

## 1. INTRODUCTION

A programming language is a set of symbols, grammars and rules with the help of which one is able to translate algorithms to programs that will be executed by the computer. The programmer communicates with a machine using programming languages. Most of the programs have a highly structured set of rules.

The primary classifications of programming languages are: Machine Languages. Assembly Languages. High level Languages.

Python is developed by Guido van Rossum. Guido van Rossum started implementing Python in 1989 and released in 1991. Python is a very simple programming language so even if you are new to programming, you can learn python without facing any issues.

### 1.1 What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.

### 1.2 Features of Python

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-oriented way or a functional way.

## 2. Install Python on Ubuntu

Before installing Python 3, check whether you have it on your system. Open a terminal and run the following command:

```
python3
```

```
bosko@pnap:~$ python3
Python 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
bosko@pnap:~$
```

If the output shows the version, Python 3 is already installed on your Ubuntu machine. You can also see additional information about copyright, license, etc. To exit the prompt, press Ctrl+D.

If the command throws an error saying "bash: python3: command not found," Python is not installed.

There are several methods for installing Python on Ubuntu:

- Via APT. Install the latest version available in the default Ubuntu repository.
- From Source Code. Install the latest version from the official Python website.
- Via PPA. Install Python from the Deadsnakes PPA, a third-party repository designed for Ubuntu.

The sections below show the process for each installation method so you can choose the one that best fits your needs.

### 2.1 Method 1: Install Python via APT

This process uses the apt package manager and the default Ubuntu repository to install Python. This method is the easiest but may not always provide the latest

Python version. The latest versions are always available on the official website or in a third-party repository.

Follow the steps below:

### Step 1: Update the Package Repository

Update the package repository to ensure you get the latest available program version. Run the following command:

```
sudo apt update
```

### Step 2: Install Python

After the package repository information is updated, run the command below to install Python:

```
sudo apt install python3
```

```
bosko@nnap:~$ sudo apt install python3
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
 python3 ←
0 upgraded, 1 newly installed, 0 to remove and 36 not upgraded.
Need to get 22.8 kB of archives.
After this operation, 92.2 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 python3 amd64
 3.10.6-1~22.04 [22.8 kB]
Fetched 22.8 kB in 0s (61.2 kB/s)
Selecting previously unselected package python3.
(Reading database ... 224481 files and directories currently installed.)
Preparing to unpack ../python3_3.10.6-1~22.04_amd64.deb ...
Unpacking python3 (3.10.6-1~22.04) ...
Setting up python3 (3.10.6-1~22.04) ...
running python rtupdate hooks for python3.10...
running python post-rtupdate hooks for python3.10...
Processing triggers for man-db (2.10.2-1) ...
```

Wait until the installation finishes to start using Python.

### Step 3: Verify Installation

Verify whether the installation was successful by checking the program version:

```
python3 --version
```

## 2.2 Method 2: Install Python from Source Code

Use this method to download and compile the source code from the official developer. It is a bit more complicated, but the trade-off is accessing a newer Python release.

Follow the steps below:

### Step 1: Update Local Repositories

Update the local package repositories:

```
sudo apt update
```

### Step 2: Install Supporting Software

Compiling a package from source code requires additional software. Run the following command to install the required packages for Python:

```
sudo apt install build-essential zlib1g-dev libncurses5-dev libgd  
bm-dev libnss3-dev libssl-dev libreadline-dev libffi-dev wget
```

```

bosko@pnap:~$ sudo apt install build-essential zlib1g-dev libncurses5-dev libgdb
m-dev libnss3-dev libssl-dev libreadline-dev libffi-dev wget
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
build-essential is already the newest version (12.9ubuntu3).
wget is already the newest version (1.21.2-2ubuntu1).
wget set to manually installed.
The following packages were automatically installed and are no longer required:
  libpython3-stdlib python3-minimal python3.10 python3.10-minimal
Use 'sudo apt autoremove' to remove them.
Suggested packages:
  ncurses-doc readline-doc libssl-doc
The following NEW packages will be installed:
  libffi-dev libgdbm-dev libncurses-dev libncurses5-dev libnspr4-dev
  libnss3-dev libreadline-dev libssl-dev zlib1g-dev
0 upgraded, 9 newly installed, 0 to remove and 33 not upgraded.
Need to get 3,725 kB of archives.
    
```

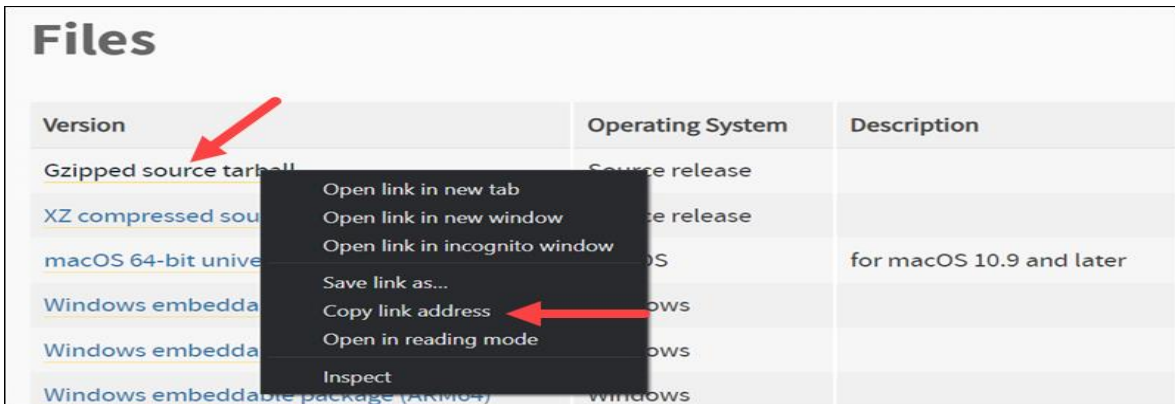
**Step 3:** Download the Latest Version of Python Source Code

1. Navigate to the /tmp directory with the cd command.

```
cd /tmp
```

The /tmp directory is often used for downloading source code because it is designated for temporary storage, allowing easy cleanup of downloaded files after installation.

2. Navigate to the official Python source code webpage and select the program version you want to install. Scroll down to the *Files* section of the page and copy the link to the Gzipped source tarball:



3. Use the [wget command](#) and the link above to download the newest release of Python Source Code:

```
wget https://www.python.org/ftp/python/3.12.1/Python-3.12.1.tgz
```

```
bosko@pnap:/tmp$ wget https://www.python.org/ftp/python/3.12.1/Python-3.12.1.tgz
--2023-12-13 09:56:16-- https://www.python.org/ftp/python/3.12.1/Python-3.12.1.tgz
Resolving www.python.org (www.python.org)... 146.75.120.223, 2a04:4e42:8e::223
Connecting to www.python.org (www.python.org)|146.75.120.223|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 27053241 (26M) [application/octet-stream]
Saving to: 'Python-3.12.1.tgz'

Python-3.12.1.tgz  100%[=====>] 25.80M  27.9MB/s   in 0.9s

2023-12-13 09:56:18 (27.9 MB/s) - 'Python-3.12.1.tgz' saved [27053241/27053241]
```

#### Step 4: Extract Compressed Files

In this step, extract the *tgz* file you downloaded. Run the command below:

```
tar -xf Python-3.12.1.tgz
```

**Replace the version numbers** in the *tgz* file name with the one you have downloaded.

#### Step 5: Test System and Optimize Python

Before you install the software, make sure you test the system and optimize Python.

The `./configure` command evaluates and prepares Python for the installation. Using the `--optimization` option speeds up code execution by 10-20%.

Switch to the directory you extracted the tgz file to, and enter:

```
./configure --enable-optimizations
```

```
bosko@pnap:/tmp$ cd Python-3.12.1
bosko@pnap:/tmp/Python-3.12.1$ ./configure --enable-optimizations
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for Python interpreter freezing... ./_bootstrap_python
checking for python3.12... no
checking for python3.12... no
checking for python3.11... no
checking for python3.10... python3.10
checking Python for regen version... Python 3.10.12
checking for pkg-config... no
checking for --enable-universalsdk... no
checking for --with-universal-archs... no
checking MACHDEP... "linux"
checking for gcc... gcc
checking whether the C compiler works... yes
```

The process takes some time to complete. Wait until it finishes before moving on to the next step.

### Step 6: Install Python

After the optimization and test are complete, run the command below to build the Python package:

```
sudo make install
```

**Note:** If you already have Python 3, but want to create a second Python installation on top of that one, run the following command:

```
sudo make altinstall
```

Allow the process to complete.

### Step 7: Verify Python Version

Check if the installation completed correctly by running:

```
python3 --version
```

The command should show the program version you installed.

**Note:** If you are starting with Python and are still looking for the right IDE or editor, see our comprehensive overview of the [best Python IDEs and code editors](#).

## 2.3 Method 3: Install Python via PPA

A Personal Package Archive (PPA) is a third-party repository in Ubuntu that offers a convenient way to access and install newer versions of programs that aren't included in the standard Ubuntu repositories.

Follow the steps below to install Python via a PPA:

### Step 1: Update and Refresh Repository Lists

Open a terminal window and run the command below:

```
sudo apt update
```

### Step 2: Install Supporting Software

The **software-properties-common** package gives better control over your package manager by letting you add PPA (Personal Package Archive) repositories. Install the supporting software with the command below:

```
sudo apt install software-properties-common
```



### Step 3: Add Deadsnakes PPA

Deadsnakes is a PPA with newer releases than the default Ubuntu repositories. Add the PPA by running the following:

```
sudo add-apt-repository ppa:deadsnakes/ppa
```

The system prompts you to press **Enter** to continue. Update the package lists again once the process is completed:

```
sudo apt update
```

### Step 4: Install Python 3

The Deadsnakes PPA has many Python versions in its database, allowing you to install older versions as well. Specify the version in the package name to install that version.

For example, run this command to install Python 3.12:

```
sudo apt install python3.12
```

```
bosko@nnap:~$ sudo apt install python3.12
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libpython3.12-minimal libpython3.12-stdlib python3.12-minimal
Suggested packages:
  python3.12-venv binutils binfmt-support
The following NEW packages will be installed:
  libpython3.12-minimal libpython3.12-stdlib python3.12 python3.12-minimal
0 upgraded, 4 newly installed, 0 to remove and 6 not upgraded.
Need to get 6,106 kB of archives.
After this operation, 23.2 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

Confirm the installation with **y** and allow the process to complete.

## Step 5: Verify Installation

Verify that Python is installed correctly by running:

```
python3 --version
```

The output should state the program version you chose to install.

## 3. Writing Your First Python Program: Hello World

One built-in function in Python is `print()`. The value or expression inside of the parenthesis of a `print()` function "prints" out to the REPL when the `print()` function is called.

An example using the `print()` function is below:

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

```
muthanna@maxo79:~$ python3  
Python 3.11.4 (main, Dec 7 2023, 15:43:41) [GCC 12.3.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> print("Hello World")  
Hello World  
>>> █
```

## 4. Python as a Calculator

Python can be used as a calculator to compute arithmetic operations like addition, subtraction, multiplication and division. Python can also be used for trigonometric calculations and statistical calculations.

### 4.1 Arithmetic

Python can be used as a calculator to make simple arithmetic calculations.

For Exaple

```
>>> 5 + 3
8
```

```
>>> 8 - 1.5
6.5
```

```
>>> 5000 / 2500
2.0
```

```
>>> 5 * 2
10
```

```
>>> 3 ** 2
9
```

```
>>> 2 * 5
10
>>> _ + 1.5 + 1
12.5
```

```
muthanna@maxo79:~$ python3
Python 3.11.4 (main, Dec 7 2023, 15:43:41) [GCC 12.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> 5+3
8
>>> 8-1.5
6.5
>>> 5000/2500
2.0
>>> 5*2
10
>>> 3**2
9
>>> 2*5
10
>>> _+1.5+1
12.5
>>>
```

## 4.2 Trigonometry: sine, cosine, and tangent

Trigonometry functions such as sine, cosine, and tangent can also be calculated using the Python REPL.

To use Python's trig functions, we need to introduce a new concept: importing modules.

In Python, there are many operations built into the language when the REPL starts. These include +, -, \*, / like we saw in the previous section. However, not all functions will work right away when Python starts. Say we want to find the sine of an angle. Try the following:

```
>>> sin(60)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'sin' is not defined
```

This error results because we have not told Python to include the sin function. The sin function is part of the *Python Standard Library*. The Python Standard Library comes with every Python installation and includes many functions, but not all of these functions are available to us when we start a new Python REPL session. To use Python's sin function, first import the sin function from the math *module* which is part of the Python Standard Library.

Importing modules and functions is easy. Use the following syntax:

```
from module import function
```

To import the sin() function from the math module try:

```
>>> from math import sin
>>> sin(60)
-0.3048106211022167
```

```
muthanna@maxo79:~$ python3
Python 3.11.4 (main, Dec 7 2023, 15:43:41) [GCC 12.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> sin(60)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'sin' is not defined. Did you mean: 'bin'?
>>> from math import sin
>>> sin(60)
-0.3048106211022167
>>> █
```

Success! Multiple modules can be imported at the same time. Say we want to use a bunch of different trig functions to solve the following problem.

An angle has a value of  $\pi/6$  radians. What is the sine, cos, and tangent of the angle? To solve this problem we need to import the `sin()`, `cos()`, and `tan()` functions. It is also useful to have the value of  $\pi$ , rather than having to write 3.14.... We can import all of these functions at the same time using the syntax:

```
from module import function1, function2, function3
```

**Note the commas in between the function names.**

Try:

```
>>> from math import sin, cos, tan, pi
>>> pi
3.141592653589793
>>> sin(pi/6)
0.49999999999999994
>>> cos(pi/6)
0.8660254037844387
>>> tan(pi/6)
0.5773502691896257
```

```

muthanna@maxo79:~$ python3
Python 3.11.4 (main, Dec 7 2023, 15:43:41) [GCC 12.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from math import sin, cos, tan, pi
>>> pi
3.141592653589793
>>> sin(pi/6)
0.49999999999999994
>>> cos(pi/6)
0.8660254037844387
>>> tan(pi/6)
0.5773502691896257
>>>

```

The following trig functions are part of Python's **math** module:

Trig function	Name	Description	Example	Result
<code>math.pi</code>	pi	mathematical constant $\pi$	<code>math.pi</code>	3.14
<code>math.sin()</code>	sine	sine of an angle in radians	<code>math.sin(4)</code>	9.025
<code>math.cos()</code>	cosine	cosine of an angle in radians	<code>cos(3.1)</code>	400
<code>math.tan()</code>	tangent	tangent of an angle in radians	<code>tan(100)</code>	2.0
<code>math.asin()</code>	arc sine	inverse sine, ouput in radians	<code>math.sin(4)</code>	9.025
<code>math.acos()</code>	arc cosine	inverse cosine, ouput in radians	<code>log(3.1)</code>	400
<code>math.atan()</code>	arc tangent	inverse tangent, ouput in radians	<code>atan(100)</code>	2.0
<code>math.radians()</code>	radians conversion	degrees to radians	<code>math.radians(90)</code>	1.57
<code>math.degrees()</code>	degree conversion	radians to degrees	<code>math.degrees(2)</code>	114.59

### 4.3 Exponents and Logarithms

Calculating exponents and logarithms with Python is easy. Note the exponent and logarithm functions are imported from the **math** module just like the trig functions were imported from the **math** module above.

The following exponents and logarithms functions can be imported from Python's math module:

- `log`
- `log10`
- `exp`
- `e`
- `pow(x, y)`
- `sqrt`

Let's try a couple of examples:

```
>>> from math import log, log10, exp, e, pow, sqrt
>>> log(3.0*e**3.4)          # note: natural log
4.4986122886681095
```

A right triangle has side lengths 3 and 4. What is the length of the hypotenuse?

```
>>> sqrt(3**2 + 4**2)
5.0
```

The power function `pow()` works like the `**` operator. `pow()` raises a number to a power.

```
>>> 5**2
25
```

```
>>> pow(5,2)
25.0
```

```

muthanna@maxo79:~$ python3
Python 3.11.4 (main, Dec 7 2023, 15:43:41) [GCC 12.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from math import log, log10, exp, e, pow, sqrt
>>> log(3.0*e**3.4)
4.4986122886681095
>>> sqrt(3**2+4**2)
5.0
>>> 5**2
25
>>> pow(5,2)
25.0
>>> █

```

The following exponent and logarithm functions are part of Python's **math** module:

Math function	Name	Description	Example	Result
<code>math.e</code>	Euler's number	mathematical constant $e$	<code>math.e</code>	2.718
<code>math.exp()</code>	exponent	$e$ raised to a power	<code>math.exp(2.2)</code>	9.025
<code>math.log()</code>	natural logarithm	log base $e$	<code>math.log(3.1)</code>	400
<code>math.log10()</code>	base 10 logarithm	log base 10	<code>math.log10(100)</code>	2.0
<code>math.pow()</code>	power	raises a number to a power	<code>math.pow(2,3)</code>	8.0
<code>math.sqrt()</code>	square root	square root of a number	<code>math.sqrt(16)</code>	4.0

## 4.4 Statistics

To round out this section, we will look at a couple of statistics functions. These functions are part of the Python Standard Library, but not part of the **math** module. To access Python's statistics functions, we need to import them from



the **statistics** module using the statement `from statistics import mean, median, mode, stdev`. Then the functions `mean`, `median`, `mode` and `stdev` (standard deviation) can be used.

```
>>> from statistics import mean, median, mode, stdev

>>> test_scores = [60, 83, 83, 91, 100]

>>> mean(test_scores)
83.4

>>> median(test_scores)
83

>>> mode(test_scores)
83

>>> stdev(test_scores)
14.842506526863986
```

```
muthanna@maxo79:~$ python3
Python 3.11.4 (main, Dec 7 2023, 15:43:41) [GCC 12.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from statistics import mean, median, mode, stdev
>>> test_scores=[60,83,83,91,100]
>>> mean(test_scores)
83.4
>>> median(test_scores)
83
>>> mode(test_scores)
83
>>> stdev(test_scores)
14.842506526863986
>>>
```

Alternatively, we can import the entire **statistics** module using the statement *import statistics*. Then to use the functions, we need to use the names *statistics.mean*, *statistics.median*, *statistics.mode*, and *statistics.stdev*. See below:

```

muthanna@maxo79:~$ python3
Python 3.11.4 (main, Dec 7 2023, 15:43:41) [GCC 12.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import statistics
>>> test_scores=[60,83,83,91,100]
>>> statistics.mean(test_scores)
83.4
>>> statistics.median(test_scores)
83
>>> statistics.mode(test_scores)
83
>>> statistics.stdev(test_scores)
14.842506526863986
>>>

```

The following functions are part of Python's **statistics** module. These functions need to be imported from the `statistics` module before they are used.

Statistics function	Name	Description	Example	Result
<code>mean()</code>	mean	mean or average	<code>mean([1, 4, 5, 5])</code>	3.75
<code>median()</code>	median	middle value	<code>median([1, 4, 5, 5])</code>	4.5
<code>mode()</code>	mode	most often	<code>mode([1, 4, 5, 5])</code>	5
<code>stdev()</code>	standard deviation	spread of data	<code>stdev([1, 4, 5, 5])</code>	1.892
<code>variance()</code>	variance	variance of data	<code>variance([1, 4, 5, 5])</code>	3.583