Learn Python In 10 Days

Younes Derfoufi. Aggregated Doctor
Teacher of Computer Science and Mathematics
Research Professor
at CRMEF Oujda

11 avril 2024

Younes Derfoufi

www.tresfacile.net

LEARN PYTHON

IN 10 DAYS

Collection : Very Easy !

2024

Book intended for those
who want to learn the
Python language quickly

Contents

About the Author

Younes Derfoufi is a research professor at CRMEF OUJDA, an aggregated doctor with over 30 years of experience teaching mathematics and computer science to teacher students at CRMF Oujda. He has taught: about ten computer science modules at CRMEF Oujda: front-end and back-end web development including HTML, CSS, Bootstrap, Javascript, JQuery, PHP, MySql..., CMS-oriented web development: Joomla, Wordpress, Moodle, Simple Machine, PHPBB... Not forgetting web frameworks: Laravel, Django... He has also taught object-oriented programming in Java, Dart & Flutter... He has also taught Python programming: basics in Python, object-oriented programming, SQLite databases... Python GUIs with Tkinter & PyQt5... For around ten years, he taught information and communication technology: Multimedia ICT: Autoplay Media Studio, Video editing with AVS Video Editor & Camtasia Studio, image processing with Gimp... Mathematical ICT: Geogebra, algorithms with Algobox, LaTeX language, Scientific Workplace, Statistics with SPSS, Statistics with Python Numpy & Pandas... Before teaching all of these, the author worked as a mathematics teacher for trainee teachers for about twenty years, teaching in particular the following concepts: general algebra: groups, rings, fields, polynomials with one or more variables, linear algebra: vector spaces, matrices, bilinear form, quadratic forms, Euclidean and Hermitian spaces, alternating multilinear forms & determinants... numerical series, sequences & series of functions, power series analytic functions..., Riemann integration, generalized integral, parameter dependent, multiple integrals..., affine geometry, Euclidean vector geometry, Euclidean affine geometry, probability & statistics... Despite all this, the author had no difficulty teaching mathematics and computer science didactics, objective-based pedagogy, competency methodology, mathematics and computer science learning planning, didactic production, mathematical and computer science ICT. The author has also participated in numerous national and international conferences & seminars.

Copyright

Code Preview

The code employed in this book has been thoughtfully crafted utilizing valuable tools and specific LaTeX packages. Here is an example of code preview:

```
# define a string variable
s = "Hello World !"
print(s)
```

Who is This Book For?

The book **"Python in 10 Days"** is a practical guide designed specifically for beginners passionate about quickly and effectively learning Python. This book is aimed at a diverse audience, ranging from students eager for new skills to teachers looking to integrate programming into their courses, to students wanting to strengthen their computer knowledge.

Over ten days, we will guide you through the **fundamentals of Python**, from initial installations to advanced concepts such as **object-oriented programming** and **file manipulation.** Whether you are an **absolute beginner** or already have experience in programming, this book will walk you step by step through your journey to becoming a competent **Python developer. Each day**, you will explore a key aspect of the **language,** reinforce your knowledge with practical examples, and be ready to move to the next level by the end of each chapter. **Get ready** to dive into the fascinating world of **Python,** and by the end of these ten days, you will be equipped to **create applications, solve complex problems,** and harness the **full potential** of this versatile **programming language.**

Day 1: Installing the Tools :

Welcome to your first day of exploring Python! Before diving into code, let's start with the essentials: installing the necessary tools for Python development. This first step is crucial to ensure that your development environment is properly configured. Get ready to discover the necessary steps to install Python and associated tools on your system.

Day 2: Variables and Operators

Now that your environment is set up, let's get started! Today, you will dive into the world of variables and operators in Python. Understand how to store data in variables, manipulate this data with operators, and start building the foundations of your Python programs.

[...]

Continue exploring the various aspects of Python each day, learning gradually and applying your newly acquired knowledge. Have a great journey into the world of Python!

Introduction

Python is a high-level programming language known for its simplicity and readability. Created by Guido van Rossum and first released in 1991, Python has gained immense popularity due to its versatility and ease of use. It is widely used in various fields such as web development, data science, artificial intelligence, scientific computing, and more.

One of the key features of Python is its clear and concise syntax, which emphasizes readability and reduces the cost of program maintenance. This makes it an excellent choice for both beginners and experienced programmers alike.

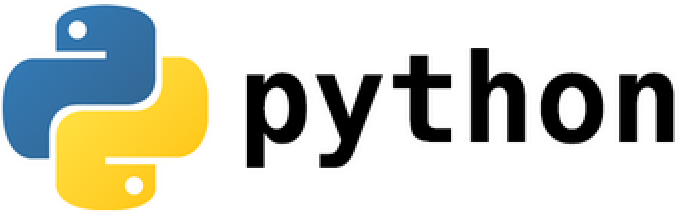
Python supports multiple programming paradigms, including procedural, object-oriented, and functional programming. It comes with a comprehensive standard library that provides support for a wide range of tasks, from file handling and networking to mathematical operations and web development.

Another notable aspect of Python is its strong community support and active development ecosystem. There are numerous third-party libraries and frameworks available, such as NumPy, Pandas, Django, Flask, TensorFlow, and PyTorch, which extend the capabilities of Python for various applications.

Overall, Python's simplicity, readability, and vast ecosystem make it a powerful and versatile language suitable for a wide range of programming tasks, from simple scripts to complex applications.

Chapter 1

Day 1: Installing the tools



Welcome to this first day of discovering Python! Before diving into the code, let's start with the essentials: installing the essential tools for developing in Python.

1.1 Python Installation: An All in One Solution

To configure **development tools in Python**, several options are available to you. However, we recommend the most **efficient** and

quick solution: using the **PySchool distribution**, a free and open source package. **PySchool** is equipped with a wide range of tools aimed at simplifying the **Python development** process, including:

PySchool Components

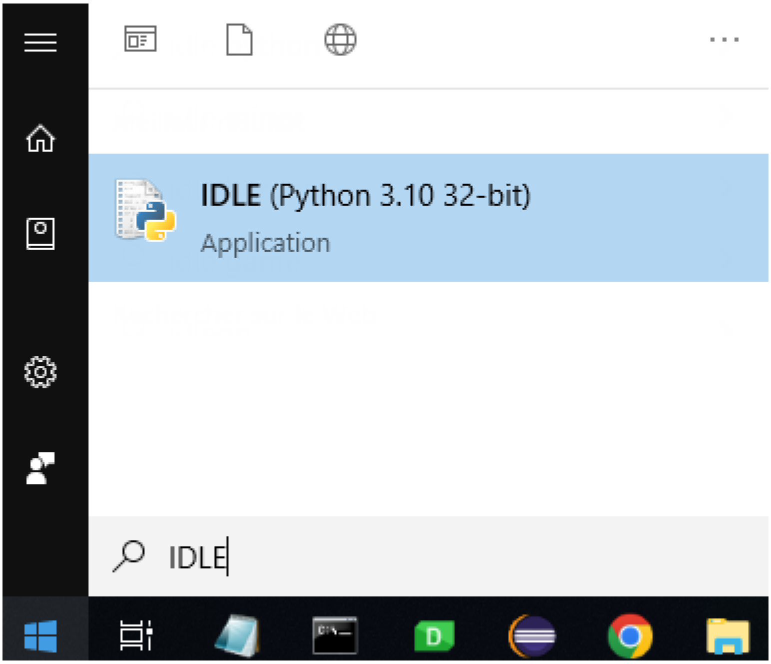
1. The latest version of the Python language
2. Integrated Ide: IDLE, Thonny Ide, VSCode, Jupyter Notbook...
3. The Qt Designer visual graphics tool
4. Python libraries necessary for machine learning and artificial intelligence.
5. As well as many other tools that you will discover!

PySchool package download link: https://sourceforge.net/projects/pyschool3/

1.2 First Python Program

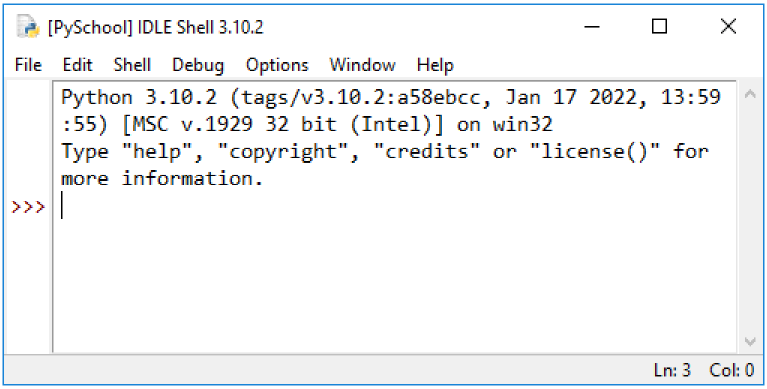
1.2.1 First Python Program Using the Integrated IDLE IDE

Python comes with an integrated development environment (IDE) called IDLE by default. To launch it, simply type IDLE in the search area of the start menu:



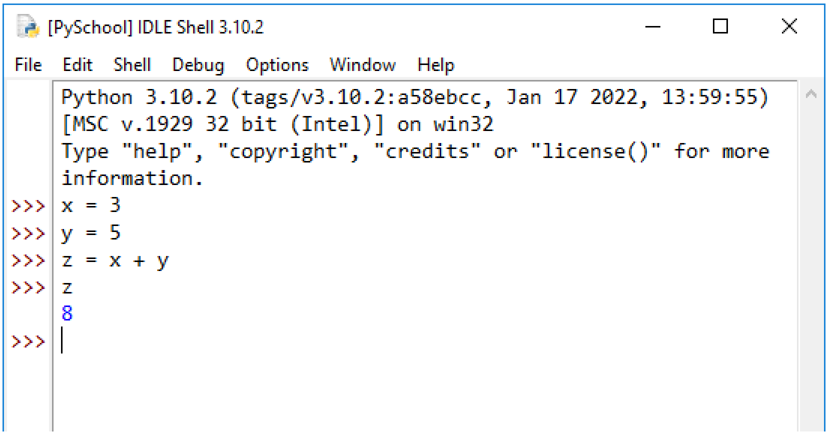
IDLE

Once launched, you will see the following view of the user interface of the Python-integrated IDE. It is a very minimalist IDE, but it is an excellent tool for beginners to get started and understand the workings of Python:


[PySchool] IDLE Shell 3.10.2
File Edit Shell Debug Options Window Help
Ln: 3 Col: 0

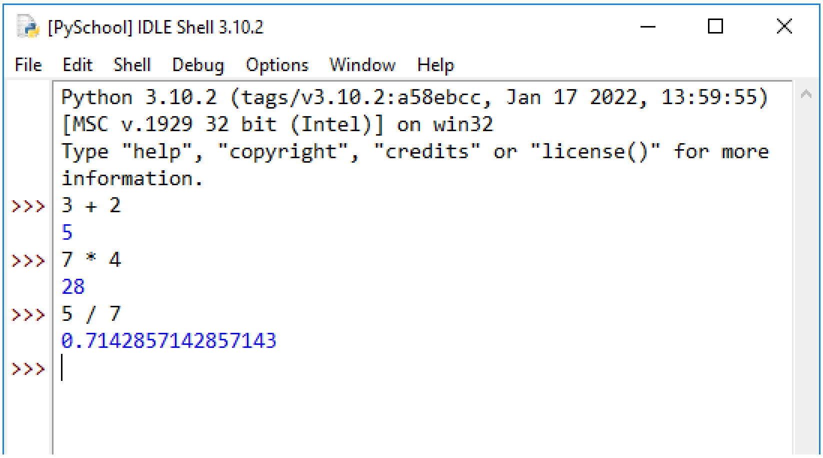
```
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 13:59
:55) [MSC v.1929 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for
more information.
>>> 
```

As you observe, the IDLE interface presents a clean and straightforward layout, with a text editor window where you can write and edit Python code. Below the text editor, there is a Python shell window where you can interactively execute Python code and see the results immediately. Let's now have fun **defining two integer variables, x** and **y**, and displaying their **sum, z = x + y:**


[PySchool] IDLE Shell 3.10.2
File Edit Shell Debug Options Window Help

```
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 13:59:55)
[MSC v.1929 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more
information.
>>> x = 3
>>> y = 5
>>> z = x + y
>>> z
8
>>> 
```

The **IDLE tool** can also be used as a **calculator.** You can utilize it to perform various **mathematical operations,** including **addition, subtraction, multiplication**, and **division**, among others. This capability makes it convenient for **quick calculations** and experimentation with numerical expressions:

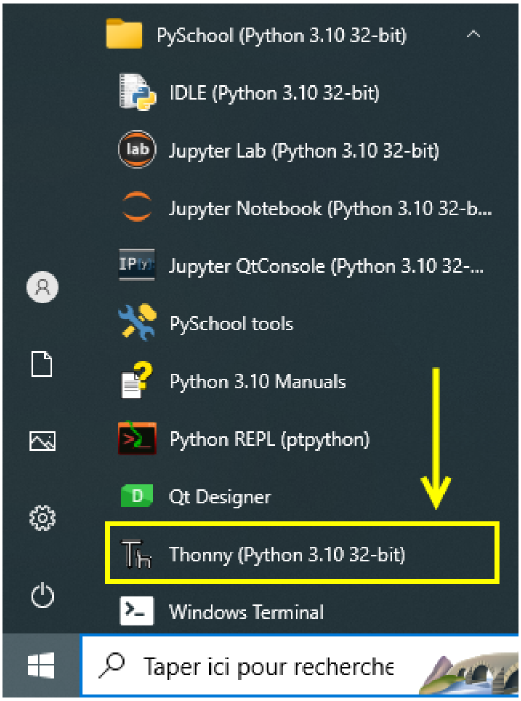

[PySchool] IDLE Shell 3.10.2
File Edit Shell Debug Options Window Help

```
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 13:59:55)
[MSC v.1929 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more
information.
>>> 3 + 2
5
>>> 7 * 4
28
>>> 5 / 7
0.7142857142857143
>>> 
```

1.2.2 First Python program using the Thonny IDE

As mentioned above, the **PySchool** distribution includes a number of **tools** by default, such as integrated development environments (IDEs) like **IDLE, Thonny IDE, VSCode, Jupyter Notebook**, among others. We can use the **Thonny IDE** to create our first Python program. Here are the steps to follow :

Step 1 : Launch Thonny IDE from the Start menu :


PySchool (Python 3.10 32-bit)
IDLE (Python 3.10 32-bit)
Jupyter Lab (Python 3.10 32-bit)
Jupyter Notebook (Python 3.10 32-b...
Jupyter QtConsole (Python 3.10 32-...
PySchool tools
Python 3.10 Manuals
Python REPL (ptpython)
Qt Designer
Thonny (Python 3.10 32-bit)
Windows Terminal
Taper ici pour recherche

Step 2 : Next, enter the following **code** and **save** the **file :**

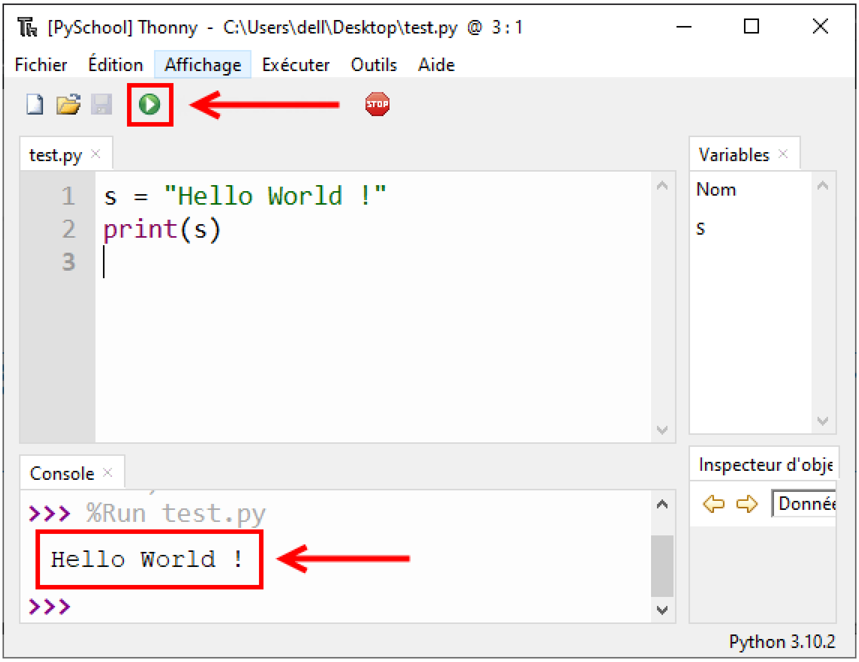
```
s = "Hello World !"
print(s)
```

Where :

1. We defined a **variable** named **s** with the value **'Hello World !'**
2. We used the **print()** statement to display the **value** of the **variable s.**

Step 3 : Running the **code**

Now, **click** on the **icon** shaped like a **small green triangle** located on the toolbar.



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

```
>>> %Run test.py
  Hello World !
>>>
```

You will then see the message **'Hello World !'** appear on the **console !** This is the **value** of the **variable s** displayed using the **print()** statement.

Chapitre 2

Day 2 : Variables and Operators

2.1 Variables in Python

Variables play a crucial role in **storing** the results of our calculations. In essence :

Python Variable

A **variable** boils down to a **name assigned** to a **value,** allowing **Python** to **memorize** it (not delete it) and **retrieve** it later by its **name.** One can conceptualize the **variable** as a kind of label affixed to our value, indicating how it is called. In **Python,** assigning a value to a variable is done using the = **operator.** On the left, **:**

```
myvariable = value
```

1. **myvariable :** here we specify the name of the variable, which must be a sequence of characters without spaces.
2. **value :** here we specify the value of the variable, which can take the form of any expression, as mentioned earlier.

3. **Unlike other programming languages :** Python does not require declaring a variable's type. A variable is created at the moment you assign it a value. The type will be automatically detected during assignment.

The main types of variables in Python are :

Type of Python variable

1. **Type integer** or **int** like as : **5 , 7 , 23 ...**
2. **Type flot** or floating point decimal type variable like **12.5 , 6.75 , 1.25 ...**
3. **Type string** like as **'Python' , 'Java'...**
4. **Boolean type** which takes only two values **True** or **False**

Exemple. variables of integer type

```
n = 11
print(n)
# output : 11
```

After execution this program **displays 11.** We can also display **explanatory text** together with the **variable n :**

Exemple. display the variable with text :

```
n = 11
print("The value of n is : " , n)
```

The program displays at runtime :
The value of n is : 11

Remarque. To display the **type** of a variable, we use the **type(variable_name)** instruction.

Exemple. displaying thevariable type

```
x = 13
y = 12.75
s = "Learn Python"
b = 10 < 3
print(type(x)) # display :<class 'int'>
print(type(y)) # display :<class 'float'>
print(type(s)) # display :<class 'str'>
print(type(b)) # display :<class 'bool'>
```

Remarque. A **Python variable** always has a **type**, even if it is undeclared. the **type** is defined when the **variable** is introduced and can be changed subsequently, which justifies the dynamism and power of the **Python** language.

Exemple. Type of variable.

```
x = 3 # x is of type int
print(x) # output : 3

# change the type of x to float
x = float(x) # x is now of type float
print(x) # output : 3.0
```

2.2 Comments in Python

2.2.1 What is a comment in Python ?