

Introduction to Python

About Python

- Python is a general-purpose, high-level programming language.
 - It was developed by **Guido van Rossum** in 1989 at the National Research Institute in the Netherlands.
 - Python was officially released to the public on **February 20, 1991**.
 - Python is highly recommended as a first programming language for beginners due to its simplicity.
-

Examples

Example 1: Print "Hello World"

Java:

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
    }  
}
```

C:

```
#include<stdio.h>  
void main() {  
    printf("Hello World");  
}
```

Python:

```
print("Hello World")
```

Example 2: Sum of Two Numbers

Java:

```
public class Add {  
    public static void main(String[] args) {  
        int a, b;  
        a = 10;  
        b = 20;  
        System.out.println("The Sum: " + (a + b));  
    }  
}
```

C:

```
#include <stdio.h>  
  
void main() {  
    int a, b;  
    a = 10;  
    b = 20;  
    printf("The Sum: %d", (a + b));  
}
```

Python:

```
a = 10  
b = 20  
print("The Sum:", (a + b))
```

Origin of the Name "Python"

The name **Python** was inspired by the British comedy TV show *"Monty Python's Flying Circus"* broadcasted by the BBC from 1969 to 1974.

Language Influences

Guido van Rossum designed Python by borrowing features from various programming languages:

- **Functional programming:** C
 - **Object-oriented programming:** C++
 - **Scripting features:** Perl, Shell Script
 - **Modular programming:** Modula-3
 - Syntax is mostly derived from **C** and **ABC languages**.
-

Applications of Python

Python can be used in a wide variety of areas:

- 1 Desktop Applications
2. Web Applications
3. Database Applications
4. Network Programming
5. Game Development
6. Data Analysis
7. Machine Learning
8. Artificial Intelligence
9. Internet of Things (IoT)

Note:

Python is used by companies like **Google, Microsoft, IBM, Yahoo.**

It is also used in YouTube, NASA, and New York Stock Exchange.

Features of Python

- 1 **Simple and Easy to Learn**
 - Python code reads like English.
 - Minimal syntax, around 30+ keywords.
 - Fewer lines of code for the same logic compared to other languages.
- 1 **Free and Open Source**
 - No license needed to use Python.
 - Source code is open for modification (e.g., Jython for Java integration).
- 1 **High-Level Language**
 - Programmer-friendly; no need to manage memory or security manually.
- 1 **Platform Independent**
 - Write once, run anywhere (PVM handles platform translation).
- 1 **Portable**
 - Code produces the same result across platforms.
- 1 **Dynamically Typed**
 - No need to declare variable types explicitly.
 - Type is inferred at runtime.
- 1 **Supports Both Procedural and Object-Oriented Programming**
- 1 **Interpreted Language**
 - Compilation handled by Python Interpreter.
 - Errors raised at runtime if any.
- 1 **Extensible**
 - Can integrate code from other languages for performance or legacy reuse.
- 1 **Embeddable**
 - Python code can be embedded within other languages.
- 1 **Extensive Library Support**

- Large standard library with built-in functions and modules.

Limitations of Python

- 1 Performance
 - Slower than compiled languages due to interpretation.
 - 1 Not Commonly Used for Mobile Development
-

Flavors of Python

Flavor	Description
CPython	Standard implementation using C
Jython/JPython	For Java applications (runs on JVM)
IronPython	For C#/.NET applications
PyPy	JIT compiler support for faster performance
RubyPython	Integrates Python with Ruby
AnacondaPython	Optimized for data processing and analytics

Python Versions

Version	Release Date
Python 1.0	January 1994
Python 2.0	October 2000
Python 3.0	December 2008

Note:

Python 3 does **not** provide backward compatibility with Python 2.

Python 2 programs may not run on Python 3 without modifications.

Current Common Versions

- Python 3.13.6
 - Python 3.6.1
 - Python 2.7.13
-

76. Python Program to Differentiate Between type() and isinstance()

```
class Polygon:
    def sides_no(self):
        pass

class Triangle(Polygon):
    def area(self):
        pass

obj_polygon = Polygon()
obj_triangle = Triangle()

print(type(obj_triangle) == Triangle)    # true
print(type(obj_triangle) == Polygon)     # false

print(isinstance(obj_polygon, Polygon))  # true
print(isinstance(obj_triangle, Polygon)) # true
```

we see that type() cannot distinguish whether an instance of a class is somehow related to the base class.

- In our case, although obj_triangle is an instance of child class Triangle,
- it is inherited from the base class Polygon. If you want to relate the object of a child class with the base class,
- you can achieve this with isinstance().

80. Python Program to Return Multiple Values From a Function

```
# Example 1: Return values using comma
def name():
    return "John","Armin"

# print the tuple with the returned values
print(name())

# get the individual items
name_1, name_2 = name()
print(name_1, name_2)
```

88. Python Program to Compute the Power of a Number

```
# Calculate the power of a number using pow() function
base = 3
exponent = 4
```

```
result = pow(base, exponent)
print("Answer = " + str(result))
```

```
# Calculate power of a number using a for loop
base = 3
exponent = 4
```

```
result = 1
```

```
for exponent in range(exponent, 0, -1):
    result *= base
```

```
print("Answer = " + str(result))
```

```
# Calculate power of a number using a while loop
base = 3
exponent = 4
```

```
result = 1
```

```
while exponent != 0:
    result *= base
    exponent-=1
```

```
print("Answer = " + str(result))
```

```
base = 3
exponent = 4
exponent+=1
result =1
```

```
while exponent:= exponent-1:
    result *= base
result
```

```
base = 3
exponent = 4
result =1
```

```
while exponent:
    result *= base
    exponent-=1
```

```
result
```

Evaluation Warning: The document was created with Spire.Doc for Python.

89. Python Program to Count the Number of Digits Present In a Number

```
num = 1234567

def count_digits(n):
    count = 0
    while n > 0:
        n = n // 10
        count += 1
    return count

digit_count = count_digits(abs(num))
print(digit_count)
```

Evaluation Warning: The document was created with Spire.Doc for Python.