

# Python for Science and Technology

Shibu Meher

Materials Research Centre, IISc Bangalore

June 25, 2023



# Table of Contents

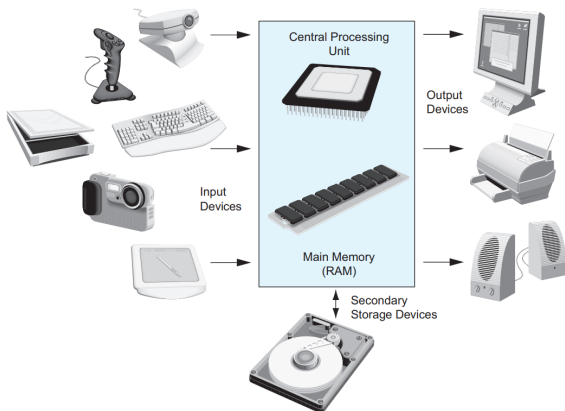
- 1 Basics of Computer Programming
- 2 Introduction to Python
- 3 Overview of Different Applications
- 4 Examples

# Computer

- A computer is a digital electronic device that can be programmed to carry out sequences of arithmetic or logical operations automatically. (Wikipedia)

# Computer

- A computer is a digital electronic device that can be programmed to carry out sequences of arithmetic or logical operations automatically. (Wikipedia)



# Hardware and Software

- **Hardware:** Physically tangible parts of computers, e.g. circuits, RAM, CPU, input devices, output devices, storage devices

# Hardware and Software

- **Hardware:** Physically tangible parts of computers, e.g. circuits, RAM, CPU, input devices, output devices, storage devices
- **Software:** Physically not tangible parts of computers, e.g. programs, data, protocols etc.

# Hardware and Software

- **Hardware:** Physically tangible parts of computers, e.g. circuits, RAM, CPU, input devices, output devices, storage devices
- **Software:** Physically not tangible parts of computers, e.g. programs, data, protocols etc.
  - Operating System/System Software: Microsoft Windows, Linux, Unix, DOS etc.

# Hardware and Software

- **Hardware:** Physically tangible parts of computers, e.g. circuits, RAM, CPU, input devices, output devices, storage devices
- **Software:** Physically not tangible parts of computers, e.g. programs, data, protocols etc.
  - Operating System/System Software: Microsoft Windows, Linux, Unix, DOS etc.
  - Data: Protocol (HTTP), File format (HTML, XML, JPEG)



# Hardware and Software

- **Hardware:** Physically tangible parts of computers, e.g. circuits, RAM, CPU, input devices, output devices, storage devices
- **Software:** Physically not tangible parts of computers, e.g. programs, data, protocols etc.
  - Operating System/System Software: Microsoft Windows, Linux, Unix, DOS etc.
  - Data: Protocol (HTTP), File format (HTML, XML, JPEG)
  - Libraries: Multimedia (OpenGL), Programming Libraries (C standard library, Standard Template Library)

# Hardware and Software

- **Hardware:** Physically tangible parts of computers, e.g. circuits, RAM, CPU, input devices, output devices, storage devices
- **Software:** Physically not tangible parts of computers, e.g. programs, data, protocols etc.
  - Operating System/System Software: Microsoft Windows, Linux, Unix, DOS etc.
  - Data: Protocol (HTTP), File format (HTML, XML, JPEG)
  - Libraries: Multimedia (OpenGL), Programming Libraries (C standard library, Standard Template Library)
  - User Interface: Graphical User Interface (Microsoft Windows), Text based interface (Command Line Interface)

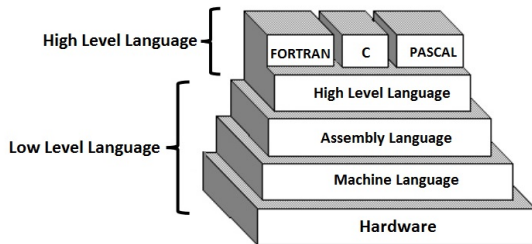
# Hardware and Software

- **Hardware:** Physically tangible parts of computers, e.g. circuits, RAM, CPU, input devices, output devices, storage devices
- **Software:** Physically not tangible parts of computers, e.g. programs, data, protocols etc.
  - Operating System/System Software: Microsoft Windows, Linux, Unix, DOS etc.
  - Data: Protocol (HTTP), File format (HTML, XML, JPEG)
  - Libraries: Multimedia (OpenGL), Programming Libraries (C standard library, Standard Template Library)
  - User Interface: Graphical User Interface (Microsoft Windows), Text based interface (Command Line Interface)
  - Application Software: Office suite, Compiler, Text Editor etc.

# Computer Programming

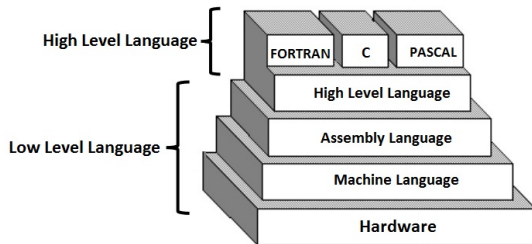
- Computer only understands 0s and 1s.

- Computer only understands 0s and 1s.



**Computer Language and its Types**

- Computer only understands 0s and 1s.



**Computer Language and its Types**

- **10 Most Popular Language in 2022:** Python, Java, JavaScript, C++, C, C, TypeScript, PHP, Perl, Ruby

# Introduction to Python

- **Python** is a very popular general-purpose interpreted, interactive, object-oriented, and high-level programming language.

- **Python** is a very popular general-purpose interpreted, interactive, object-oriented, and high-level programming language.
- **Why Python?**
  - Easy to use and learn
  - Mature and Supportive Python Communities
  - Hundreds of Libraries and Framework
  - Big Data, Machine Learning, Cloud Computing, Web Development (Server Side), Software Development, System Scripting, Scientific Computing, etc
  - Good for rapid prototyping as well as production-ready software development



- **Python** is a very popular general-purpose interpreted, interactive, object-oriented, and high-level programming language.
- **Why Python?**
  - Easy to use and learn
  - Mature and Supportive Python Communities
  - Hundreds of Libraries and Framework
  - Big Data, Machine Learning, Cloud Computing, Web Development (Server Side), Software Development, System Scripting, Scientific Computing, etc
  - Good for rapid prototyping as well as production-ready software development
- **Disadvantages of Python:**
  - Speed Limitations (Use Cython)
  - Weak in Mobile Computing

# A Simple Code in C Vs Python

## C (main.c)

```
#include <stdio.h>
int main() {
int number1, number2, sum;
printf("Enter two integers: ");
scanf("%d %d", &number1, &number2);
// calculating sum
sum = number1 + number2;
printf("%d + %d = %d", number1, number2, sum);
return 0;
}
```

# A Simple Code in C Vs Python

## C (main.c)

```
#include <stdio.h>
int main() {
int number1, number2, sum;
printf("Enter two integers: ");
scanf("%d %d", &number1, &number2);
// calculating sum
sum = number1 + number2;
printf("%d + %d = %d", number1, number2, sum);
return 0;
}
```

## Python (main.py)

```
number1, number2 = input("Enter two integers: ").split()
# calculating sum
sum = int(number1) + int(number2)
print('{0} + {1} = {2}'.format(number1, number2, sum))
```

- **Physics:**

- Computational solid state physics, Computational mechanics, Computational electrodynamics, Computational astrophysics, Computational biophysics etc.

- **Physics:**

- Computational solid state physics, Computational mechanics, Computational electrodynamics, Computational astrophysics, Computational biophysics etc.

- **Chemistry:**

- Computational chemistry (Mainly to solve quantum many body problem)

- **Physics:**

- Computational solid state physics, Computational mechanics, Computational electrodynamics, Computational astrophysics, Computational biophysics etc.

- **Chemistry:**

- Computational chemistry (Mainly to solve quantum many body problem)

- **Mathematics:**

- Numerical Analysis, Probability, statistics, integration, derivatives etc.

- **Physics:**

- Computational solid state physics, Computational mechanics, Computational electrodynamics, Computational astrophysics, Computational biophysics etc.

- **Chemistry:**

- Computational chemistry (Mainly to solve quantum many body problem)

- **Mathematics:**

- Numerical Analysis, Probability, statistics, integration, derivatives etc.

- **Biology:**

- Computational biology (includes bioinformatics, Computational Neuroscience, Computational Pharmacology)

- **Physics:**

- Computational solid state physics, Computational mechanics, Computational electrodynamics, Computational astrophysics, Computational biophysics etc.

- **Chemistry:**

- Computational chemistry (Mainly to solve quantum many body problem)

- **Mathematics:**

- Numerical Analysis, Probability, statistics, integration, derivatives etc.

- **Biology:**

- Computational biology (includes bioinformatics, Computational Neuroscience, Computational Pharmacology)

- In all the above cases there is already available python packages or python is used to efficiently do the calculations using other software and analysing the data after calculations.



# Some Popular Libraries for Science

- NumPy: the fundamental package for scientific computing with Python, adding support for large, multidimensional arrays and matrices, along with a large library of high-level mathematical functions to operate on these arrays.

# Some Popular Libraries for Science

- NumPy: the fundamental package for scientific computing with Python, adding support for large, multidimensional arrays and matrices, along with a large library of high-level mathematical functions to operate on these arrays.
- SciPy: Fundamental algorithms for scientific computing in Python

# Some Popular Libraries for Science

- NumPy: the fundamental package for scientific computing with Python, adding support for large, multidimensional arrays and matrices, along with a large library of high-level mathematical functions to operate on these arrays.
- SciPy: Fundamental algorithms for scientific computing in Python
- SymPy: library for symbolic computation, offering features ranging from basic symbolic arithmetic to calculus, algebra, discrete mathematics, and quantum physics

# Some Popular Libraries for Science

- NumPy: the fundamental package for scientific computing with Python, adding support for large, multidimensional arrays and matrices, along with a large library of high-level mathematical functions to operate on these arrays.
- SciPy: Fundamental algorithms for scientific computing in Python
- SymPy: library for symbolic computation, offering features ranging from basic symbolic arithmetic to calculus, algebra, discrete mathematics, and quantum physics
- Astropy: collection of packages designed for use in astronomy

# Some Popular Libraries for Science

- NumPy: the fundamental package for scientific computing with Python, adding support for large, multidimensional arrays and matrices, along with a large library of high-level mathematical functions to operate on these arrays.
- SciPy: Fundamental algorithms for scientific computing in Python
- SymPy: library for symbolic computation, offering features ranging from basic symbolic arithmetic to calculus, algebra, discrete mathematics, and quantum physics
- Astropy: collection of packages designed for use in astronomy
- Biopython: Python tools for computational biology and bioinformatic

# Some Popular Libraries for Science

- NumPy: the fundamental package for scientific computing with Python, adding support for large, multidimensional arrays and matrices, along with a large library of high-level mathematical functions to operate on these arrays.
- SciPy: Fundamental algorithms for scientific computing in Python
- SymPy: library for symbolic computation, offering features ranging from basic symbolic arithmetic to calculus, algebra, discrete mathematics, and quantum physics
- Astropy: collection of packages designed for use in astronomy
- Biopython: Python tools for computational biology and bioinformatic
- Bokey: quickly and easily create interactive plots, dashboards, and data applications

# Some Popular Libraries for Science

- NumPy: the fundamental package for scientific computing with Python, adding support for large, multidimensional arrays and matrices, along with a large library of high-level mathematical functions to operate on these arrays.
- SciPy: Fundamental algorithms for scientific computing in Python
- SymPy: library for symbolic computation, offering features ranging from basic symbolic arithmetic to calculus, algebra, discrete mathematics, and quantum physics
- Astropy: collection of packages designed for use in astronomy
- Biopython: Python tools for computational biology and bioinformatic
- Bokey: quickly and easily create interactive plots, dashboards, and data applications
- 40 Most Popular Python Scientific Libraries  
(<https://www.stxnnext.com/blog/most-popular-python-scientific-libraries/>)

- Artificial Intelligence and Machine Learning (Keras, TensorFlow, Sci-kit learn, PyTorch, Caffee, Seaborn, NumPy, Pandas, Matplotlib)



# Applications in Technology

- Artificial Intelligence and Machine Learning (Keras, TensorFlow, Sci-kit learn, PyTorch, Caffee, Seaborn, NumPy, Pandas, Matplotlib)
- Automation and Robotics (Dart, PyDy, pyro, PyRobot)
- Image Processing (OpenCV, Blender, PIL, Houdini)

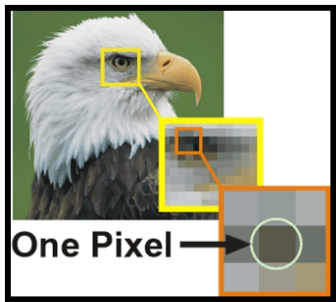
# Applications in Technology

- Artificial Intelligence and Machine Learning (Keras, TensorFlow, Sci-kit learn, PyTorch, Caffee, Seaborn, NumPy, Pandas, Matplotlib)
- Automation and Robotics (Dart, PyDy, pyro, PyRobot)
- Image Processing (OpenCV, Blender, PIL, Houdini)
- Web Scrapping (Selenium, Scrapy)
- Desktop GUI (PyQt, PyGtk, Kivy, Tkinter, WxPython, PyGUI, and PySide)

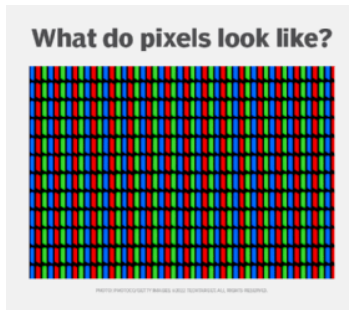
# Applications in Technology

- Artificial Intelligence and Machine Learning (Keras, TensorFlow, Sci-kit learn, PyTorch, Caffee, Seaborn, NumPy, Pandas, Matplotlib)
- Automation and Robotics (Dart, PyDy, pyro, PyRobot)
- Image Processing (OpenCV, Blender, PIL, Houdini)
- Web Scrapping (Selenium, Scrapy)
- Desktop GUI (PyQt, PyGtk, Kivy, Tkinter, WxPython, PyGUI, and PySide)
- Game Development (PyGame, PySoy)
- Web Development (Django, Pyramid, Flask, and Bottle)

# Basics of Image Processing



Concept of pixel



Pixels

# Hand-written Digits Recognition

Demonstration using Google Colab: [Click here for notebook](#)