Intorduction to Python

Scientific Computing

Stefan Abi-Karam

Summer 2023

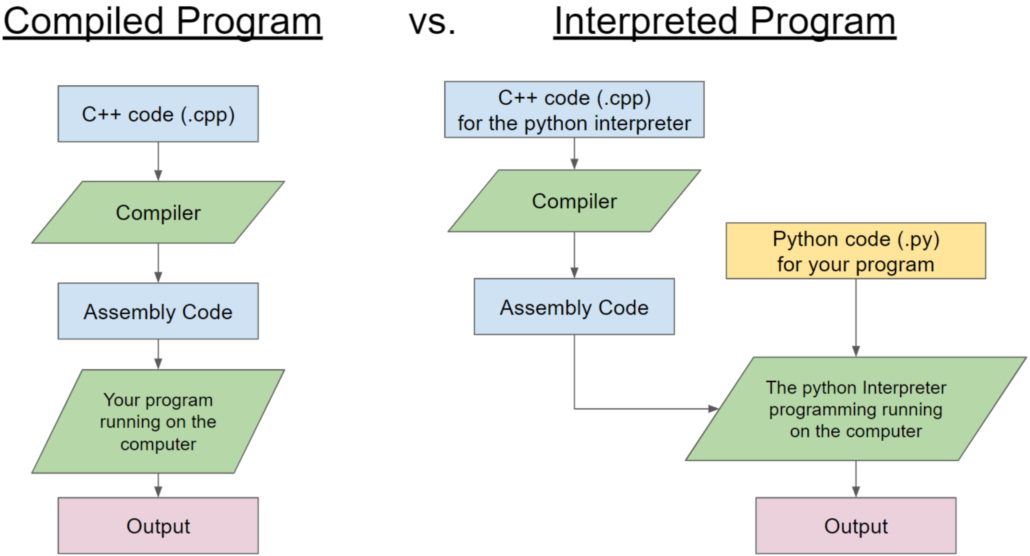
Table of Contents

Table of Contents

What Is Python

- Python is an interpreted, high-level, general-purpose programming language
- Interpreted means there is no python-to-assembly compiler, someone wrote and compiled a program in C (CPython) which is a program that just reads and executes your Python code
- Python also has a large standard library to achieve most tasks with relatively little code
- There are thousands of other libraries out that that other people have written that you can download and import into your code to do more complex stuff (ex. scientific computations and plotting)

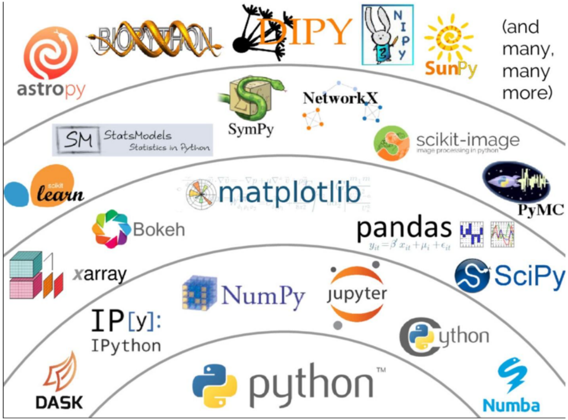
Compiled Program vs. Interpreted Program


Compiled Program
vs.
Interpreted Program
C++ code (.cpp)
Compiler
Assembly Code
Your program running on the computer
Output
C++ code (.cpp) for the python interpreter
Compiler
Assembly Code
Python code (.py) for your program
The python Interpreter programming running on the computer
Output

Why Choose Python

- There is the least amount of friction from the idea in your head to executing code that works and answers your research questions
- Python has a large community across many different fields of research
    - Very strong popularity in scientific computing and data science
- This means that many people in these fields has put in effort to write and maintain python libraries specific for scientific computing and data science
- In the absence of libraries for a particular task you are attempting or area of expertise you are working in, you can easily package and publish your code as its own library.

Why Choose Python



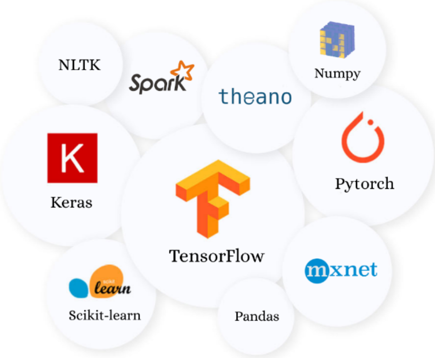


Table of Contents

1. Pyhton Overview

2. Installation and Usage

3. Python Core Concepts

3.1 Types

3.2 Variables

3.3 Operators / Expressions

3.4 Control Flow

3.5 Loops

3.6 Functions

4. Python Advanced Concepts

5. Python Standard Library

6. Python Packages

Python Installation and Versions

- When you install Python you download a version of the Python interpreter
- Please use version 3.9, it has the newest features and supported by most libraries you will use
- You can install multiple versions on the same computer, you just end up with more than one interpreter you can use
- If you want to download Python you can do so at their website: `https://www.python.org/`
- There's also a program called Conda / Anaconda that used a lot in the research community to manage Python installations and other libraries you may want to install and use
- **I recommend using Conda to install Python and manage your Python environment and packages if you are going to install Python on your own computer**

Using Python

Once Python is installed on your machine you can you use the Python command to run programs in the command line

```
> python your_file.py
```

Each Python installation also comes with a package manager named `pip` which you can also run from command line

```
> pip install numpy
```

If you are using Conda, you can call the `conda` command to install packages and manage Conda environments

```
> conda install numpy
```

Table of Contents

1. Pyhton Overview

2. Installation and Usage

3. Python Core Concepts

3.1 Types

3.2 Variables

3.3 Operators / Expressions

3.4 Control Flow

3.5 Loops

3.6 Functions

4. Python Advanced Concepts

5. Python Standard Library

6. Python Packages

What Are Types

Python Types

How Are Types Helpful

- Operations: Some operations between objects require the objects to be the same type
- Comparisons: Types allow for you to define how you can compare objects of different or same types
- Built in functions: Different types have different built in functions that let you things that make sense only with that type
- Conversion methods: Each type may also have defined how to convert it to another type

Table of Contents

1. Pyhton Overview

2. Installation and Usage

3. Python Core Concepts

3.1 Types

3.2 Variables

3.3 Operators / Expressions

3.4 Control Flow

3.5 Loops

3.6 Functions

4. Python Advanced Concepts

5. Python Standard Library

6. Python Packages

Table of Contents

1. Pyhton Overview

2. Installation and Usage

3. Python Core Concepts

3.1 Types

3.2 Variables

3.3 Operators / Expressions

3.4 Control Flow

3.5 Loops

3.6 Functions

4. Python Advanced Concepts

5. Python Standard Library

6. Python Packages

Table of Contents

1. Pyhton Overview

2. Installation and Usage

3. Python Core Concepts

3.1 Types

3.2 Variables

3.3 Operators / Expressions

3.4 Control Flow

3.5 Loops

3.6 Functions

4. Python Advanced Concepts

5. Python Standard Library

6. Python Packages