# python

## Programming

# Introduction to Python

- What is Python?
- History of Python?
- Why is it Popular? - Python packages

# What is Python?

Python is a widely-used, interpreted, object-oriented, and high-level programming language with dynamic semantics, used for general-purpose programming

# History of Python

## 1991

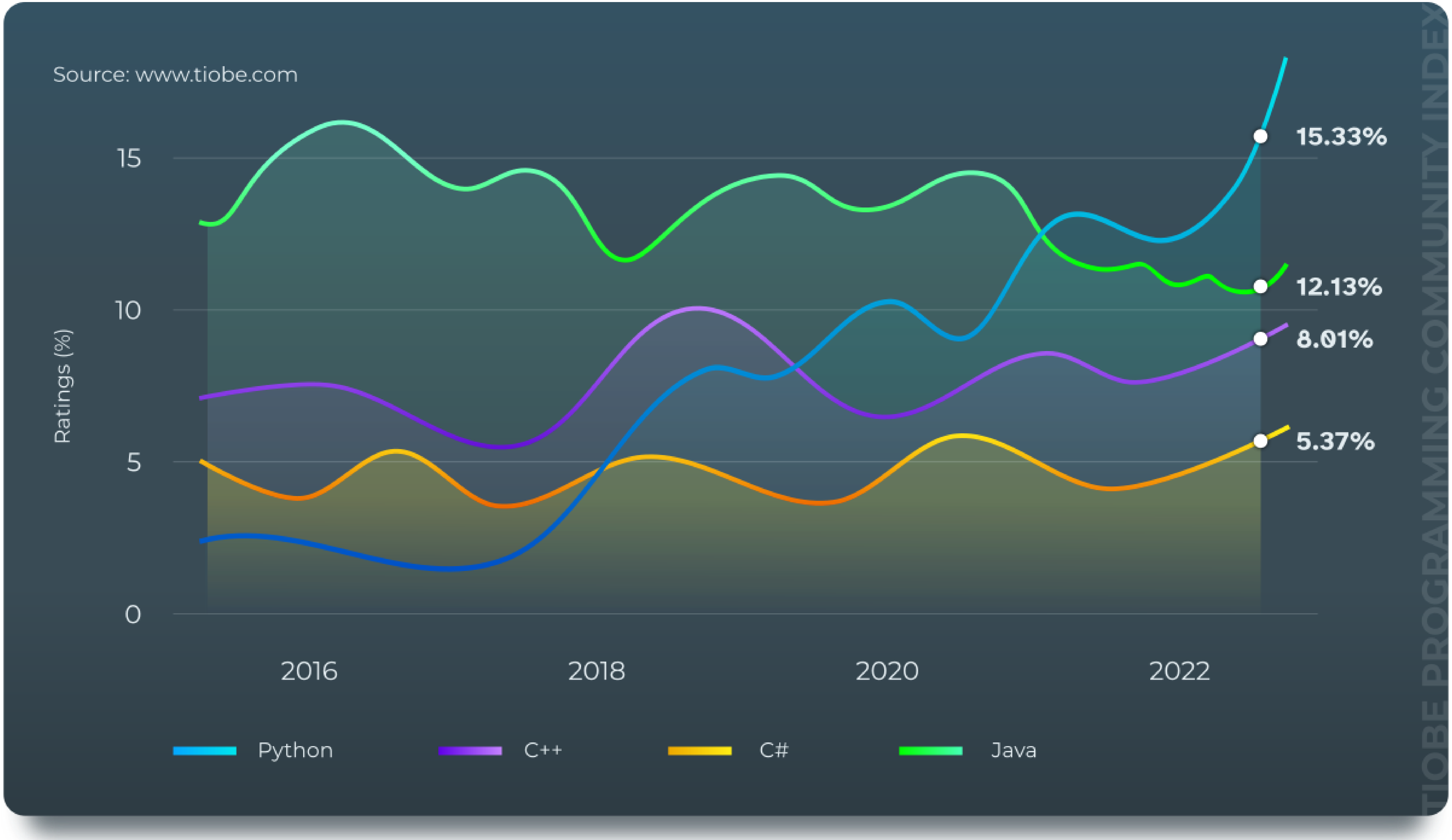
Python was created by "Guido van Rossum" in 1991 as an alternative scripting language for Linux

## 2005

Around 2005 it became a programming language

## 2008

In 2008, version 3 was released (which does not support backwards)
In 2019 they stopped supporting version 2



© Tamar Kossover

# Why is it Popular?

Some of the reasons why Python is very popular:

- It's an open source,
- It's simplicity in reading and writing,
- It supports different platforms, and integrates with different languages,
- Interpreted Language,
- Object oriented and functional
- And mainly because of the huge collection of **packages** it includes

# Python installation

**python.org**

After installation we will get an interpreter window

It can be used in 2 ways:

1. Interactive - command and result
2. Running a .py file by the command:

```
python file.py
```

# iPython

**iPython  Package** (Interactive Python)
is a command shell for interactive computing, that
offers shell syntax, tab completion, history and more.

installation command: pip install ipython

# Anaconda

**A set of IDEs (integrated development environment)**

Free, contains good IDE tools:

Contains **Conda** - Anaconda's package manager

which installs lots of libraries and keeps the libraries compatible with

Python versions.

ANACONDA®

# PyCharm

It provides a user-friendly interface for coding, debugging and testing, working with db, client, etc.

Good option for more complex projects.

# SPYDER

Integrated environment -

interpreter + scripts/files at the same time

Good option for tests.

# JUPITER NOTEBOOK

Browser-based environment, Interactive, saved as a notebook.
Can be shared with others (also through the cloud),
allow adding comments, HTML code and more.
Saved with the .ipynb extension

# Python Variables and Data Types

# Python Syntax

- Python uses indentation to indicate a block of code (and not in {})

  (Best practice - use tabs (4 spaces), but one space is enough)

  We have to use the same number of spaces in the same code block.

- To use the packages - we will import them by:

  Import math

  math.floor(5.8888)

- we can add comments by #

# Python Variables

- Variables in python are dynamically types - they are created the moment you first assigning a value to it.
- The type is not declared in python, it can be changed later (bad practice)
- type() function - returns the data type of a variable
- Variable names are case-sensitive

- input & output:

```
name = input("Please enter your name:")
print("Hello", name)
```

- Exchange between variables:

```
var1 = 1
var2 = 2
var1, var2 = var2, var1
```

- Delete variables: by the "del" keyword

```
del var1
```

- Global variable

Global variables can be used by everyone, both inside functions and out.

```
x = "awesome"
def myfunc():
    print("Python is " + x)
myfunc()
```

To create or use a global variable inside a function, we can use the global keyword.

```
def myfunc():
    global x
    x = "fantastic"
```

# Data Types

- Boolean: True , False
- Numbers: 3.1415, 1234, 999L, 3+4j
- Strings: 'spam', "guido's"
- Lists: [1, [2, 'three'], 4]
- Dictionaries: {'food':'spam', 'taste':'yum'}
- Tuples: (1, 'spam', 4 ,'U')

## Null

optional_data = **None**

## Boolean

returns: True / False

operators:

== , != , < , <= , > , >=

# Numbers

**Int, float and complex**

operators:

+, -,/, //, %, +=, -=, *,**

functions:

int(x) / float(x)

abs(x)

pow(x**y)

sqrt(x)