

**SAVEETHA SCHOOL OF ENGINEERING**

**SAVEETHA INSTITUTE OF MEDICAL AND SCIENCES**

**CHENNAI-602105**

**INSTITUTE OF PLACEMENT AND TRAINING**

**PYTHON COURSE MODULE**

**INTRODUCTION TO PYTHON**

* **Introduction to Python, History and Versions**
* **Python Installation and Environmental Setup**
* **Uses of Python, Characteristics, Advantages, Features and Python Application**
* **Syntax, Scripts and Modules**
* **Naming, Keywords & Variables**
* **Decision Making**

**Statements**

* [Multi-Line Statements](https://pynative.com/python-statements/" \l "h-multi-line-statements)
* [Python Compound Statements](https://pynative.com/python-statements/" \l "h-python-compound-statements)
* [Simple Statements](https://pynative.com/python-statements/" \l "h-simple-statements)
  + [Expression statements](https://pynative.com/python-statements/" \l "h-expression-statements)
  + [The pass statement](https://pynative.com/python-statements/" \l "h-the-pass-statement)
  + [The del statement](https://pynative.com/python-statements/" \l "h-the-del-statement)
  + [The return statement](https://pynative.com/python-statements/" \l "h-the-return-statement)
  + [The import statement](https://pynative.com/python-statements/" \l "h-the-import-statement)
  + [The continue and break statement](https://pynative.com/python-statements/" \l "h-the-continue-and-break-statement)

1. **Comments**

* [What is Comment in Python?](https://pynative.com/python-comments/" \l "h-what-is-comment-in-python)
* [Single-line Comment](https://pynative.com/python-comments/" \l "h-single-line-comment)
* [Multi-Line Comments](https://pynative.com/python-comments/" \l "h-multi-line-comments)
* [Add Sensible Comments](https://pynative.com/python-comments/" \l "h-add-sensible-comments)
* [Inline Comments](https://pynative.com/python-comments/" \l "h-inline-comments)
* [Block Comments](https://pynative.com/python-comments/" \l "h-block-comments)
* [Docstring Comments](https://pynative.com/python-comments/" \l "h-docstring-comments)
* [Commenting Out Code for Testing](https://pynative.com/python-comments/" \l "h-commenting-out-code-for-testing)
* [Using String Literals for Multi-line Comments](https://pynative.com/python-comments/" \l "h-using-string-literals-for-multi-line-comments)
* [Summary](https://pynative.com/python-comments/" \l "h-summary)

1. **Keywords**

* [Get the List of Keywords](https://pynative.com/python-keywords/" \l "h-get-the-list-of-keywrods)
* [Understand Any keyword](https://pynative.com/python-keywords/" \l "h-understand-any-keyword)
* [How to Identify Python Keywords](https://pynative.com/python-keywords/" \l "h-how-to-identify-python-keywords)
* [Keyword Module](https://pynative.com/python-keywords/" \l "h-keyword-module)
* [Types of Keyword](https://pynative.com/python-keywords/" \l "h-types-of-keywords)

1. **Variables**

* [Creating a variable](https://pynative.com/python-variables/" \l "h-creating-a-variable)
* [Changing the value of a variable](https://pynative.com/python-variables/" \l "h-changing-the-value-of-a-variable)
* [Create Number, String, List variables](https://pynative.com/python-variables/" \l "h-create-number-string-list-variables)
* [Get the data type of variable](https://pynative.com/python-variables/" \l "h-get-the-data-type-of-variable)
* [Delete a variable](https://pynative.com/python-variables/" \l "h-delete-a-variable)
* [Variable’s case-sensitive](https://pynative.com/python-variables/" \l "h-variable-s-case-sensitive)
* [Constant](https://pynative.com/python-variables/" \l "h-constant)
* Type conversion in python
* Byte objects vs string in python
* Print single and multiple variable
* [Rules and naming convention for variables and constants](https://pynative.com/python-variables/" \l "h-rules-and-naming-convention-for-variables-and-constants)
* [Multiple assignments](https://pynative.com/python-variables/" \l "h-multiple-assignments)
* [Variable scope](https://pynative.com/python-variables/" \l "h-variable-scope)
* Swap variable
* Private variables
* Global and local variables in python
* [Object/Variable identity and references](https://pynative.com/python-variables/" \l "h-object-variable-identity-and-references)
* Pack and [Unpack a collection into a variable](https://pynative.com/python-variables/" \l "h-unpack-a-collection-into-a-variable)

1. Data Types

* [String](https://pynative.com/python-data-types/" \l "h-str-data-type)
* [Integer](https://pynative.com/python-data-types/" \l "h-int-data-type)
* [Float](https://pynative.com/python-data-types/" \l "h-float-data-type)
* [Complex](https://pynative.com/python-data-types/" \l "h-complex-data-type)
* [List](https://pynative.com/python-data-types/" \l "h-list-data-type)
* [Tuple](https://pynative.com/python-data-types/" \l "h-tuple-data-type)
* [Dict](https://pynative.com/python-data-types/" \l "h-dict-data-type)
* [Set](https://pynative.com/python-data-types/" \l "h-set-data-type)
* [Bool](https://pynative.com/python-data-types/" \l "h-bool-data-type)
* [Bytes](https://pynative.com/python-data-types/" \l "h-bytes-data-type)
* [Range](https://pynative.com/python-data-types/" \l "h-range-data-type)

1. Operators

* [Arithmetic operator](https://pynative.com/python-operators/" \l "h-arithmetic-operator)
* [Relational (comparison) operators](https://pynative.com/python-operators/" \l "h-relational-comparison-operators)
* [Assignment operators](https://pynative.com/python-operators/" \l "h-assignment-operators)
* [Logical operators](https://pynative.com/python-operators/" \l "h-logical-operators)
* [Membership operators](https://pynative.com/python-operators/" \l "h-membership-operators)
* [Identity operators](https://pynative.com/python-operators/" \l "h-identity-operators)
* [Bitwise Operators](https://pynative.com/python-operators/" \l "h-bitwise-operators)
* [Python Operators Precedence](https://pynative.com/python-operators/" \l "h-python-operators-precedence)

1. Flow Controls

* [Control Flow Statements](https://pynative.com/python-control-flow-statements/" \l "h-control-flow-statements)
* [If statement](https://pynative.com/python-control-flow-statements/" \l "h-if-statement-in-python)
* [for loop](https://pynative.com/python-control-flow-statements/" \l "h-for-loop-in-python)
* [While loop](https://pynative.com/python-control-flow-statements/" \l "h-while-loop-in-python)
* [Break Statement](https://pynative.com/python-control-flow-statements/" \l "h-break-statement-in-python)
* [Continue and Pass statement](https://pynative.com/python-control-flow-statements/" \l "h-continue-statement-in-python)
* Pattern Printing
* Iterations

1. List

* [Iterating a List](https://pynative.com/python-lists/" \l "h-iterating-a-list)
* List Operations and Methods
* [Built-in functions with List](https://pynative.com/python-lists/" \l "h-python-built-in-functions-with-list)
* [List Comprehension](https://pynative.com/python-lists/" \l "h-list-comprehension)
* List1D vs List 2D

1. String

* String Introduction
* Accessing
* Slicing
* Mutability
* Formatted Strings
* String Operations
* More inbuilt methods
* Problems

10.Tuples, Dictionaries and Sets

* Creating
* Accessing
* Updating
* Properties
* Functions
* Built in Methods

11. Functions

* [Types of Functions](https://pynative.com/python-functions/" \l "h-types-of-functions)
* [Creating a Function](https://pynative.com/python-functions/" \l "h-creating-a-function)
* [Calling a function](https://pynative.com/python-functions/" \l "h-calling-a-function)
* [Docstrings](https://pynative.com/python-functions/" \l "h-docstrings)
* Function Decorators
* [Return multiple](https://pynative.com/python-functions/" \l "h-return-value-from-a-function) values
* Partial Functions in Python
* \*args and \*\*kwargs
* [The pass Statement](https://pynative.com/python-functions/" \l "h-the-pass-statement)
* [How does Function work in Python?](https://pynative.com/python-functions/" \l "h-how-does-function-work-in-python)
* [Scope and Lifetime of Variables](https://pynative.com/python-functions/" \l "h-scope-and-lifetime-of-variables)
* [Python Function Arguments](https://pynative.com/python-functions/" \l "h-python-function-arguments)
* [Recursive Function](https://pynative.com/python-functions/" \l "h-recursive-function)
* [Python Anonymous/Lambda Function](https://pynative.com/python-functions/" \l "h-python-anonymous-lambda-function)

12. Modules

* [Types of modules](https://pynative.com/python-modules/" \l "h-types-of-modules)
* [import modules](https://pynative.com/python-modules/" \l "h-how-to-import-modules)
* [Create Module](https://pynative.com/python-modules/" \l "h-create-module)
* [Python Module Search Path](https://pynative.com/python-modules/" \l "h-python-module-search-path)
* Python Build-in Modules
* [Reloading a module](https://pynative.com/python-modules/" \l "h-reloading-a-module)
* [The dir() function](https://pynative.com/python-modules/" \l "h-the-dir-function)
* Packages
* Understanding \_\_init\_\_.py

**13.Input-Output**

Output Operation

READING INPUT FROM THE KEYBOARD (INPUT OPERATION)

Files in Python

open a file

read file

write file

close file

create file

delete file

**14.Exception Handling**

Types of Errors

Syntax Errors

Semantic Errors

Run Time Errors

Logical Errors

Exceptions

Handling Exceptions

Raising Exceptions

Exception Chaining

User-defined Exceptions

Defining Clean-up Actions

Predefined Clean-up Actions

Raising and Handling Multiple Unrelated Exceptions

Enriching Exceptions with Notes

**15.OOPS Concept**

Advantages of OOPS in Python

Difference Between Object Oriented & Procedural Oriented Programming

Class

**Class Definition Syntax**

The \_\_init\_\_ method

Class Attribute

Instance Attribute

Creating an empty Class in Python

Objects

Creating an object

The self

Instance Methods

Creating a class and object with class and instance attributes

Creating Class and objects with methods

Inheritance

Types of Inheritance

Single Inheritance

Multilevel Inheritance

Hierarchical Inheritance

Multiple Inheritance

Super()

Polymorphism

In-built polymorphic functions

Polymorphism with Class Methods

Polymorphism with Inheritance

Encapsulation

Getters and setters

Data Abstraction

Key Points of Abstract classes

Syntax of Abstract Class in Python

Access Modifiers

Default

Private

Public

Protected

Constructors in Python

Destructors in Python

Garbage collection

**16.Regular Expression**

Regular Expression in python

Special Characters

Rules of Regular Expressions

Matching and Searching

Modifiers

Patterns

**17.CGI**

Architecture

CGI environment variable

GET and POST methods

Cookies

File upload

**18. DATABASE**

Connections

Executing queries

Transactions

Handling error

**19.MULTITHREADING**

Thread

Starting a thread

Threading module

Synchronizing threads

Multithreaded Priority Queue

**20.GUI PROGRAMMING**

Tkinter Programming

Tkinter widgets

Mail Communication in python scripts