**History of python**

* Python is one of the **General-purpose,** **High Level**, **Object Oriented,** and **Interpreted** programming language.
* Python was developed by **Guido Van Rossum** in 1989 while working at Centrum Wiskunde & Informatica (CWI) in the Netherlands.
* But officially Python was made available to the public in 1991. The official date was 20th Feb 1991.
* The procedure of Python programming language uses ABC Programming language.
* Python software is maintained by PSF (Python Software Foundation)
* [www.python.org](http://www.python.org)
* The name Python was selected from the TV show “Monty Python’s flying circus” which was broadcasted on BBC from 1969 to 1974.
* Python software has two versions   
  1. Python 2 x 🡪 Outdated version   
  2. Python 3 x 🡪 Current version example (3.11.2)
* There is no backward compatibility (which means that Python 2 is not compatible with Python 3 in some cases).
* Python is recommended as the first programming language for beginners.

**Chapter – 2**

**Features of Python**

**Frameworks**

Django, pyramid

**Tkinter**

(Library for GUI apps

)

**Realtime application developed by python**

**CLI based**

(PIP, Git, IPython)

**Application Installer**

(Setuptools, pip, virtualenv, tox, PyInstaller, Sphinx)

CRM, Accounting Software, ERP

Basic standard Applications

Optical character recognition (OCR), Image segmentation

**Libraries:** Pygame, Arcade

**Game Engines:** Panda3D, PyOpenGL, Ren’Py, PyOgre

NumPy, Pandas, Matplotlib, Scikit-learn, TensorFlow, Keras, PyTorch

MoviePy, PyDub, OpenCV, VLC bindings

**Web Applications**

Python is a versatile programming language that is widely used in the development of web applications. Here are some examples of web applications that can be built using Python:

1. **Flask:** Flask is a micro web framework written in Python that is used to build web applications. It is a lightweight framework that allows developers to quickly build web applications without much overhead.
2. **Django:** Django is a high-level web framework written in Python that is designed for rapid development and clean, pragmatic design. It includes several built-in features such as authentication, database management, and templates, making it easy to build complex web applications.
3. **Pyramid:** Pyramid is a web framework that is designed for building large-scale web applications. It is highly customizable and flexible, making it suitable for a wide range of applications.
4. **TurboGears:** TurboGears is a full-stack web framework that is built on top of several Python libraries, including SQLAlchemy, Mako, and Repoze. It includes several built-in features such as authentication, security, and internationalization.
5. **CherryPy:** CherryPy is a minimalist web framework that is designed for building web applications that require high performance and scalability. It includes a built-in web server and supports HTTP/1.1 and WSGI.

These are just a few examples of the many web applications that Python can build. Python's versatility and ease of use make it a popular choice for web developers around the world.

# **GUI Applications**

Python has several popular GUI frameworks that can be used to develop desktop applications. Here are some examples:

1. **Tkinter:** Tkinter is the standard Python GUI framework and is included with most Python installations. It is a lightweight and simple GUI toolkit that is easy to use and learn.
2. **PyQT:** PyQT is a set of Python bindings for the Qt application framework, which is widely used for developing cross-platform desktop applications. It is a powerful and flexible GUI framework that includes support for a wide range of widgets and controls.
3. **wxPython:** wxPython is a set of Python bindings for the wxWidgets C++ GUI toolkit. It includes a wide range of widgets and controls and provides a native look and feels on a variety of platforms.
4. **Kivy:** Kivy is an open-source Python library for developing multi-touch applications. It is designed for creating applications that run on multiple platforms, including Windows, macOS, Linux, Android, and iOS.
5. **PySide:** PySide is another set of Python bindings for the Qt application framework. It is similar to PyQT but has a different licensing model that may be more suitable for some developers.

These are just a few examples of the many GUI frameworks available for Python. Python's simplicity and ease of use make it a popular choice for developing desktop applications.

# **Console Base Applications**

Python is a popular language for developing console-based applications, also known as command-line interfaces (CLI). Here are some examples of console-based applications developed using Python:

1. **pip:** pip is a package management system used to install and manage software packages written in Python. It is a console-based application that is built using Python.
2. **Git:** Git is a popular version control system used by developers to manage source code. It includes a command-line interface that is built using Python.
3. **YouTube-DL:** YouTube-DL is a command-line utility used to download videos from YouTube and other video-sharing websites. It is written in Python and can be used from the command line.
4. **Ansible:** Ansible is a powerful automation tool used for configuration management, application deployment, and task automation. It includes a command-line interface that is built using Python.
5. **IPython:** IPython is an interactive Python shell that provides enhanced introspection, debugging, and code execution features. It is built using Python and is used primarily from the command line.

These are just a few examples of the many console-based applications that can be developed using Python. Python's simplicity and ease of use make it a popular choice for developing command-line interfaces.

# **Software Development Building Tools**

Python has several popular software development building tools that are widely used by developers. Here are some examples:

1. **Setuptools:** setuptools is a package management system used to package and distribute Python libraries and applications. It provides tools for building, distributing, and installing Python packages.
2. **pip:** pip is a package management system used to install and manage software packages written in Python. It provides a simple command-line interface for installing, upgrading, and removing packages.
3. **virtualenv:** virtualenv is a tool used to create isolated Python environments. It allows developers to work on multiple projects with different dependencies without conflicts.
4. **tox:** tox is a tool used to automate the testing and packaging of Python software. It provides a simple way to run tests in multiple environments and ensures that packages are built and distributed correctly.
5. **PyInstaller:** PyInstaller is a tool used to bundle Python applications into standalone executable files. It supports multiple platforms and can be used to package applications into a single file for easy distribution.
6. **Sphinx:** Sphinx is a tool used to generate documentation from Python source code. It provides a simple way to generate HTML, PDF, and other document formats from Python code.

These are just a few examples of the many software development building tools available for Python. Python's simplicity and ease of use make it a popular choice for building software development tools.

# **Business Applications**

Python is a versatile language that is widely used in the development of business applications. Here are some examples of business applications that can be built using Python:

1. **Inventory management system:** Python can be used to build an inventory management system that tracks inventory levels, manages purchase orders, and generates reports.
2. **Customer relationship management (CRM) system:** Python can be used to build a CRM system that manages customer data, tracks sales leads, and automates marketing campaigns.
3. **Accounting software:** Python can be used to build accounting software that manages accounts payable and receivable, generates financial reports, and tracks expenses.
4. **Human resources management system (HRMS):** Python can be used to build an HRMS that manages employee data, tracks time and attendance, and automates payroll.
5. **Project management tool:** Python can be used to build a project management tool that tracks project timelines, manages resources, and assigns tasks.
6. **E-commerce platform:** Python can be used to build an e-commerce platform that manages online sales, processes payments, and tracks inventory.
7. **Data analytics tool:** Python can be used to build a data analytics tool that collects and analyses data, generates reports, and provides insights into business operations.

These are just a few examples of the many business applications that Python can build. Python's flexibility and ease of use make it a popular choice for building a wide range of business applications.

# **Standard Application**

Python is a versatile programming language that can be used for a wide range of applications. Here are some examples of standard applications that can be implemented using Python:

1. **Web development:** Python is commonly used for web development using frameworks such as Django and Flask. These frameworks provide a lot of features for building web applications.
2. **Data analysis and visualization:** Python has libraries such as NumPy, Pandas, and Matplotlib that can be used for data analysis and visualization. These libraries provide efficient data structures and visualization tools for analyzing and presenting data.
3. **Machine learning:** Python has several libraries such as TensorFlow, Keras, and PyTorch that can be used for machine learning tasks. These libraries allow you to train and deploy machine-learning models for various applications.
4. **Scientific computing:** Python has libraries such as SciPy and SymPy that can be used for scientific computing tasks. These libraries provide functions and tools for numerical integration, optimization, and symbolic mathematics.
5. **Game development:** Python has libraries such as Pygame and Arcade that can be used for game development. These libraries provide functions and tools for building 2D and 3D games.
6. **Desktop applications:** Python can be used to build desktop applications using frameworks such as PyQt and wxPython. These frameworks provide a lot of features for building cross-platform desktop applications.
7. **Automation:** Python can be used for automation tasks such as web scraping, file management, and process automation. Python provides libraries such as BeautifulSoup and Selenium that can be used for web scraping and automation.

These are just a few examples of the many standard applications that can be implemented using Python. With its large community and extensive library support, Python is a popular choice for a wide range of applications.

# **Image Processing Application**

Python is a popular programming language used for image processing and computer vision applications due to its simplicity, ease of use, and availability of a wide range of libraries and frameworks. Here are some examples of image-processing applications in Python:

1. **Face detection and recognition:** Python has several libraries such as OpenCV, Dlib, and face\_recognition that can be used to detect and recognize faces in images and videos.
2. **Image segmentation:** Python provides libraries such as scikit-image, SimpleITK, and OpenCV that can be used to segment images into regions of interest.
3. **Object detection:** Python has libraries such as TensorFlow, Keras, and PyTorch that can be used for object detection tasks. These libraries allow you to train deep-learning models to detect objects in images and videos.
4. **Image filtering and enhancement:** Python provides libraries such as Pillow, scikit-image, and OpenCV that can be used to apply various filters to images to enhance their quality.
5. **Image restoration:** Python provides libraries such as scikit-image and OpenCV that can be used for image restoration tasks such as denoising, deblurring, and inpainting.
6. **Medical image analysis:** Python provides libraries such as SimpleITK, PyMedPhys, and ITK that can be used for medical image analysis tasks such as segmentation, registration, and classification.
7. **Optical character recognition (OCR):** Python provides libraries such as pytesseract and OpenCV that can be used for OCR tasks to extract text from images.

These are just a few examples of the many image-processing applications that can be implemented in Python using various libraries and frameworks.

# **Game Development**

Python is a versatile language that can be used for game development. Here are some examples of game development applications that can be implemented using Python:

1. **Pygame:** Pygame is a library for game development that is based on the Simple DirectMedia Layer (SDL). Pygame provides functions and tools for building 2D games. It is a popular choice for game development in Python.
2. **Panda3D:** Panda3D is a game engine that is designed for 3D game development. It provides a lot of features for building 3D games such as physics simulation, collision detection, and animation.
3. **Arcade:** Arcade is a game development library that is designed for 2D game development. It provides a lot of features for building 2D games such as sprites, animation, and sound effects.
4. **PyOpenGL:** PyOpenGL is a library for game development that provides Python bindings for the OpenGL graphics API. It can be used to build 3D games and applications that require advanced graphics rendering.
5. **Ren'Py:** Ren'Py is a visual novel engine that can be used for building interactive stories and games. It provides a lot of features for building visual novels such as scripting, animation, and character customization.
6. **PyOgre:** PyOgre is a Python wrapper for the OGRE 3D graphics engine. It can be used to build 3D games and applications that require advanced graphics rendering.
7. **PyBullet:** PyBullet is a physics engine that can be used for game development. It provides a lot of features for building games that require physics simulation such as racing games and sports games.

These are just a few examples of the many game development applications that can be implemented using Python. With its large community and extensive library support, Python is a popular choice for game development.

**Data Science, Artificial Intelligence, And Machine Learning**

Python is a popular language for data science, artificial intelligence, and machine learning due to its simplicity, ease of use, and availability of a wide range of libraries and frameworks. Here are some examples of data science, artificial intelligence, and machine learning applications in Python:

1. **NumPy:** NumPy is a library for numerical computing in Python. It provides functions for array manipulation, linear algebra, and statistical analysis. It is a fundamental library for data science and machine learning in Python.
2. **Pandas:** Pandas is a library for data manipulation and analysis in Python. It provides data structures for handling structured data such as data frames and time series. It is widely used for data analysis in Python.
3. **Matplotlib:** Matplotlib is a library for data visualization in Python. It provides functions and tools for creating various types of plots and charts. It is widely used for visualizing data in Python.
4. **Scikit-learn:** Scikit-learn is a library for machine learning in Python. It provides functions and tools for classification, regression, clustering, and dimensionality reduction. It is widely used for building machine-learning models in Python.
5. **TensorFlow:** TensorFlow is an open-source library for machine learning developed by Google. It provides functions and tools for building deep learning models. It is widely used for building neural networks in Python.
6. **Keras:** Keras is a high-level API for building deep learning models in Python. It provides a simple interface for building and training neural networks. It is widely used for building deep-learning models in Python.
7. **PyTorch:** PyTorch is an open-source library for machine learning developed by Facebook. It provides functions and tools for building deep learning models. It is widely used for building neural networks in Python.

These are just a few examples of the many data science, artificial intelligence, and machine learning applications that can be implemented in Python using various libraries and frameworks.

# **Audio/Video Apps**

Python can be used to develop various audio and video applications. Here are some examples:

1. **MoviePy:** MoviePy is a Python library for video editing that can be used to create, modify, and concatenate video clips. It can be used for tasks such as video cutting, trimming, and merging.
2. **PyDub:** PyDub is a Python library for audio processing that can be used to manipulate audio files. It can be used for tasks such as cutting, concatenating, and applying effects to audio files.
3. **OpenCV:** OpenCV is a popular computer vision library that can be used for object detection and image processing tasks. It also provides features for video processing such as capturing video from a camera or file and applying various effects to video streams.
4. **PyGame:** PyGame is a Python library for game development that can also be used for multimedia applications such as audio and video playback. It provides features for playing audio and video files, as well as capturing audio and video from external sources.
5. **GStreamer:** GStreamer is a multimedia framework that provides a pipeline-based architecture for processing multimedia data. It can be used to build complex multimedia applications such as video editors and media players.
6. **VLC bindings:** VLC bindings provide a Python interface for the VLC media player, which can be used for playing audio and video files, as well as streaming media over a network.

These are just a few examples of the many audio and video applications that can be developed using Python. With its extensive library support and ease of use, Python can be a powerful tool for multimedia development.

**Features of python**

Features of a programming language are nothing but services/facilities provided by language Developers in the language, which are used by language programmers for developing real-time applications.**Top of Form**

1. Simple and easy to learn.
2. Freeware and open source.
3. Dynamically programming language.
4. Platform-independent language.
5. Portable.
6. Interpreted programming language.
7. High-level programming language.
8. Procedural and object-oriented.
9. Robust.
10. Extensible programming language.
11. Embedded programming language.
12. Support Third-party APIs.

### **Simple and easy to learn**

Python is simple and easy to learn because when we read it, its syntax is readable like the English language. Python language provides a rich set of modules (Libraries) and APIs (Application programming interface).

Python provides developer-friendly syntaxes so that we can develop error-free programs in a limited space of time.

Python programming provides an in-built facility called a garbage collector, which collects unused memory space and improves the performance of Python-based applications.

**What is a Garbage collector:** A Garbage collector is one of the software components in Python software, which is running in the background of a regular Python program whose role is to collect/remove unused memory space.

Hence, the garbage collector is taking care of memory management.

Python is often considered simple and easy to learn for several reasons:

1. Readability: Python code is easy to read and understand, as it uses a clear and concise syntax that resembles the English language.
2. Minimalist design: Python has a minimalist design philosophy, which means that it aims to do more with less code. This makes it easier for beginners to write and understand code.
3. Large community support: Python has a large and active community that provides helpful resources and support to beginners, including online tutorials, forums, and libraries.
4. Versatility: Python is a versatile language that can be used for a wide range of applications, including web development, data analysis, artificial intelligence, and more. This makes it appealing to people from different backgrounds and industries.
5. Interpreted language: Python is an interpreted language, which means that it can execute code line by line, making it easier for beginners to debug and test their code.

Python's simplicity and ease of use make it an ideal language for beginners who want to learn programming.

### **Freeware and open source**

We can use Python software without any license and it is freeware its source code is open so that we can customize based on our requirements.