Introduction to Python for Beginners

A guide to the basics of Python programming language

# Overview

* Python is a high-level, interpreted programming language known for its simplicity and readability.
* It supports multiple programming paradigms, including procedural, object-oriented, and functional programming.
* Python is widely used for web development, data science, automation, scripting, and more.

# Python History

* Python was created by Guido van Rossum and first released in 1991.
* It was developed as a successor to the ABC programming language and was designed to be easy to read and use.
* Python’s name was inspired by the British comedy group Monty Python, and not the snake as many might think.

# Key Releases

* Python 1.0 (1994): The first official version of Python, which introduced basic data types like strings, lists, and dictionaries.
* Python 2.0 (2000): Added features like list comprehensions and garbage collection. Python 2 was widely adopted and used for many years.
* Python 3.0 (2008): A major revision of the language that is not backward compatible with Python 2. Introduced features like print as a function, better Unicode support, and more.

# Difference Between Python 2 and Python 3

* Print Statement: In Python 2, print is a statement, whereas in Python 3, print() is a function.
* Division: In Python 2, dividing two integers performs integer division. In Python 3, it performs true division.
* Unicode: Python 3 has better support for Unicode, which is essential for working with text in multiple languages.

# Python Installation

* Installation via python.org: Visit python.org and download the installer for your operating system. Run the installer and follow the prompts (make sure to check "Add Python to PATH" during installation). After installation, you can verify it by running python --version in your command prompt or terminal.
* Installation via Anaconda: Visit Anaconda.com/download and download the Anaconda installer for your operating system. Run the installer and follow the instructions. Anaconda comes with Python and many scientific libraries pre-installed.

# Environment Setup

* IDE: You can use an Integrated Development Environment (IDE) like PyCharm, VS Code, or Jupyter Notebook.
* Virtual Environments: It's a good practice to use virtual environments to manage dependencies for different projects.

# Syntax

* Python syntax is clean and straightforward. It uses indentation to define blocks of code rather than braces or keywords.
* Comments are lines that Python will ignore. They start with a #.
* Indentation is used to define code blocks (e.g., the body of a function or loop).

# Command Line Arguments

* Python scripts can accept command line arguments using the sys module.
* sys.argv is a list of command-line arguments.

# Basic Data Types and Variables

* Integer: Whole numbers (e.g., 5, -3).
* Float: Numbers with a decimal point (e.g., 3.14, -0.5).
* String: Text (e.g., "Hello").
* Boolean: True or False values (e.g., True, False).

# Indexing and Slicing

* Indexing and slicing allow you to access parts of a string or list.
* Indexing allows you to access specific elements, while slicing lets you access a range of elements.

# Type Casting

* Type casting is converting a variable from one type to another.
* You can use functions like str(), int(), float(), and bool() to perform type casting.

# Operators

* Arithmetic Operators: +, -, \*, /, % (modulus), \*\* (exponentiation).
* Comparison Operators: ==, !=, >, =, <=.
* Logical Operators: and, or, not.
* Identity Operators: is, is not.
* Membership Operators: in, not in.

# Control Structures

* If-Else: Execute code based on conditions.
* For Loop: Iterate over a sequence (like a list).
* While Loop: Repeat as long as a condition is true.

# Functions and Modules

* Functions: A function is a block of code that performs a specific task.
* Modules: A module is a file containing Python code that can be imported into another script.

# Exercise questions

* Write a Python program to swap the values of two variables without using a third variable.
* Create a program that takes a string input from the user and prints out the string in reverse order.
* Write a Python script that converts temperature from Fahrenheit to Celsius. The formula for conversion is: Celsius = (Fahrenheit - 32) \* 5/9.
* Write a program that asks the user for their name and age, and then prints out how many years until they turn 100 years old.
* Create a program that takes an integer input from the user and prints whether it is an odd or even number.
* Write a Python program that takes two numbers as input and prints their sum, difference, product, and quotient.
* Create a program that asks the user to input two numbers and then checks if the first number is divisible by the second number. Print an appropriate message based on the result.
* Write a program that accepts three numbers from the user and prints the largest of the three.
* Create a Python script that calculates the area of a triangle using the formula: Area = sqrt(s \* (s - a) \* (s - b) \* (s - c)), where a, b, and c are the lengths of the sides of the triangle and s is the semi-perimeter. Use the formula: s = (a + b + c) / 2.
* Write a Python program that uses bitwise operators to swap two numbers.
* Write a Python program that prints all the numbers from 1 to 100. For multiples of 3, print "Fizz" instead of the number, and for multiples of 5, print "Buzz". For numbers that are multiples of both 3 and 5, print "FizzBuzz".
* Create a Python program that calculates the factorial of a given number using a while loop.
* Write a Python script that asks the user for a number and then prints the multiplication table for that number up to 10.
* Create a program that takes a list of numbers and prints only the even numbers using a for loop.
* Write a program that prompts the user to enter a word and then determines if the word is a palindrome (a word that reads the same backward as forward).
* Write a Python function that takes two numbers as arguments and returns their greatest common divisor (GCD).
* Create a Python function that accepts a string and returns the number of vowels in the string.
* Write a Python function that accepts a list of numbers and returns the second largest number in the list.
* Create a module that contains a function to check whether a number is prime. Write a script that imports this module and uses the function to check if a given number is prime.
* Write a Python function that takes a list of integers as input and returns a new list containing only the unique elements from the original list (i.e., remove duplicates).