

# PYTHON

→ Mrs. SONIYA P. SHARMA.

# Introduction

- Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL).
- Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

# Python Features

- **Easy-to-learn:** Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read:** Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain:** Python's source code is fairly easy-to-maintain.
- **A broad standard library:** Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode:** Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable:** Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable:** You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases:** Python provides interfaces to all major commercial databases.

# Python Features

- **GUI Programming:** Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable:** Python provides a better structure and support for large programs than shell scripting.
- **Python is Interpreted:** Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive:** You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented:** Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language:** Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

# Experimental Debugging

- One of the most important skills you will acquire is debugging. Although it can be frustrating, debugging is one of the most intellectually rich, challenging, and interesting parts of programming.
- In some ways, debugging is like detective work. You are confronted with clues, and you have to infer the processes and events that led to the results you see.
- Debugging is also like an experimental science. Once you have an idea what is going wrong, you modify your program and try again. If your hypothesis was correct, then you can predict the result of the modification, and you take a step closer to a working program. If your hypothesis was wrong, you have to come up with a new one.
- For some people, programming and debugging are the same thing. That is, programming is the process of gradually debugging a program until it does what you want. The idea is that you should start with a program that does *something* and make small modifications, debugging them as you go, so that you always have a working program.

# Formal & Natural Languages

- **Natural languages** are the languages that people speak, such as English, Spanish, and French. They were not designed by people (although people try to impose some order on them); they evolved naturally.
- **Formal languages** are languages that are designed by people for specific applications. For example, the notation that mathematicians use is a formal language that is particularly good at denoting relationships among numbers and symbols. Chemists use a formal language to represent the chemical structure of molecules. And most importantly:

Programming languages are formal languages that have been designed to express computations.

- Formal languages tend to have strict rules about syntax. For example,  $3+3=6$  is a syntactically correct mathematical statement, but  $3=+6\$$  is not.  $H_2O$  is a syntactically correct chemical name, but  $_2Zz$  is not.
- **Ambiguity** - Natural languages are full of ambiguity, which people deal with by using contextual clues and other information. Formal languages are designed to be nearly or completely unambiguous, which means that any statement has exactly one meaning, regardless of context.
- **Redundancy** - In order to make up for ambiguity and reduce misunderstandings, natural languages employ lots of redundancy. As a result, they are often verbose. Formal languages are less redundant and more concise.
- **Literalness** - Formal languages mean exactly what they say. On the other hand, natural languages are full of idiom and metaphor. If someone says, “The other shoe fell”, there is probably no shoe and nothing falling.

# Difference – Brackets, Braces & Parentheses

	<p><b>Braces ("curly braces")</b></p> <p>Braces are used to group <a href="#">statements</a> and <a href="#">declarations</a>. The contents of a <a href="#">class</a> or <a href="#">interface</a> are enclosed in braces. <a href="#">Method</a> bodies and <a href="#">constructor</a> bodies are enclosed in braces. Braces are used to group the statements in an <a href="#">if statement</a>, a <a href="#">loop</a>, or other <a href="#">control structures</a>.</p>
	<p><b>Brackets ("square brackets")</b></p> <p>Brackets are used to index into an <a href="#">array</a>.</p>
	<p><b>Parentheses</b></p> <p>Parentheses are used for two purposes: (1) to control the order of operations in an expression, and (2) to supply parameters to a <a href="#">constructor</a> or <a href="#">method</a>.</p>

# Variables and Expressions

- Assigning Values to Variables
- Python variables do not need explicit declaration to reserve memory space. The declaration happens automatically when you assign a value to a variable. The equal sign (=) is used to assign values to variables.
- `counter = 100 # An integer assignment`
- `miles = 1000.0 # A floating point`
- `name = "John" # A string`
- Single values assign to several variables. `a=b=c=1`
- Multiple values to multiple variables `a,b,c = 1,2,"priya"`