

Twitter Thread by Pratham Prason



Pratham Prason

@PrasonPratham



This is a step-by-step guide on getting started with Python as a total beginner.

(includes free resources and everything else you need to get started)



Before we begin, I want to congratulate you on your decision to learn how to code using Python.

I still remember how I wrote my first piece of code 6 years and all the amazing and cool things I've been able to do with it ever since.

(2 / 24)

Make sure to keep in mind that it is probably best for you to keep your expectations in check.

Don't expect to make AAA games or state of the art machine learning models in a week.

(3 / 24)

Programming is not something that you can learn in a single week, it takes consistent effort and dedication over time to get good at it.

With all that being said, let's dive straight in.

(4 / 24)

In order to write Python code, you'll need to install Python on your system.

Linux and macOS users can skip this step because they come pre-installed with Python.

Download link: <https://t.co/KSZ4Qd6CNk>

(5 / 24)

The screenshot shows the Python.org website's download page. At the top, there is a navigation bar with links for Python, PSF, Docs, PyPI, Jobs, and Community. Below this is the Python logo and a search bar. A main heading reads "Download the latest source release" with a button for "Download Python 3.9.1". There are also links for other operating systems and pre-releases. Below this is a table of active Python releases.

Python version	Maintenance status	First released	End of support	Release schedule
3.9	bugfix	2020-10-05	2025-10	PEP 596
bugfix		2019-10-14	2024-10	PEP 569

Click on the .exe file and follow the instructions.

Make sure to Add python to path by checking this option ■

(6 / 24)

The screenshot shows the "Python 3.7.2 (32-bit) Setup" window. It features the Python logo and the text "python for windows". The main heading is "Install Python 3.7.2 (32-bit)". Below this, it says "Select Install Now to install Python with default settings, or choose Customize to enable or disable features." There are two main options: "Install Now" with the path "C:\Users\Ron\AppData\Local\Programs\Python\Python37-32" and "Customize installation" with the subtext "Choose location and features". At the bottom, there are two checked checkboxes: "Install launcher for all users (recommended)" and "Add Python 3.7 to PATH", which is highlighted with a red box. A "Cancel" button is located in the bottom right corner.

Now you need to install a place where you can write your Python code, just like how you write your essays in Word or Google docs.

We'll be installing VS-Code, one of the best code editors out there and it's free!

(7 / 24)

Fortunately, Microsoft has this wonderful guide that'll help you out.

■<https://t.co/0cN2JyM2di>

(8 / 24)

In case you are unable to set up Python on your own system then I recommend using repl, a great way to write and run Python code without any hassle.

■<https://t.co/eijcOcz42c>

(8 / 24)

Here are the topics you should be focusing

- Printing statements
- Variables
- Operators
- Conditions
- Functions
- Loops

Let's take a closer look.

(9 / 24)

Printing statements

You can 'print' or output in Python using the print() function.

print('Hello World') will give you an output of `Hello World`

(10 / 24)

Variables

These store certain values that can change.

For example, You can declare a variable 'x' with a certain value.

x = 9 [Here 9 is assigned to variable x]

y = 'Hello World' ['Hello World' is assigned to y]

x = 8 [x has been updated to a new value which is 8]

(11 / 24)

Operators

These carry out arithmetic operations in Python

+ : Addition

- : Subtraction

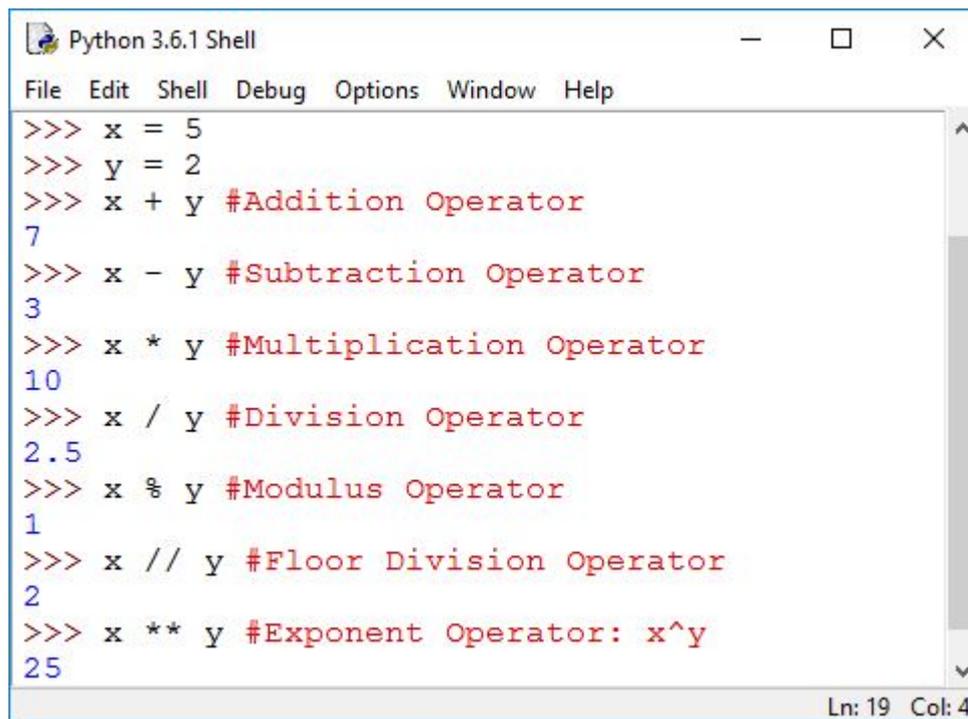
* : Multiplication

/ : Divide

% : Modulus (Remainder after dividing)

** : Exponents

(12 / 24)

A screenshot of a Python 3.6.1 Shell window. The window title is "Python 3.6.1 Shell" and it has standard window controls (minimize, maximize, close). The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main area shows a series of Python commands and their outputs:

```
>>> x = 5
>>> y = 2
>>> x + y #Addition Operator
7
>>> x - y #Subtraction Operator
3
>>> x * y #Multiplication Operator
10
>>> x / y #Division Operator
2.5
>>> x % y #Modulus Operator
1
>>> x // y #Floor Division Operator
2
>>> x ** y #Exponent Operator: x^y
25
```

The status bar at the bottom right shows "Ln: 19 Col: 4".

Conditions

They look like this ■

```
a = 33
```

```
b = 200
```

```
if b > a:
```

```
    print("b is greater than a")
```

```
else:
```

```
    pass
```

Think of it as telling Python to do something based on conditions: if this is true, do this; else do something different.

(13 / 24)

Functions

These are basically blocks of code that can be run when you call them. This helps us write code more efficiently.

In Python, you can make functions using the def keyword.

```
def myFunction():  
    print("Hello")  
myFunction()
```

(14 / 24)

Loops

These essentially run a piece of code over and over again until a certain condition is met.

There are 2 types of loops in Python:

- While loops
- For loops

Let's take a look at them.

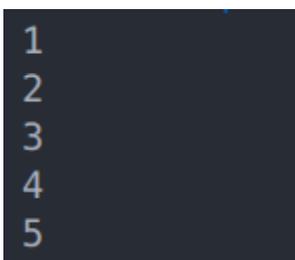
(15 / 24)

Here's what While Loops look like.

```
i = 1  
while i < 6: //While i is less than 6  
    print(i) // do  
    i += 1 //this
```

The output is :

(16 / 24)



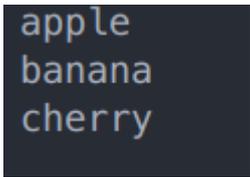
```
1  
2  
3  
4  
5
```

Here's a For loop

```
fruits = ["apple", "banana", "cherry"]  
for fruit in fruits:  
    print(x)
```

The output:

(17 / 24)



```
apple  
banana  
cherry
```

The for loops can be a bit tricky to understand, let me try to break it down for you.

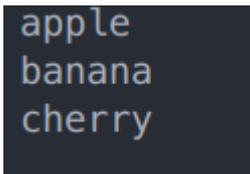
'fruit' in the above loop is a temporary variable. It gets a value from the list called 'fruits' starting from "apple" then to "banana" and then "cherry".

(18 / 24)

Each time it gets a value, it is printed.

This explains why we get this output ■

(19 / 24)



```
apple  
banana  
cherry
```

All that I explained before was to give you a taste of the basics of Python, having strong fundamentals at this stage is very important.

I have even more threads for further steps in Python coming up, stay tuned.

(20 / 24)

These are some of the resources I would recommend to you for further learning ■

Incredible text-based tutorials: <https://t.co/qRvZNIiSM2>

(21 / 24)



Learn Python Programming

TUTORIALS

EXAMPLES

REFERENCES

ONLINE COMPILER

Python is a powerful general-purpose programming language. It is used in web development, data science, creating software prototypes, and so on. Fortunately for beginners, Python has simple easy-to-use syntax. This makes Python an excellent language to learn to program for beginners.

Our Python tutorial will guide you to learn Python one step at a time.

If you prefer videos, watch our [Python for beginners](#) playlist on Youtube.

Page Index

Introduction

Flow Control

Function

Data Types

File Handling

Object & Class

Introduction

Getting Started

Keywords and Identifiers

Statements & Comments

Python Variables

Python Datatypes

Python Type Conversion

Object Oriented Programming (advanced) : <https://t.co/Bpxrjlv4MQ>

This course on Traversy Media's channel is a great next step for taking your python skills to the next level!

(22 / 24)

`class User:
def __init__(self):
def __init__(self):
class
def __init__(self, name, membership_type):
self.name = name
self.membership_type = membership_type`

Object Oriented Python

With Caleb Curry

This is probably the most underrated resource for learning python.

(23 / 24)

Python » English » 3.9.1 » Documentation » | [modules](#) | [index](#)

Download
Download these documents

Docs by version
Python 3.10 (in development)
Python 3.9 (stable)
Python 3.8 (stable)
Python 3.7 (security-fixes)
Python 3.6 (security-fixes)
Python 3.5 (EOL)
Python 2.7 (EOL)
All versions

Other resources
PEP Index
Beginner's Guide
Book List
Audio/Visual Talks
Python Developer's Guide

Python 3.9.1 documentation

Welcome! This is the documentation for Python 3.9.1.

Parts of the documentation:

- [What's new in Python 3.9?](#)
or all "What's new" documents since 2.0
- [Installing Python Modules](#)
installing from the Python Package Index & other sources
- [Tutorial](#)
start here
- [Distributing Python Modules](#)
publishing modules for installation by others
- [Library Reference](#)
keep this under your pillow
- [Extending and Embedding](#)
tutorial for C/C++ programmers
- [Language Reference](#)
describes syntax and language elements
- [Python/C API](#)
reference for C/C++ programmers
- [Python Setup and Usage](#)
how to use Python on different platforms
- [FAQs](#)
frequently asked questions (with answers!)
- [Python HOWTOs](#)
in-depth documents on specific topics

Indices and tables:

- [Global Module Index](#)
quick access to all modules
- [Search page](#)
search this documentation
- [General Index](#)
all functions, classes, terms
- [Complete Table of Contents](#)
lists all sections and subsections
- [Glossary](#)