# INTRODUCTION TO THE PYTHON PROGRAMMING LANGUAGE

**Rizayeva Bahoroy Jahongir qizi**
Karshi state university, applied mathematics department, 2nd year student
rizayevabahoroy@gmail.com

**Annotation:** This article covers the basic concepts of the Python programming language in a simple and understandable way. The article explains the capabilities, syntax, and practical uses of the Python language for new users with consistent and clear examples. The article also provides information about Python's "pythonic" - that is, its unique and recommended writing style. The material is intended for students, students, and independent learners who are just starting out in programming.

**Keywords:** Python, Arithmetic Operations in Python, NumPy**,** Ins, talling Python, TensorFlow**,** Math Library, math.sqrt(x), math.pow(x,y), import, sympy library, Web Development, Exponents**.**

## ВВЕДЕНИЕ В ЯЗЫК ПРОГРАММИРОВАНИЯ PYTHON

**Аннотация:** В этой статье простым и понятным языком рассматриваются основные концепции языка программирования Python. В статье объясняются возможности, синтаксис и практическое использование языка Python для новых пользователей с последовательными и понятными примерами. В статье также приводится информация о "питоновском" Python - то есть его уникальном и рекомендуемом стиле письма. Материал предназначен для студентов, учащихся и самостоятельных учащихся, которые только начинают изучать программирование.

**Ключевые слова:** Python, арифметические операции в Python, NumPy, Ins, talling Python, TensorFlow, математическая библиотека, math.sqrt(x), math.pow(x,y), импорт, библиотека sympy, веб-разработка, экспоненты.

## PYUTON DASTURLASH TILIGA KIRISH

**Annotatsiya:** Ushbu maqola Python dasturlash tilining asosiy tushunchalarini sodda va tushunarli tarzda yoritadi. Maqolada yangi foydalanuvchilar uchun Python tilining imkoniyatlari, sintaksisi va amaliy qo‘llanilishi izchil va aniq misollar bilan tushuntiriladi. Maqolada Pythonning “pythonic” – ya’ni o‘ziga xos va tavsiya etilgan yozish uslubi haqida ham ma’lumot berilgan. Material talabalar, talabalar va dasturlashni endigina boshlayotgan mustaqil o‘quvchilar uchun mo‘ljallangan.

**Kalit so‘zlar:** Python, Pythonda arifmetik amallar, NumPy, Ins, talling Python, TensorFlow, Math Library, math.sqrt(x), math.pow(x,y), import, simpy kutubxonasi, Web Development, Exponents.

Python is a powerful and versatile programming language with a simple syntax. Created by Guido van Rossum in 1991, it is widely used today in artificial intelligence, web development, data analysis, and automation. One of Python’s greatest advantages is its ease of learning and its extensive library support.

Python’s syntax is very close to human language, making it easy for beginners to grasp. Additionally, its readability speeds up the development process.

Python offers a vast collection of libraries for various domains:

**NumPy, Pandas** — for data analysis

**TensorFlow, PyTorch** — for artificial intelligence and machine learning

**Django, Flask** — for web development

**Pygame** — for game development

**OpenCV** — for image processing

Artificial Intelligence and Data Analysis: Python is the leading language for machine learning, deep learning, and data analysis. Libraries like TensorFlow and Scikit-learn enable the development of AI models.

Web Development: Frameworks such as Django and Flask make Python efficient for building dynamic and secure web applications.

Automation and Scripting: Python allows for the automation of system processes and routine tasks such as file management, data collection, and email automation.

**Guide for Beginners**

**1. Installing Python:** Python can be downloaded from the official website (**https://www.python.org/**) and installed on a computer. Additionally, tools like Jupyter Notebook and VS Code make coding more convenient.

**2. Writing Your First Program**

A simple way to start programming in Python is with the following code:

```
print("Hello, World!")
```

This code outputs "Hello, World!" to the screen.

**3. Basic Concepts**

**3.1. What is a Variable?**

A variable is a named container that temporarily stores data. It can hold numbers, text, lists, or other types of data. In Python, declaring a variable does not require a specific keyword—just assign a value to a name:

```
name = "Ali"
age = 16
```

**3.2. Rules for Naming Variables**

Variable names should be chosen according to the following guidelines:

Must start with a letter or an underscore (_)

Can contain only letters, numbers, and underscores

Are case-sensitive (e.g., name and Name are different)

Cannot be a Python keyword (e.g., if, else, for)

Correct examples:

```
name = "Laylo"
phone_number = "998901234567"
```

Incorrect examples:

```
2name = "Ali"  # Cannot start with a number
for = 5        # Cannot use a keyword
```

**3.3. Basic Data Types**

The most common data types in Python include:

```
name = "Shahzoda"  # str
age = 20           # int
grade = 4.5        # float
is_student = True  # bool
```

**3.4. Using the type() Function to Check Data Type**

```
print(type(name))  # <class ‘str’>
print(type(age))   # <class ‘int’>
```

**3.5. Assigning Values to Multiple Variables**

```
x, y, z = 5, 10, 15
print(x, y, z)
```

**3.6. Updating Variable Values**

A variable's value can be updated at any time:

```
age = 17
age = age + 1
print(age)  # 18
```

Or using a shorter syntax:

```
age += 1
```

**4.Arithmetic Operations in Python :** Python supports basic arithmetic operations, making it easy to perform mathematical calculations. The main arithmetic operators include:

**4.1. Basic Arithmetic Operators**

| Operator | Description | Example | Output |
|---|---|---|---|
| + | Addition | 5 + 3 | 8 |
| - | Subtraction | 10 - 4 | 6 |
| * | Multiplication | 6 * 7 | 42 |
| / | Division | 15 / 3 | 5.0 |
| // | Floor Division | 17 // 4 | 4 |
| % | Modulus (Remainder) | 17 % 4 | 1 |
| ** | Exponentiation | 2 ** 3 | 8 |

**4.2. Using Arithmetic Operators in Python**

```
# Addition
sum_result = 10 + 5
print("Sum:", sum_result)                    # Output: 15
# Subtraction
diff_result = 20 - 8
print("Difference:", diff_result)            # Output: 12
# Multiplication
product_result = 4 * 7
print("Product:", product_result)            # Output: 28
# Division
div_result = 16 / 4
print("Division:", div_result)               # Output: 4.0
# Floor Division
floor_div_result = 17 // 4
print("Floor Division:", floor_div_result)   # Output: 4
# Modulus (Remainder)
mod_result = 17 % 4
print("Remainder:", mod_result)              # Output: 1
# Exponentiation
exp_result = 2 ** 3
print("Exponentiation:", exp_result)         # Output: 8
```

**3. Order of Operations (PEMDAS Rule)**

Python follows the standard mathematical precedence:

**Parentheses ()**

**Exponents ****

**Multiplication *, Division /, Floor Division //, Modulus %** (left to right)

**Addition +, Subtraction -** (left to right)

Example:

```
result = 5 + 2 * 3 ** 2 - (8 // 4)
print(result)  # Output: 18
```

This follows:

3 ** 2 → 9

2 * 9 → 18

8 // 4 → 2

5 + 18 - 2 → 18

Using parentheses helps clarify calculations:

```
result = (5 + 2) * (3 ** 2) - (8 // 4)
print(result)    # Output: 61
```

**5.Working with Math Libraries in Python**

Python provides built-in and third-party libraries to handle advanced mathematical operations efficiently. The most commonly used libraries include math, numpy, and sympy.

**5.1. The math Module**

The math module in Python provides various mathematical functions such as trigonometry, logarithms, and advanced arithmetic.

**Importing the math Module**

```
import math
```

**Common Functions in the math Module**

| Function | Description | Example | Output |
|---|---|---|---|
| math.sqrt(x) | Square root of x | math.sqrt(25) | 5.0 |
| math.pow(x,y) | x raised to the power y | math.pow(2, 3) | 8.0 |
| math.factorial(x) | Factorial of x | math.factorial(5) | 120 |
| math.log(x) | Natural logarithm (base e) | math.log(10) | 2.302 |
| math.log10(x) | Logarithm base 10 | math.log10(100) | 2.0 |
| math.sin(x) | Sine of x (radians) | math.sin(math.pi/2) | 1.0 |
| math.cos(x) | Cosine of x (radians) | math.cos(math.pi) | -1.0 |
| math.pi | Value of π | math.pi | 3.1415 |
| math.e | Value of Euler’s number e | math.e | 2.718 |

Example Usage:

```
import math
num = 16
print("Square root:", math.sqrt(num))          # Output: 4.0
print("Factorial:", math.factorial(5))          # Output: 120
print("Log base 10:", math.log10(100))         # Output: 2.0
```

**5.2. The numpy Library**

NumPy is a powerful library for numerical computations, commonly used in scientific and engineering applications.

**Installing NumPy:** pip install numpy

**Using NumPy**

```
import numpy as np
arr = np.array([1, 2, 3, 4])
print("Mean:", np.mean(arr))                    # Output: 2.5
print("Standard Deviation:", np.std(arr))       # Output: 1.118
```

**5.3. The sympy Library (For Symbolic Mathematics)**

SymPy is used for symbolic mathematics, including algebraic manipulation and solving equations.

**Installing SymPy:** pip install sympy

**Using SymPy**

```
from sympy import symbols, solve
x = symbols('x')
expr = x**2 - 4
solutions = solve(expr, x)
print("Solutions:", solutions)    # Output: [-2, 2]
```

In conclusion, the Python programming language has become one of the most popular programming languages today, thanks to its simplicity, readable syntax, and powerful capabilities. This article covers the basics of Python in a simple and understandable way, providing essential knowledge, especially for novice programmers. Python's "pythonic" approach ensures that the code is not only functional, but also understandable and aesthetically clean. Through this article, students will increase their interest in programming and take a step closer to mastering Python programming.

**References**

1. Lutz, M. (2013). Learning Python (5th ed.). O'Reilly Media.

2. Sweigart, A. (2015). Automate the Boring Stuff with Python. No Starch Press.

3. Zelle, J. M. (2016). Python Programming: An Introduction to Computer Science (3rd ed.). Franklin, Beedle & Associates.

4. Martelli, A., Ravenscroft, A., & Ascher, D. (2005). Python Cookbook (2nd ed.). O'Reilly Media.

5. Van Rossum, G., & Drake, F. L. (2009). The Python Language Reference Manual. Network Theory Ltd.

6.Python Software Foundation. (2024). The Python Tutorial. https://docs.python.org/3/tutorial

7. W3Schools. Python Tutorial. https://www.w3schools.com/python

8. Real Python. Python Basics and Guides. https://realpython.com.

## KORXONA OMBOR HISOBINI AVTOMATLASHTIRISH – ZAMONAVIY YONDASHUVLAR VA AMALIY YECHIMLAR

**Safarov Shag'zod Zokir o'g'li**

Jizzax shahridagi Qozon (Volgabo'yi) federal universiteti filiali 4-kurs talabasi

safarovshzs@gmail.com