



Advanced Certification Course in **Python**

- Practical Training
- Training From Expert Trainer
- Interview Preparation
- Complete Placement Assistance

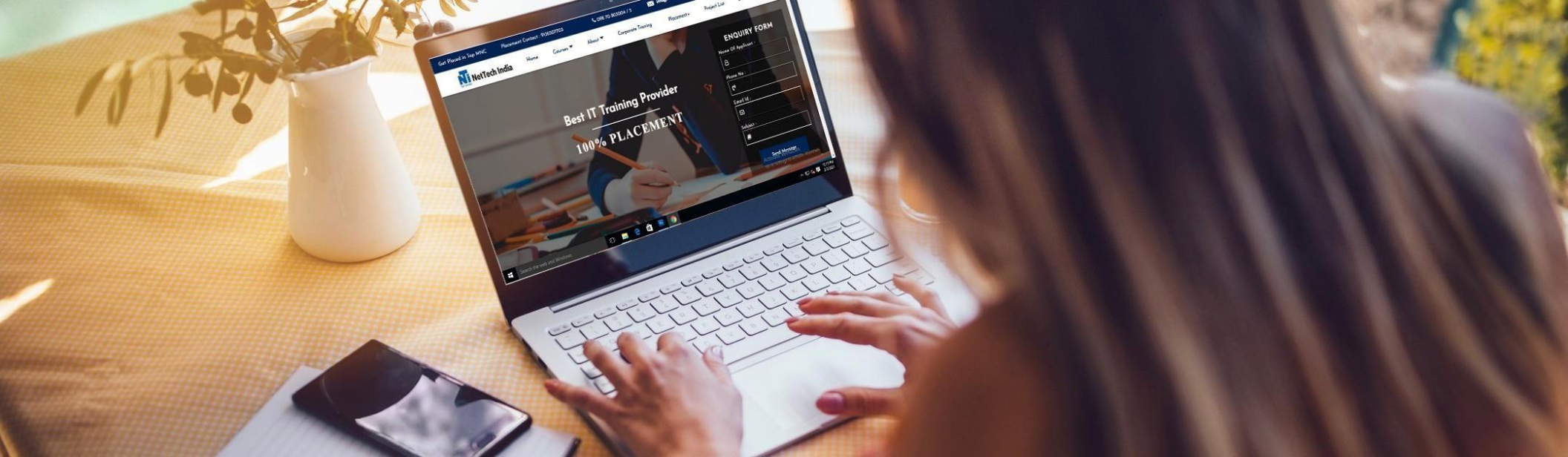


CERTIFICATIONS OPTIONS AVAILABLE



JAINxJAIN
THE LEARNING HUB DEEMED-TO-BE UNIVERSITY





ABOUT US

Infobyte Computers offers a high-quality learning experience in the field of IT training ~~to students~~ on brand new technologies and train them to deliver the desired results with commercially relevant and re-organized technical skills.

The probability of achieving your dream job will keep on increasing day by day once you complete a course in **Infobyte Computers**. We also focus on improving soft skills in terms of communication, leadership, teamwork, external appearance, and attitude which helps everyone to be professional in all the aspects of their career.



25%

Theory



75%

Practicals

ABOUT PYTHON

Python is general-purpose programming language having applications in a wide the field range of web development, science, and mathematical computing to a desktop graphical user interface.

Python is easy to comprehend and the syntax feels natural. Even if you are new to this language, you can easily get a grip of it. Python comprises a large community base. There are several active forums that can let you handle the challenges if you are struck. It helps you to write in a language that has a greater functionality with few lines of code.



BENEFITS OF PYTHON

- Career Growth - Higher Pay & Position
- Encourages professional Development
- Enriches self-image and Reputation
- Enhances professional Credibility.
- Abundant Job Opportunities
- Used In Many Industries
- Global Recognition
- Secure and Flexible
- 50+ Case Studies
- 50+ Projects



Python

1. Introduction to Python

- History of Python
- Why to learn python
- How is Python Different?
- Installing Python

2. Python Interpreter

- Using the interpreter
- Integrated Development Environments (IDE) How to run Python programs?

3. Basics of Python

- Variable
- Keywords
- Statements & Comments
- Indentation
- Data types

- Static Typing vs Dynamic Typing
- Input and output
- Operators Arithmetic operator Relational Operator Assignment Operator
- Logical operator Bitwise operator Membership Operator
- Identity Operator

4. Control Flow

- If statement
 - If - else
 - If – elif -else
 - Nested if-else
 - while loop
 - for – in loop
 - Nested for loop
 - Nester while loop
 - Loop with else
 - Pass statement
 - Break and continue
-

5. Functions

- Basics Defining function
- function call Return statement
- Function with parameter and without parameter
- Function parameters Call by value or call by reference local and global variable
- Recursion, Anonymous (lambda) function
- User define functions
- Examples

6. Modules

- Defining module
- How to create a module
- Importing module
- Dir()
- Module search path
- Reloading a module
- Sys module
- Os module
- namespace

7. Package

- Defining package
- How to create the package
- Importing package
- Installing third party packages

8. Numeric Types

- Numeric type basics
- Hexadecimal, Octal, and Binary Notation Complex Numbers
- Typecasting Numeric Functions
- Random number generation(Using Random Modules)

String

- Defining a string
- Different ways to create string Accessing elements of the string Escape sequence
- Raw string String methods String formatting Expressions

10. List

- Defining a list
- Creating list
- Accessing list elements of list
- Deleting list
- List methods
- Functions used with list
- List comprehension
- Implementation of stack and queue using list
- Use of Zip ()
- Matrix operations using list

11. Tuple

- Defining a tuple
- Creating a tuple
- Accessing elements of the tuple
- What is Immutability
- List vs tuples
- Tuple Methods Functions used with tuple
- Advantage of Tuple

12. Dictionary

- Defining a dictionary
- Creating a dictionary
- Accessing elements of the dictionary
- Deleting a dictionary
- Dictionary methods
- Dictionary Comprehension

13. Set

- Defining a set
- Creating set
- Set operations
- Set methods
- Set comprehension

14. Files

- Defining a file
- Types of file operations
- Opening a File
- Closing file
- File modes

- File attributes
- Writing to file
- Reading from file
- Appending to file
- File positions
- Binary file
- Pickle module

15. Exception Handling

- Defining an exception?
 - Default exception handler
 - Exception handling techniques
 - a. Detecting Exception (try)
 - b. Catching exceptions (catch)
 - c. Catching multiple exceptions
 - d. Raising exception (raise) Finally block
 - User-defined exceptions
-

16. Object-Oriented Programming

OOPS concepts Defining

Class Creating object

- Method vs function Calling methods
- Instance attribute vs class attribute
- Instance method vs class method
- Private attribute and method Static Method
- Method Overloading Constructor
- Method Overriding Constructor
- List of objects Inheritance
- Examples

17. Multi-Threading

- Process-based multitasking
- Thread based multitasking
- Creating a Thread without using class
- Creating thread using class
- Sleep() method
- Join() method Getting and setting the name of the Thread
Logging module
- Synchronization
- Lock concept
- Object-Oriented

- Inter thread communication
- Is_Alive() method
- Active_count() method
- Enumerate() method
- Current_thread() method
- Daemon Thread

18. GUI Programming with Tkinter

- Introduction to Tkinter
- Creating a window Tkinter widgets Label
- Button Entry Messagebox List
- Radio Button CheckButton Creating Frame
- Creating Menu Assignments on Tkinter
- Examples

19. Event Handling

- Defining an event
 - Bind() method
 - Mouse events
 - Keyboard events
 - Examples
-

20. Data Base Programming

- Introduction to MySQL.connector module, Connecting to the database by using MySQL, Creating a table by MySQL
- Performing SQL operations, Introduction to mysql, Installing mysql, Creating database using mysql
- Connecting MySQL database from python, Creating a table, Performing
- SQL operations
- Examples

21. Conversion of Python script to executable file

- Defining an executable file , Deploying the application

22. LIVE PROJECTS

- Create GUI and store data in the Database. (5-day session) Create a server-client program. (using TCP)

And Many More...

WHO CAN LEARN ?

- Anyone who wants to build a career in Data Science
- Anyone who wish to gain knowledge about Programming Students
- who are currently in college or university

CAREER OPPORTUNITIES

- Software Engineer
- Python Developer
- Research Analyst
- Software Developer
- Product Manager
- Data Scientist
- Data Journalist
- Python Trainer
- Data Architect

And Many More...



OUR RECRUITERS



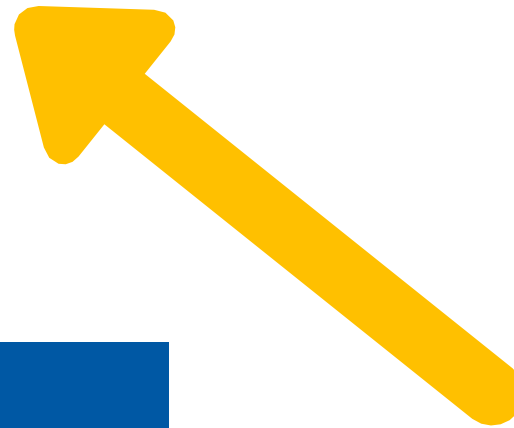
and Many

PROCESS FOR SUCCESS

GET PLACED

GET TRAINED

ENROLL



FACILITIES OFFERED

- Practical Training on Live Projects
- Complete Placement Assistance
- Interview Preparation
- Global Certification
- Fully functional labs
- Online / Offline Training
- Study Materials
- Expert Level Industry Recognized Training

