Python Intro

Let's get started with Python!

1.

Intro

What is Python and what is it about?

Who?

Guido van Rossum created Python, first released on February 20, 1991.

While you may know the python as a giant snake, the name of the Python programming language comes from an old BBC television comedy sketch series called **Monty Python's Flying Circus**.

Who?

One of Python's fantastic features is that it is one person’s work.

Usually, new programming languages are developed and published by large companies employing many professionals. Due to copyright rules, it is tough to name any of the people involved in the project.

`Python is an exception.`

Who?

Of course, Guido van Rossum did not develop and **`evolve`** all the Python components himself.

Python is maintained by the Python Software Foundation, a non-profit membership organization and a community devoted to developing, improving, expanding, and popularizing the Python language and its environment.

What?

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

It’s everywhere, and people use numerous Python-powered devices daily, whether they realize it or not.

Question: Have you been in touch with python before?

Why?

There are also a couple of factors that make Python great for learning:

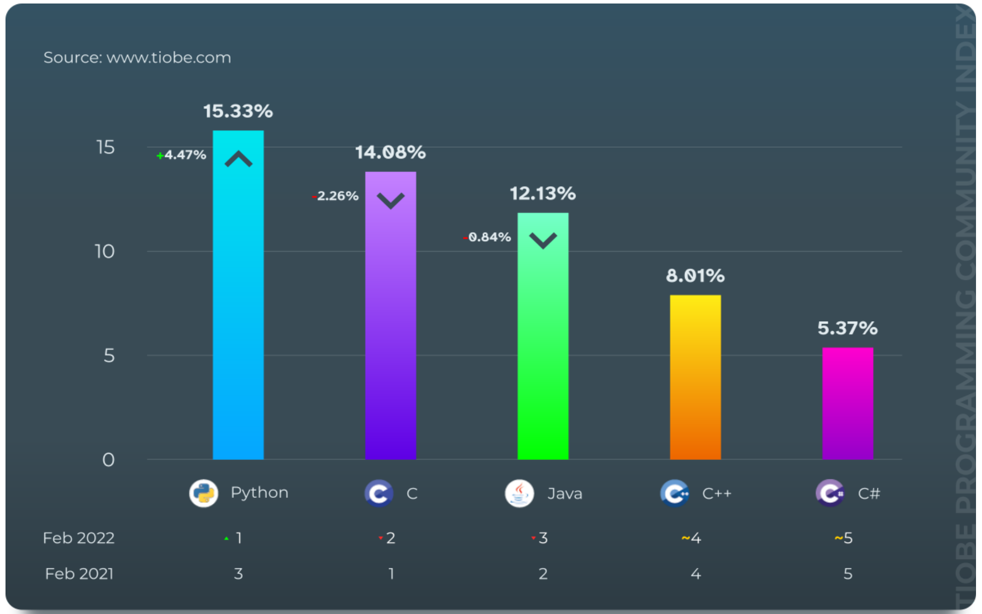
- **`It is easy to learn`** – the time needed to learn Python is shorter than for many other languages
- **`It is easy to use`** for writing new software – it’s often possible to write code faster when using Python
- **`It is easy to obtain, install and deploy`** – Python is free, open and multiplatform

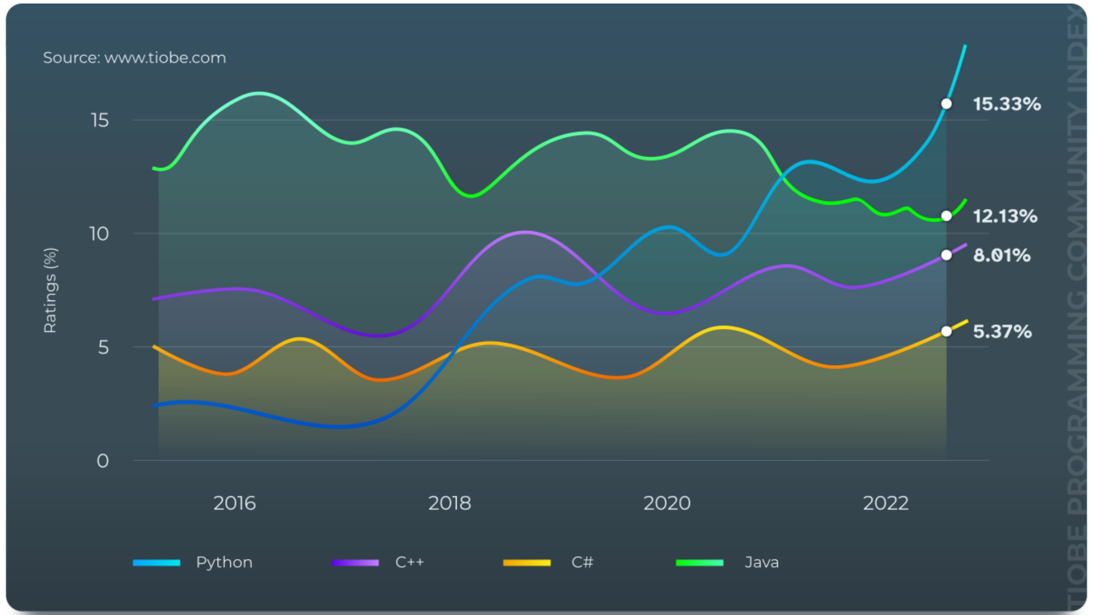
Python goals

In 1999, Guido van Rossum defined his goals for Python:

- an easy and intuitive language just as powerful as those of the major competitors
- open source so that anyone can contribute to its development
- code that is as understandable as plain English
- suitable for everyday tasks, allowing for short development times.

Some statistics





Python isn’t a young language.

It is mature and trustworthy.

It’s not a one-hit-wonder.

It’s a bright star in the programming firmament, and time spent learning Python is an excellent investment.

2.

History

Some History about Python

Python 1.0

Python reached version 1.0 in January 1994. The significant new features included in this release were the functional programming tools `lambda`, `map`, `filter` and `reduce`.

Van Rossum stated that "Python acquired lambda, reduce(), filter() and map(), courtesy of a Lisp hacker who missed them and submitted working patches".

Python 2.0

Python 2.0, released in October 2000, introduced **`list comprehensions`**, a feature borrowed from the functional programming languages SETL and Haskell.

Python's syntax for this construct is very similar to Haskell's, apart from Haskell's preference for punctuation characters and Python's **`preference for alphabetic keywords`**.

Python 2.0 also introduced a **`garbage collector`** capable of collecting reference cycles.

Python 2.7

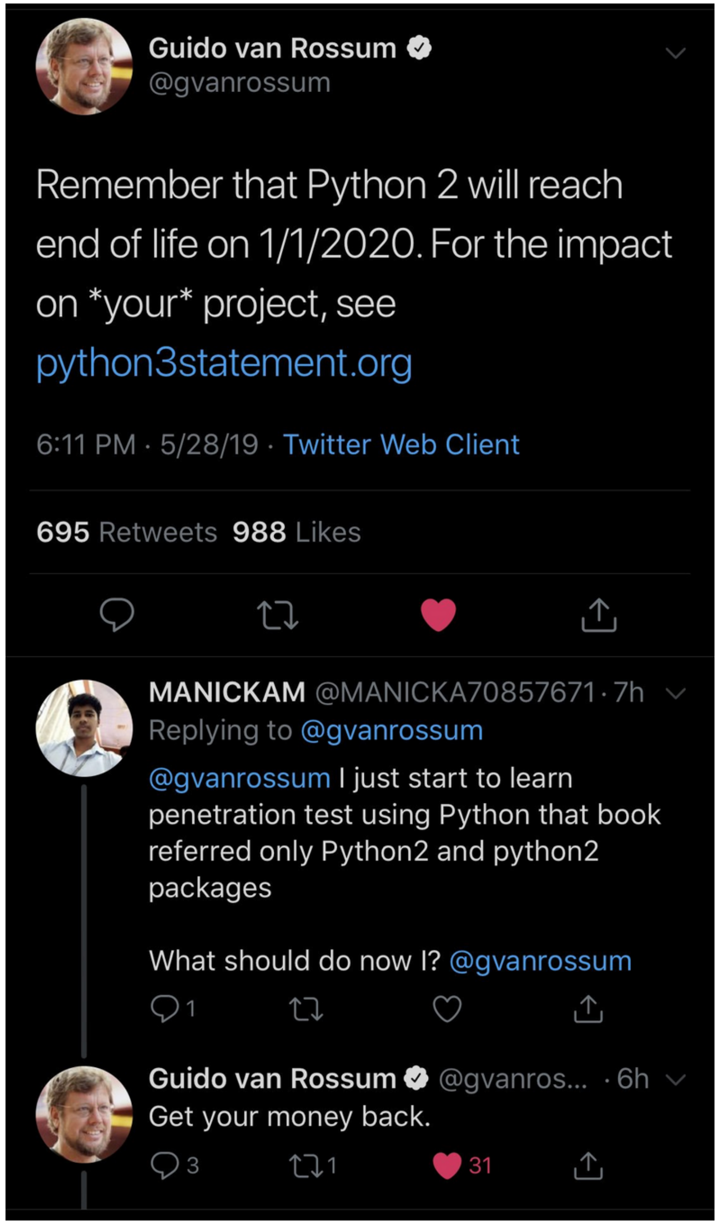
In November 2014, it was announced that Python 2.7 would be supported until 2020, but users were encouraged to move to Python 3 as soon as possible.

Python 2.7 support ended on January 1, 2020, along with code freeze of 2.7 development branch.

A final release, 2.7.18, occurred on April 20, 2020, and included fixes for critical bugs and release blockers.

`This marked the end-of-life of Python 2.`

Python 2.7



Guido van Rossum
@gvanrossum

Remember that Python 2 will reach end of life on 1/1/2020. For the impact on *your* project, see python3statement.org

6:11 PM · 5/28/19 · Twitter Web Client

695 Retweets 988 Likes

MANICKAM @MANICKA70857671 · 7h
Replying to @gvanrossum

@gvanrossum I just start to learn penetration test using Python that book referred only Python2 and python2 packages

What should do now I? @gvanrossum

1

Guido van Rossum @gvanros... · 6h
Get your money back.

3 1 31

Python 2.7

Comparison Parameter	**Python 2**	**Python 3**
“Print” Keyword	In Python 2, print is considered to be a statement and not a function.	In Python 3, print is considered to be a function and not a statement.
Division of Integers	On the division of two integers, we get an integral value in Python 2. **For instance, 7/2 yields 3 in Python 2.**	On the division of two integers, we get a floating-point value in Python 3. **For instance, 7/2 yields 3.5 in Python 3.**
Backward compatibility	Python 2 codes can be ported to Python 3 with a lot of effort.	Python 3 is not backward compatible with Python 2.

Python 3.x

Python 3.0 (also called "Python 3000" or "Py3K") was released on December 3, 2008.

It was designed to rectify fundamental design flaws in the language – the changes required could not be implemented while retaining full backwards compatibility with the 2.x series, which necessitated a new major version number.

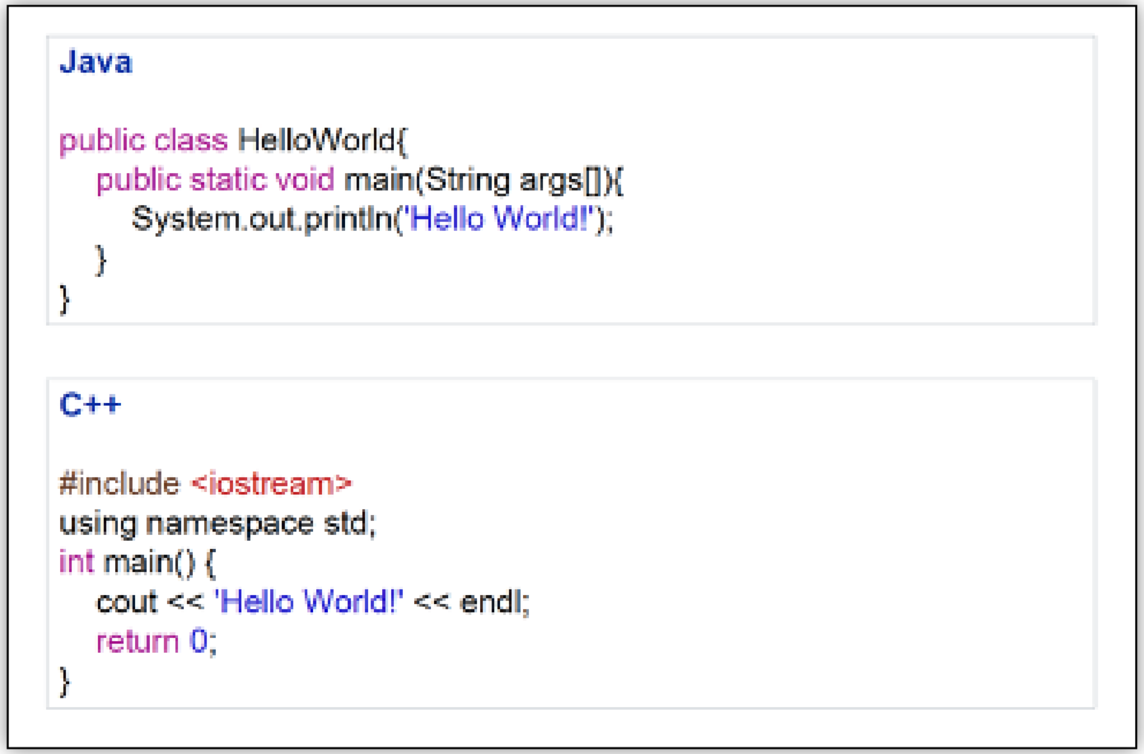
The guiding principle of Python 3 was: "reduce feature duplication by removing old ways of doing things".

3.

Syntax

Let us see some examples

The proof why Python is loved



Java

```
public class HelloWorld{
    public static void main(String args[]){
        System.out.println('Hello World!');
    }
}
```

C++

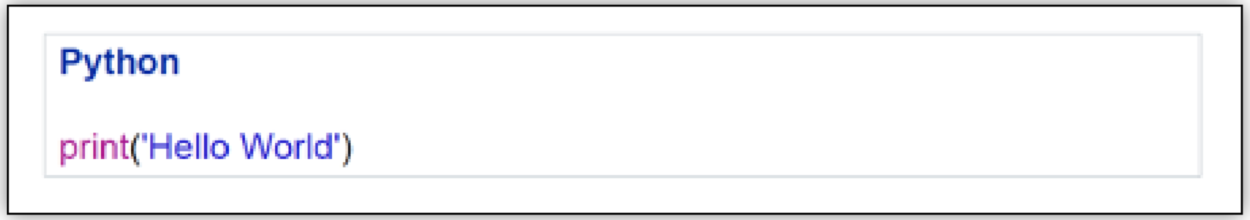
```
#include <iostream>
using namespace std;
int main() {
    cout << 'Hello World!' << endl;
    return 0;
}
```

The proof why Python is loved

And now the same in Python

The proof why Python is loved

And now the same in Python



Python

```
print('Hello World')
```

Why Python?

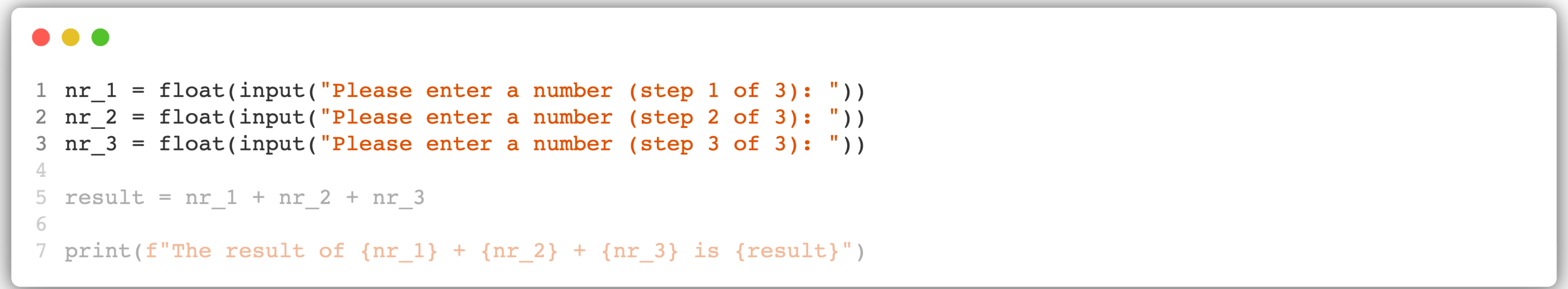
- Simple Syntax
- High-Level programming language
- Cross-Platform
- Interpreted
- Object-Oriented
- Many libraries available

Python Interpreter

- Makes sure that code is written correctly (syntax check)
- Then compiles Python code and executes the code as well
- Different Python interpreter versions available
  - Python 3.11 interpreter can compile Python 3.8 code
  - Python 3.8 interpreter may not be able to translate Python 3.11 code
- Python 3.11 offers new features not known by the 3.10 or 3.8 interpreter

A mysterious first program

Let us see our first code example:

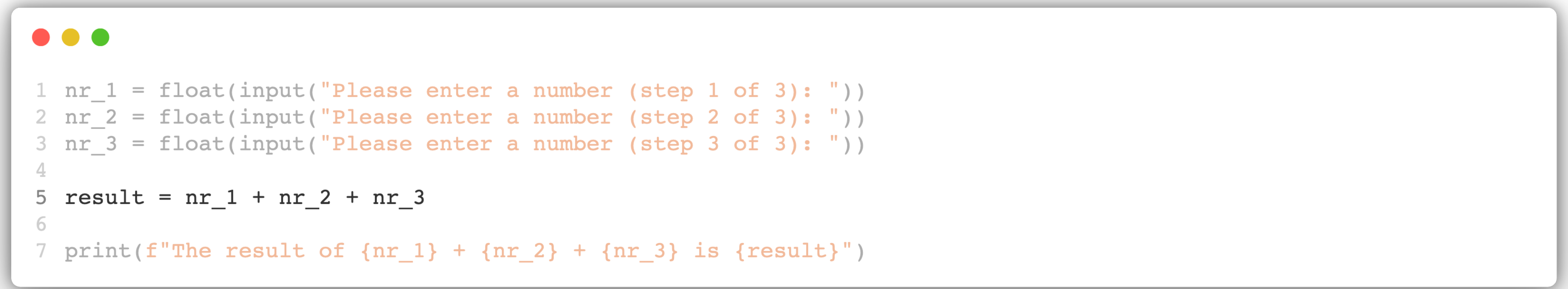


```python
nr_1 = float(input("Please enter a number (step 1 of 3): "))
nr_2 = float(input("Please enter a number (step 2 of 3): "))
nr_3 = float(input("Please enter a number (step 3 of 3): "))

result = nr_1 + nr_2 + nr_3

print(f"The result of {nr_1} + {nr_2} + {nr_3} is {result}")
```

Can you spot what the program will do?

A mysterious first program

Let us see our first code example:

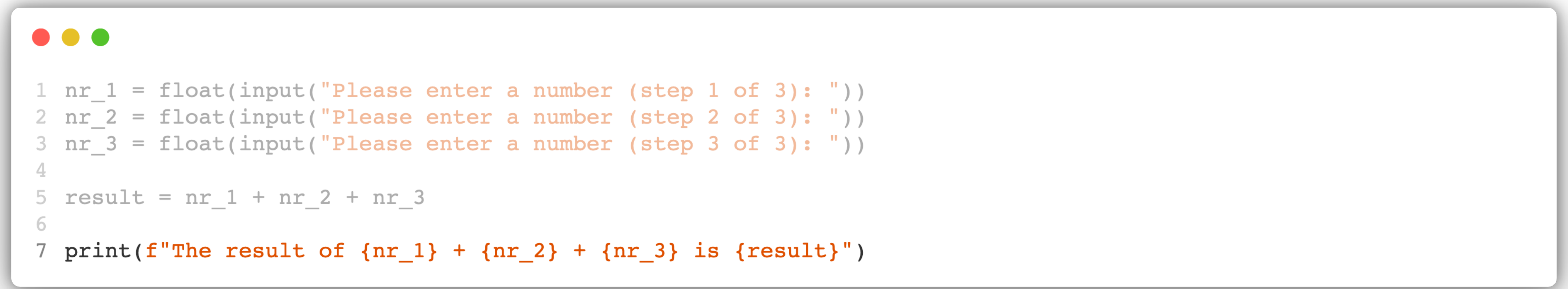


```python
nr_1 = float(input("Please enter a number (step 1 of 3): "))
nr_2 = float(input("Please enter a number (step 2 of 3): "))
nr_3 = float(input("Please enter a number (step 3 of 3): "))

result = nr_1 + nr_2 + nr_3

print(f"The result of {nr_1} + {nr_2} + {nr_3} is {result}")
```

Can you spot what the program will do?

A mysterious first program

Let us see our first code example:

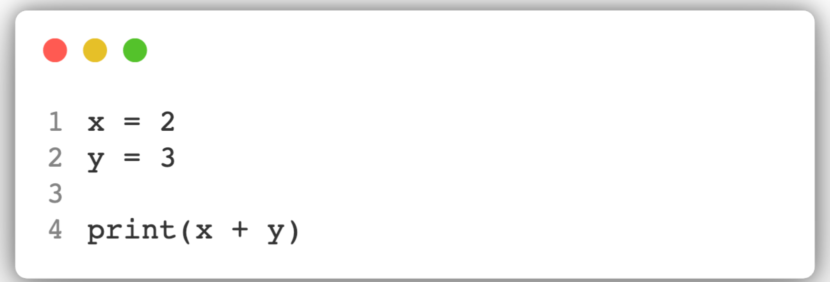


```python
nr_1 = float(input("Please enter a number (step 1 of 3): "))
nr_2 = float(input("Please enter a number (step 2 of 3): "))
nr_3 = float(input("Please enter a number (step 3 of 3): "))

result = nr_1 + nr_2 + nr_3

print(f"The result of {nr_1} + {nr_2} + {nr_3} is {result}")
```

Can you spot what the program will do?

Another Code-Example

Here is an even easier example:



```
x = 2
y = 3

print(x + y)
```

Do you know the cool thing, you can try it directly with your console!

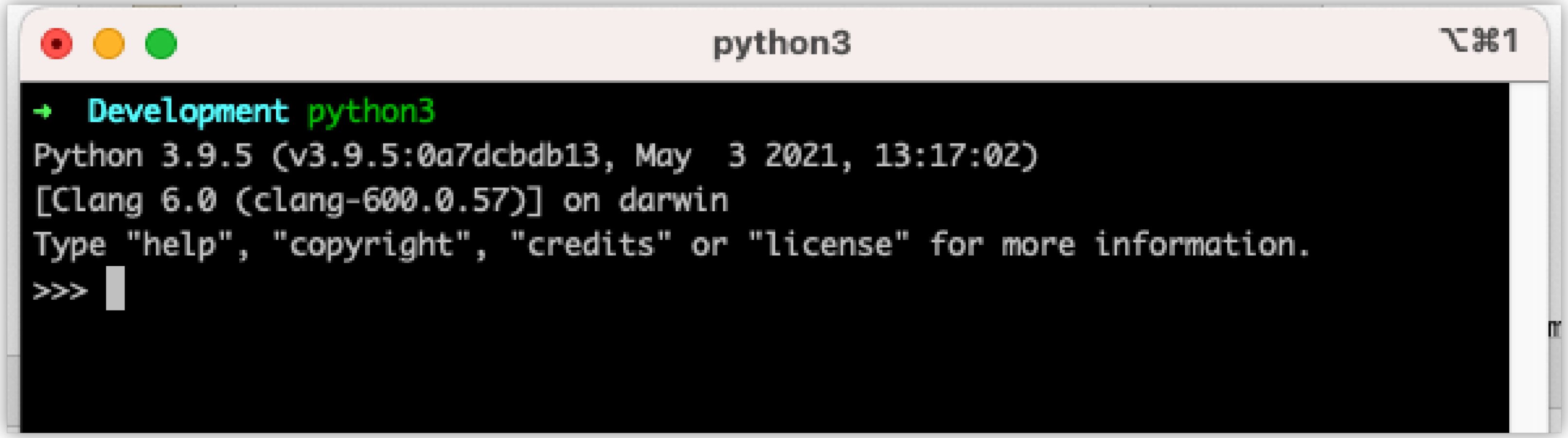
Let us start up your CMD, PowerShell or any other Console.

Another Code-Example

Windows: Python.exe or Python3.exe

Mac: Python or Python3

The version does not matter!

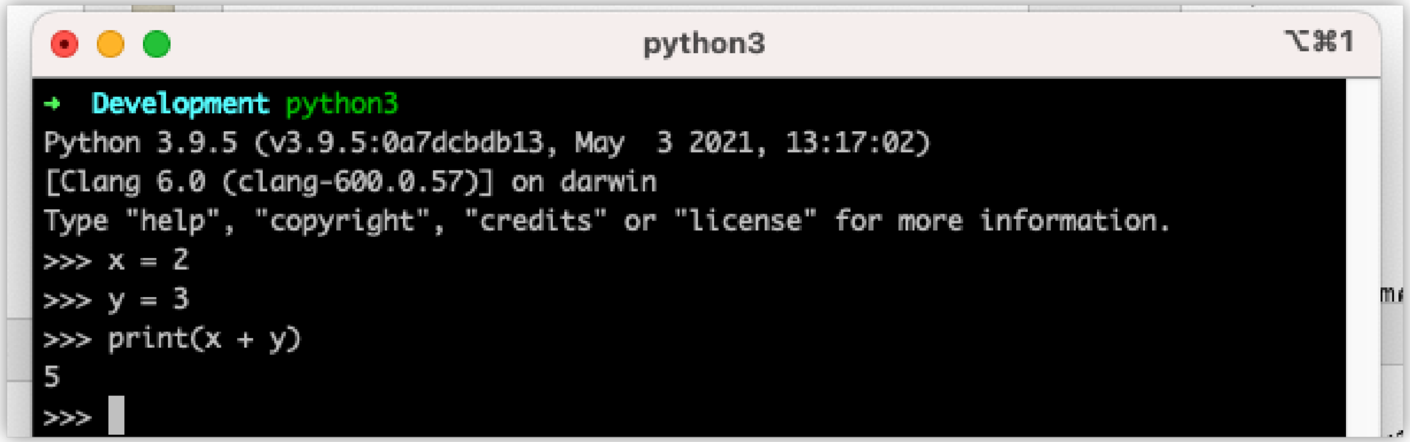


```
➜  Development python3
Python 3.9.5 (v3.9.5:0a7dcbdb13, May  3 2021, 13:17:02)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

Another Code-Example

Now please type line-by-line:

```
x = 2
y = 3

print(x + y)
```


python3 ⌥⌘1

```
→  Development python3
Python 3.9.5 (v3.9.5:0a7dcbdb13, May  3 2021, 13:17:02)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> x = 2
>>> y = 3
>>> print(x + y)
5
>>> 
```

Congratulations

You just have written your first python script!

End

That was all for this chapter