Python Tutorial | Learn Python Programming

This Programming Language Python Tutorial is very well suited for beginners and also for experienced programmers. This specially designed free Python tutorial will help you learn Python programming most efficiently, with all topics from basics to advanced (like Web-scraping, Django, Learning, etc.) with examples.

What is Python?

Python is a high-level, general-purpose, and very popular programming language. Python programming language (latest Python 3) is being used in web development, and Machine Learning applications, along with all cutting-edge technology in Software Industry. Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

Writing your first Python Program to Learn Python Programming

There are two ways you can execute your Python program:

1. First, we write a program in a file and run it one time.
2. Second, run a code line by line.

Here we provided the latest Python 3 version compiler where you can edit and compile your written code directly with just one click of the RUN Button. So test yourself with Python first exercises.

Python3

```
# Python Program to print Hello World
print("Hello World! I Don't Give a Bug")
```

Run on IDE

Output

```
Hello World! I Don't Give a Bug
```

Let us now see what will you learn in this Python Tutorial, in detail:

The first and foremost step to get started with Python tutorial is to setup Python in your system. Below are the steps based your system requirements:

Setting up Python

- Download and Install Python 3 Latest Version
- How to set up Command Prompt for Python in Windows10
- Setup Python VS Code or PyCharm
- Creating Python Virtual Environment in Windows and Linux

Now let us deep dive into the basics and components to learn Python Programming:

Getting Started with Python Programming

Welcome to the Python tutorial section! Here, we'll cover the essential elements you need to kickstart your journey in Python programming. From syntax and keywords to comments, variables, and indentation, we'll explore the foundational concepts that underpin Python development.

- Learn Python Basics
- Syntax
- Keywords in Python
- Comments in Python
- Learn Python Variables
- Learn Python Data Types

- Indentation and why is it important in Python

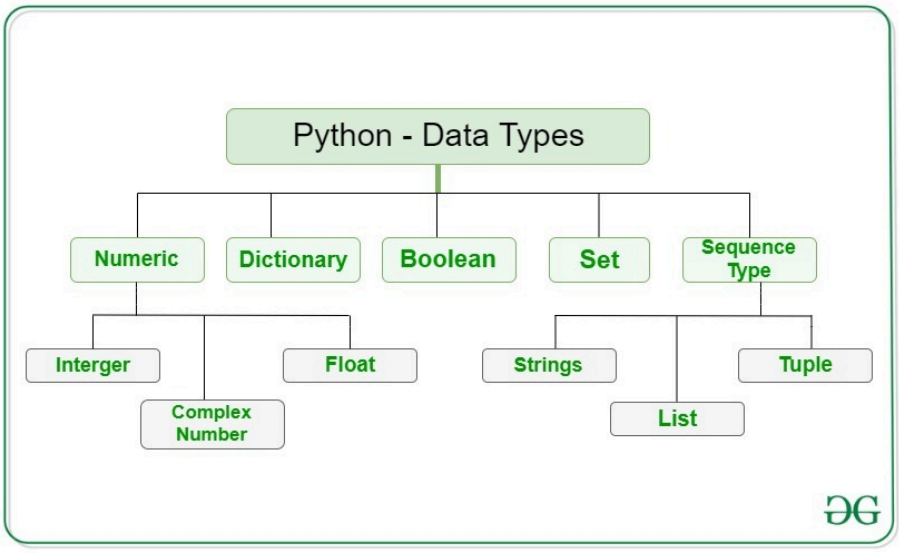
Learn Python Input/Output

In this segment, we delve into the fundamental aspects of handling input and output operations in Python, crucial for interacting with users and processing data effectively. From mastering the versatile print() function to exploring advanced formatting techniques and efficient methods for receiving user input, this section equips you with the necessary skills to harness Python's power in handling data streams seamlessly.

- Python print() function
- f-string in Python
- Print without newline in Python
- Python | end parameter in print()
- Python | sep parameter in print()
- Python | Output Formatting
- Taking Input in Python
- Taking Multiple Inputs from users in Python

Python Data Types

Python offers, enabling you to manipulate and manage data with precision and flexibility. Additionally, we'll delve into the dynamic world of data conversion with casting, and then move on to explore the versatile collections Python provides, including lists, tuples, sets, dictionaries, and arrays.



By the end of this section, you'll not only grasp the essence of Python's data types but also wield them proficiently to tackle a wide array of programming challenges with confidence.

- Strings
- Numbers
- Booleans
- Python List
- Python Tuples

- Python Sets
- Python Dictionary
- Python Arrays
- Type Casting

Python Operators

From performing basic arithmetic operations to evaluating complex logical expressions, we'll cover it all. We'll delve into comparison operators for making decisions based on conditions, and then explore bitwise operators for low-level manipulation of binary data. Additionally, we'll unravel the intricacies of assignment operators for efficient variable assignment and updating. Lastly, we'll demystify membership and identity operators, such as in and is, enabling you to test for membership in collections and compare object identities with confidence.

- Arithmetic operators
- Comparison Operators
- Logical Operators
- Bitwise Operators
- Assignment Operators
- Membership & Identity Operators | Python "in", and "is" operator

Python Conditional Statement

These statements are pivotal in programming, enabling dynamic decision-making and code branching. In this section of Python Tutorial, we'll explore Python's conditional logic, from basic if…else statements to nested conditions and the concise ternary operator. We'll also introduce the powerful match case statement, new in Python 3.10. By the end, you'll master these constructs, empowering you to write clear, efficient code that responds intelligently to various scenarios. Let's dive in and unlock the potential of Python's conditional statements.

- If..else
- Nested-if statement
- Ternary Condition in Python
- Match Case Statement

Python Loops

Here, we'll explore Python's loop constructs, including the for and while loops, along with essential loop control statements like break, continue, and pass. Additionally, we'll uncover the concise elegance of list and dictionary comprehensions for efficient data manipulation. By mastering these loop techniques, you'll streamline your code for improved readability and performance.

- For Loop
- While Loop

- Loop control statements (break, continue, pass)
- Python List Comprehension
- Python Dictionary Comprehension

Python Functions

Functions are the backbone of organized and efficient code in Python. Here, we'll explore their syntax, parameter handling, return values, and variable scope. From basic concepts to advanced techniques like closures and decorators. Along the way, we'll also introduce versatile functions like range(), and powerful tools such as *args and **kwargs for flexible parameter handling. Additionally, we'll delve into functional programming with map, filter, and lambda functions.

- Python Function syntax
- Arguments and Return Values in Python Function
- Python Function Global and Local Scope Variables
- Use of pass Statement in Function
- Return statemen in Python Function
- Python range() function
- *args and **kwargs in Python Function
- Python closures
- Python 'Self' as Default Argument
- Decorators in Python
- Python closures
- Map Function
- Filter Function
- Reduce Function
- Lambda Function

Python OOPs Concepts

In this segment, we'll explore the core principles of object-oriented programming (OOP) in Python. From encapsulation to inheritance, polymorphism, abstract classes, and iterators, we'll cover the essential concepts that empower you to build modular, reusable, and scalable code.

- Python Classes and Objects
- Polymorphism
- Inheritance
- Abstract
- Encapsulation
- Iterators

Python Exception Handling

In this section of Python Tutorial, we'll explore how Python deals with unexpected errors, enabling you to write robust and fault-tolerant code. We'll cover file handling, including reading from and writing to files, before diving into exception handling with try and except blocks. You'll also learn about user-defined exceptions and Python's built-in exception types.

- Python File Handling
- Python Read Files
- Python Write/Create Files
- Exception handling
- User defined Exception
- Built-in Exception
- Try and Except in Python

Python Packages or Libraries

The biggest strength of Python is a huge collection of standard libraries which can be used for the following:

- Built-in Modules in Python
- Python DSA Libraries
- Machine Learning
- Python GUI Libraries
- Web Scraping Pakages
- Game Development Packages
- Web Frameworks like, Django, Flask
- Image processing (like OpenCV, Pillow)

Python Collections

Here, we'll explore key data structures provided by Python's collections module. From counting occurrences with Counters to efficient queue operations with Deque, we'll cover it all. By mastering these collections, you'll streamline your data management tasks in Python.

- Counters
- Heapq
- Deque
- OrderedDict
- Defaultdict

Python vs. Other Programming Languages

Here's a comparison of Python with the programming languages C, C++, and Java in a table format:

Comparing Python with Other Programming Languages

FEATURE	PYTHON	C	C++	JAVA
TYPE	Interpreted	Compiled	Compiled	Compiled
PARADIGM	Multi-paradigm procedural, functional)	Procedural,	Multi-paradigm ject-oriented, generic)	Object-ori- structured
MEMORY MANAGEMENT	Automatic	Manual	Manual	Automatic
SYNTAX	Simple	Complex	Complex	Complex
POPULARITY	High	High	High	High
USE CASES	Web development, data analysis, machine learning	System programming, embedded systems, game development	System programming, game development, high-performance applications	Large-scale applications, enterprise software
NOTABLE FRAMEWORKS/LIBRARIES	Django, Flask	Standard Library	Standard Library, Boost	Spring, Hibernate
LEARNING CURVE	Moderate	Steep	Steep	Moderate
COMMUNITY SUPPORT	Strong	Strong	Strong	Strong
JOB MARKET	Abundant	Abundant	Abundant	Abundant

Let us now begin learning about various important steps required in this Python Tutorial.

Learn More about Python with diffent Applications:

Python is a versatile and widely-used programming language with a vast ecosystem. Here are some areas where Python is commonly used:

1. **Web Development**: Python is used to build web applications using frameworks like Django, Flask, and Pyramid. These frameworks provide tools and libraries for handling web requests, managing databases, and more.

2. **Data Science and Machine Learning**: Python is popular in data science and machine learning due to libraries like NumPy, pandas, Matplotlib, and scikit-learn. These libraries provide tools for data manipulation, analysis, visualization, and machine learning algorithms.
3. **Artificial Intelligence and Natural Language Processing**: Python is widely used in AI and NLP applications. Libraries like TensorFlow, Keras, PyTorch, and NLTK provide tools for building and training neural networks, processing natural language, and more.
4. **Game Development**: Python can be used for game development using libraries like Pygame and Panda3D. These libraries provide tools for creating 2D and 3D games, handling graphics, and more.
5. **Desktop Applications**: Python can be used to build desktop applications using libraries like Tkinter, PyQt, and wxPython. These libraries provide tools for creating graphical user interfaces (GUIs), handling user input, and more.
6. **Scripting and Automation**: Python is commonly used for scripting and automation tasks due to its simplicity and readability. It can be used to automate repetitive tasks, manage files and directories, and more.
7. **Web Scraping and Crawling**: Python is widely used for web scraping and crawling using libraries like BeautifulSoup and Scrapy. These libraries provide tools for extracting data from websites, parsing HTML and XML, and more.
8. **Education and Research**: Python is commonly used in education and research due to its simplicity and readability. Many universities and research institutions use Python for teaching programming and conducting research in various fields.
9. **Community and Ecosystem**: Python has a large and active community, which contributes to its ecosystem. There are many third-party libraries and frameworks available for various purposes, making Python a versatile language for many applications.
10. **Cross-Platform**: Python is a cross-platform language, which means that Python code can run on different operating systems without modification. This makes it easy to develop and deploy Python applications on different platforms.

Python Online Quiz

To achieve a solid understanding of Python, it’s very important to engage with Python quizzes and MCQs. These quizzes can enhance your ability to solve similar questions and improve your problem-solving skills.

Here are some quiz articles related to Python Tutorial:

- Python MCQs
- Python Quizzes
  - Python Sets Quiz
  - Python List Quiz
  - Python String Quiz
  - Python Tuple Quiz
  - Python Dictionary Quiz

Here's a complete roadmap for you to become a developer: **Learn DSA -> Master Frontend/Backend/Full Stack -> Build Projects -> Keep Applying to Jobs**