**Dear Dr. Loganathan:**

**Training Description**

This workshop contains a set of basic to intermediate Python lessons and hands on labs for writing programs. By completing this training, the audience can create academic and commercial projects using Python.

Audience: For any student or professional interested in learning the Python from scratch to professional level.

Trainer: 8+ years of corporate experience with companies like CISCO, Juniper and Aruba. NIT Allahabad Computer Science Graduate

Sincerely,

Awantik Das

+91 7022251144

awantikdas@gmail.com

PROFESSIONAL TRAINING PROPOSED

Course Content

* [1. Whetting Your Appetite](https://docs.python.org/2/tutorial/appetite.html)
* [2. Using the Python Interpreter](https://docs.python.org/2/tutorial/interpreter.html)
  + [2.1. Invoking the Interpreter](https://docs.python.org/2/tutorial/interpreter.html#invoking-the-interpreter)
    - [2.1.1. Argument Passing](https://docs.python.org/2/tutorial/interpreter.html#argument-passing)
    - [2.1.2. Interactive Mode](https://docs.python.org/2/tutorial/interpreter.html#interactive-mode)
  + [2.2. The Interpreter and Its Environment](https://docs.python.org/2/tutorial/interpreter.html#the-interpreter-and-its-environment)
    - [2.2.1. Source Code Encoding](https://docs.python.org/2/tutorial/interpreter.html#source-code-encoding)
* [3. An Informal Introduction to Python](https://docs.python.org/2/tutorial/introduction.html)
  + [3.1. Using Python as a Calculator](https://docs.python.org/2/tutorial/introduction.html#using-python-as-a-calculator)
    - [3.1.1. Numbers](https://docs.python.org/2/tutorial/introduction.html#numbers)
    - [3.1.2. Strings](https://docs.python.org/2/tutorial/introduction.html#strings)
    - [3.1.3. Unicode Strings](https://docs.python.org/2/tutorial/introduction.html#unicode-strings)
    - [3.1.4. Lists](https://docs.python.org/2/tutorial/introduction.html#lists)
  + [3.2. First Steps Towards Programming](https://docs.python.org/2/tutorial/introduction.html#first-steps-towards-programming)
* [4. More Control Flow Tools](https://docs.python.org/2/tutorial/controlflow.html)
  + [4.1. **if** Statements](https://docs.python.org/2/tutorial/controlflow.html#if-statements)
  + [4.2. **for** Statements](https://docs.python.org/2/tutorial/controlflow.html#for-statements)
  + [4.3. The **range()** Function](https://docs.python.org/2/tutorial/controlflow.html#the-range-function)
  + [4.4. **break** and **continue** Statements, and **else** Clauses on Loops](https://docs.python.org/2/tutorial/controlflow.html#break-and-continue-statements-and-else-clauses-on-loops)
  + [4.5. **pass** Statements](https://docs.python.org/2/tutorial/controlflow.html#pass-statements)
  + [4.6. Defining Functions](https://docs.python.org/2/tutorial/controlflow.html#defining-functions)
  + [4.7. More on Defining Functions](https://docs.python.org/2/tutorial/controlflow.html#more-on-defining-functions)
    - [4.7.1. Default Argument Values](https://docs.python.org/2/tutorial/controlflow.html#default-argument-values)
    - [4.7.2. Keyword Arguments](https://docs.python.org/2/tutorial/controlflow.html#keyword-arguments)
    - [4.7.3. Arbitrary Argument Lists](https://docs.python.org/2/tutorial/controlflow.html#arbitrary-argument-lists)
    - [4.7.4. Unpacking Argument Lists](https://docs.python.org/2/tutorial/controlflow.html#unpacking-argument-lists)
    - [4.7.5. Lambda Expressions](https://docs.python.org/2/tutorial/controlflow.html#lambda-expressions)
    - [4.7.6. Documentation Strings](https://docs.python.org/2/tutorial/controlflow.html#documentation-strings)
  + [4.8. Intermezzo: Coding Style](https://docs.python.org/2/tutorial/controlflow.html#intermezzo-coding-style)
* [5. Data Structures](https://docs.python.org/2/tutorial/datastructures.html)
  + [5.1. More on Lists](https://docs.python.org/2/tutorial/datastructures.html#more-on-lists)
    - [5.1.1. Using Lists as Stacks](https://docs.python.org/2/tutorial/datastructures.html#using-lists-as-stacks)
    - [5.1.2. Using Lists as Queues](https://docs.python.org/2/tutorial/datastructures.html#using-lists-as-queues)
    - [5.1.3. Functional Programming Tools](https://docs.python.org/2/tutorial/datastructures.html#functional-programming-tools)
    - [5.1.4. List Comprehensions](https://docs.python.org/2/tutorial/datastructures.html#list-comprehensions)
      * [5.1.4.1. Nested List Comprehensions](https://docs.python.org/2/tutorial/datastructures.html#nested-list-comprehensions)
  + [5.2. The **del** statement](https://docs.python.org/2/tutorial/datastructures.html#the-del-statement)
  + [5.3. Tuples and Sequences](https://docs.python.org/2/tutorial/datastructures.html#tuples-and-sequences)
  + [5.4. Sets](https://docs.python.org/2/tutorial/datastructures.html#sets)
  + [5.5. Dictionaries](https://docs.python.org/2/tutorial/datastructures.html#dictionaries)
  + [5.6. Looping Techniques](https://docs.python.org/2/tutorial/datastructures.html#looping-techniques)
  + [5.7. More on Conditions](https://docs.python.org/2/tutorial/datastructures.html#more-on-conditions)
  + [5.8. Comparing Sequences and Other Types](https://docs.python.org/2/tutorial/datastructures.html#comparing-sequences-and-other-types)
* [6. Modules](https://docs.python.org/2/tutorial/modules.html)
  + [6.1. More on Modules](https://docs.python.org/2/tutorial/modules.html#more-on-modules)
    - [6.1.1. Executing modules as scripts](https://docs.python.org/2/tutorial/modules.html#executing-modules-as-scripts)
    - [6.1.2. The Module Search Path](https://docs.python.org/2/tutorial/modules.html#the-module-search-path)
    - [6.1.3. “Compiled” Python files](https://docs.python.org/2/tutorial/modules.html#compiled-python-files)
  + [6.2. Standard Modules](https://docs.python.org/2/tutorial/modules.html#standard-modules)
  + [6.3. The **dir()** Function](https://docs.python.org/2/tutorial/modules.html#the-dir-function)
  + [6.4. Packages](https://docs.python.org/2/tutorial/modules.html#packages)
    - [6.4.1. Importing \* From a Package](https://docs.python.org/2/tutorial/modules.html#importing-from-a-package)
    - [6.4.2. Intra-package References](https://docs.python.org/2/tutorial/modules.html#intra-package-references)
    - [6.4.3. Packages in Multiple Directories](https://docs.python.org/2/tutorial/modules.html#packages-in-multiple-directories)
* [7. Input and Output](https://docs.python.org/2/tutorial/inputoutput.html)
  + [7.1. Fancier Output Formatting](https://docs.python.org/2/tutorial/inputoutput.html#fancier-output-formatting)
    - [7.1.1. Old string formatting](https://docs.python.org/2/tutorial/inputoutput.html#old-string-formatting)
  + [7.2. Reading and Writing Files](https://docs.python.org/2/tutorial/inputoutput.html#reading-and-writing-files)
    - [7.2.1. Methods of File Objects](https://docs.python.org/2/tutorial/inputoutput.html#methods-of-file-objects)
    - [7.2.2. Saving structured data with **json**](https://docs.python.org/2/tutorial/inputoutput.html#saving-structured-data-with-json)
* [8. Errors and Exceptions](https://docs.python.org/2/tutorial/errors.html)
  + [8.1. Syntax Errors](https://docs.python.org/2/tutorial/errors.html#syntax-errors)
  + [8.2. Exceptions](https://docs.python.org/2/tutorial/errors.html#exceptions)
  + [8.3. Handling Exceptions](https://docs.python.org/2/tutorial/errors.html#handling-exceptions)
  + [8.4. Raising Exceptions](https://docs.python.org/2/tutorial/errors.html#raising-exceptions)
  + [8.5. User-defined Exceptions](https://docs.python.org/2/tutorial/errors.html#user-defined-exceptions)
  + [8.6. Defining Clean-up Actions](https://docs.python.org/2/tutorial/errors.html#defining-clean-up-actions)
  + [8.7. Predefined Clean-up Actions](https://docs.python.org/2/tutorial/errors.html#predefined-clean-up-actions)
* [9. Classes](https://docs.python.org/2/tutorial/classes.html)
  + [9.1. A Word About Names and Objects](https://docs.python.org/2/tutorial/classes.html#a-word-about-names-and-objects)
  + [9.2. Python Scopes and Namespaces](https://docs.python.org/2/tutorial/classes.html#python-scopes-and-namespaces)
  + [9.3. A First Look at Classes](https://docs.python.org/2/tutorial/classes.html#a-first-look-at-classes)
    - [9.3.1. Class Definition Syntax](https://docs.python.org/2/tutorial/classes.html#class-definition-syntax)
    - [9.3.2. Class Objects](https://docs.python.org/2/tutorial/classes.html#class-objects)
    - [9.3.3. Instance Objects](https://docs.python.org/2/tutorial/classes.html#instance-objects)
    - [9.3.4. Method Objects](https://docs.python.org/2/tutorial/classes.html#method-objects)
    - [9.3.5. Class and Instance Variables](https://docs.python.org/2/tutorial/classes.html#class-and-instance-variables)
  + [9.4. Random Remarks](https://docs.python.org/2/tutorial/classes.html#random-remarks)
  + [9.5. Inheritance](https://docs.python.org/2/tutorial/classes.html#inheritance)
    - [9.5.1. Multiple Inheritance](https://docs.python.org/2/tutorial/classes.html#multiple-inheritance)
  + [9.6. Private Variables and Class-local References](https://docs.python.org/2/tutorial/classes.html#private-variables-and-class-local-references)
  + [9.7. Odds and Ends](https://docs.python.org/2/tutorial/classes.html#odds-and-ends)
  + [9.8. Exceptions Are Classes Too](https://docs.python.org/2/tutorial/classes.html#exceptions-are-classes-too)
  + [9.9. Iterators](https://docs.python.org/2/tutorial/classes.html#iterators)
  + [9.10. Generators](https://docs.python.org/2/tutorial/classes.html#generators)
  + [9.11. Generator Expressions](https://docs.python.org/2/tutorial/classes.html#generator-expressions)
* [10. Brief Tour of the Standard Library](https://docs.python.org/2/tutorial/stdlib.html)
  + [10.1. Operating System Interface](https://docs.python.org/2/tutorial/stdlib.html#operating-system-interface)
  + [10.2. File Wildcards](https://docs.python.org/2/tutorial/stdlib.html#file-wildcards)
  + [10.3. Command Line Arguments](https://docs.python.org/2/tutorial/stdlib.html#command-line-arguments)
  + [10.4. Error Output Redirection and Program Termination](https://docs.python.org/2/tutorial/stdlib.html#error-output-redirection-and-program-termination)
  + [10.5. String Pattern Matching](https://docs.python.org/2/tutorial/stdlib.html#string-pattern-matching)
  + [10.6. Mathematics](https://docs.python.org/2/tutorial/stdlib.html#mathematics)
  + [10.7. Internet Access](https://docs.python.org/2/tutorial/stdlib.html#internet-access)
  + [10.8. Dates and Times](https://docs.python.org/2/tutorial/stdlib.html#dates-and-times)

*Project*

*Demo to create a python app on Raspberry Pie & light sensors. A message will be generated on registered number on waving hand in front of sensor.*

**TRAINER PROFILE**

8+ Years of corporate experience with CISCO Systems, Juniper Networks & Reliance Industries, Co-Founded MagicLifeMantra Technologies Pvt. Ltd. All these bring an unique training experience which is a perfect blend of current industry trend & real time issues.

**EDUCATION**

B.Tech from MNNIT Allahabad. Computer Science & Engineering, 2007

**CORPORATE EXPERIENCE**

Jan 2015 – Oct 2015, Co-Founder MagicLifeMantra Tech. Pvt. Ltd.

**Domain**: Automated content publisher, appointment booking system, profile browsing, anonymous discussion forum.

**Languages & Technologies:** Java, Spring, Hibernate, Python, Databases, Django, Hadoop,Jan 2012 – Dec 2014, CISCO Systems – Sr. Member of Technical Staff

**Domain**: Analytics Driven Switches & Routers. QoS, Netwflow, ACL, sFlow,

Inventions: Automated Virtual Switching System, Controller Based Dynamic QoS, Analytics based Networking

**Skill:** Java, Python, Databases, Spring

Jan 2010 – Dec 2011, Juniper Networks – Sr. Software Developer

**Domain**: Programmable Network Devices, Network Analytics, Automatic Deployment

Skill: C, Java, Python

Sept 2007 – Dec 2009, Reliance Industries – Software Developer

**Domain:** WiMAX Apps for Management & Monitoring

**Skill:** Java, C, C++

TRAINING EXPERTISE

Trained more than 500 candidates so far.

In association with Real Time Signals Pvt. Ltd & Inventateq for Core Java, Adv. Java trainings & Hadoop.Python Trainer at Real Time Signals & Inventateq Pvt. Ltd, Marathahalli.Provide consulting to working professionals & startups.

Java Training Details – Specialized in Collections, Spring, Threading, Web Services, Hibernate, Web Scrapping, Java 8.Java training for employees from CISCO, IBM, Cognizant & CTS.

In house Java 8 & Web Services training at DataGenic.Python training for employees of EMC2, MaCafee. Python trainer at SkillSpeed

4. Student Engagement

Will be able to engage and control the batch of students and understand each student’s specific learning

need and clarify questions and doubts with real life examples to the satisfaction of each student.

5. Values & Ethics

Will display a high level of personal discipline, hygiene and adhere to the ethics of education morals and

values in the organization.

6. Quality Delivery

The trainer will be keenly aware that he will be evaluated on the quality of his delivery and will organize

frequent feedback from his students to evaluate his performance and continuously improve.