan variables para almacenar los datos que utilizará el programa.
- Estos datos pueden números, una cadena, un Booleano, una lista o algún otro tipo de datos.
- Cada variable tiene un nombre que puede consistir en letras, números y el carácter de subrayado \_.

```
# These are all valid variable names and assignment
user_name = "jdoe"
userID = 100
prompt21 = "Enter your user name"
existing_user = False
another_userID = userID

# A variable's value can be changed after assignment
car_weight = 1000
car_weight = 2000

# This variable name is invalid
1entry = "Test"
```

- El intérprete de Python informará los errores presentes en su código.
- Para la mayoría de los casos de error, el intérprete mostrará la línea de código e inmediatamente debajo del código, mostrará una línea con el carácter de intercalación ^ debajo de la parte del código donde se detectó el error.

```
if False ISNOTEQUAL True:
    ^
SyntaxError: invalid syntax
```

- Python admite la unión (concatenación) de cadenas mediante el operador +

```
# String concatenation

first = "Hello "
second = "World"

result = first + second

long_result = first + second + "!"
```

- Algunas tareas deben realizarse varias veces dentro de un programa. En lugar de reescribir el mismo código en múltiples lugares, se puede definir una función usando la palabra clave **def**

```
# define a function called
# my_function with parameter x
def my_function(x):
    return x + 1

# invoke our function
print(my_function(2))
# outputs: 3
print(my_function(3 + 5))
# outputs: 9
```

- La instrucción **if** de Python se utiliza para determinar la ejecución del código en función de la evaluación de una expresión booleana.

```
# else Statement

test_value = 50

if test_value < 1:
    print("Value is < 1")
else:
    print("Value is >= 1")

test_string = "VALID"

if test_string == "NOT_VALID":
    print("String equals NOT_VALID")
else:
    print("String equals something else!")
```

- Se puede utilizar un bloque **try / except** para manejar el error en un bloque de código.

```
def check_leap_year(year):  
    is_leap_year = False  
    if year%4 == 0:  
        is_leap_year = True  
  
try:  
    check_leap_year(2018)  
    print(is_leap_year) #The variable is_leap_year is declared  
inside the function.  
except:  
    print('Your code raised an error!')
```



- En Python, las listas son un tipo de datos versátil que puede contener múltiples tipos de datos diferentes dentro de los mismos corchetes.

```
numbers = [1, 2, 3, 4, 10]
names = ['Jenny', 'Alexis', 'Sam']
mixed = ['Jenny', 1, 2]
list_of_lists = [['a', 1], ['b', 2]]
```

- En Python, se puede usar un bucle **for** para realizar una acción un número específico de veces seguidas.

```
# Print the numbers 0, 1, 2:  
for i in range(3):  
    print(i)  
  
# Print "WARNING" 3 times:  
for i in range(3):  
    print("WARNING")
```

- En Python, un bucle **while** ejecutará repetidamente un bloque de código siempre que una condición se evalúe como Verdadero

```
# This loop will only run 1 time
hungry = True
while hungry:
    print("Time to eat!")
    hungry = False

# This loop will run 5 times
i = 1
while i < 6:
    print(i)
    i = i + 1
```

- W3schools. (2019). Python Tutorial. Retrieved 25 July, 2019, from <https://www.w3schools.com/python/>
- Codecademy. (2019). Learn Python 3. Retrieved 25 July, 2019, from <https://www.codecademy.com/learn/learn-python-3>
- Python.org. (2019). Beginner's Guide to Python. Retrieved 25 July, 2019, from <https://wiki.python.org/moin/BeginnersGuide>

# MUCHAS GRACIAS

Eliezer Figueroa  
MCT,MOS,MCSA,MCSE,ITIL,SCF

**UNIBE**

EDUCACIÓN  
CONTINUA

