

# Python Syllabus

For Apprenticeship Training Program

```
1 command = ""
2 while command != "quit":
3     command = input("> ").lower()
4     if command == "start":
5         print('Car starts.')
6     elif command == "stop":
7         print('Car stopped.')
8     elif command == "help":
9         print("""
10 start - to start the car
11 stop - to stop the car
12 quit - quit the game.
13 """)
14 else:
15     print("Sorry we dont un
```

PYTHON AI



**HINDTECH**  
LEARNING POINT





Python is a powerful, high-level, interpreted programming language that emphasizes code readability, developer productivity, and a minimalist design philosophy. Developed by Guido van Rossum and released in 1991, Python was intended to be an easy-to-understand language that could help both beginners and professionals write clean and maintainable code.

It supports multiple programming paradigms, including procedural, object-oriented, and functional programming, giving developers the flexibility to choose the most suitable approach for a given task. One of Python's most appealing features is its dynamic typing and automatic memory management, which allows developers to write less boilerplate code and focus more on solving real problems. The Python interpreter processes code line-by-line, making debugging easier and more intuitive. Its extensive standard library—sometimes referred to as "batteries included"—covers everything from file I/O and system calls to regular expressions, web services, and unit testing, drastically reducing the need for external dependencies in many applications.

In addition to the built-in library, Python boasts a massive ecosystem of third-party modules and frameworks available through the Python Package Index (PyPI). These libraries extend Python's capabilities in nearly every domain of computing. In web development, frameworks like Django, Flask, and FastAPI enable developers to build full-scale, secure web applications with minimal overhead. For scientific computing and data analysis, libraries such as NumPy, SciPy, Pandas, and Matplotlib offer high-performance operations on arrays, statistical analysis tools, and powerful data visualizations. Python has become the de facto language for artificial intelligence (AI), machine learning (ML), and deep learning, with tools like TensorFlow, PyTorch, Keras, and Scikit-learn facilitating the development of complex neural networks and predictive models. The language also excels in automation, scripting, and system administration tasks thanks to modules like os, shutil, and subprocess.

Python's cross-platform nature ensures that programs written on one operating system can usually run on another with little or no modification. This portability, combined with the simplicity of Python syntax, makes it ideal for rapid application development and prototyping. Educational institutions worldwide adopt Python as the first language for computer science students because it abstracts many of the complexities found in lower-level languages while still exposing the fundamentals of software development.





## MODULE 1: PYTHON PROGRAMMING – FROM BASICS TO INTERMEDIATE

**Objective:** To build foundational Python programming skills required for image processing.

### Introduction to Python

- Installation (Anaconda, Jupyter, VS Code)
- Basic syntax, variables, data types

### Control Structures

- If-else, loops (for, while)

### Functions and Modules

- Defining functions, importing modules

### Data Structures

- Lists, tuples, dictionaries, sets

### File Handling in Python

### Exception Handling

### Object-Oriented Programming Basics

### Working with Libraries (NumPy, Matplotlib, OpenCV)

## MODULE 2: BASICS OF IMAGE PROCESSING

**Objective:** To understand image representation, manipulation, and analysis using Python.

### Introduction to Image Processing

### Loading and Displaying Images (Matplotlib, OpenCV)

### Image Types (Grayscale, RGB, Binary)

### Image Transformations

- Resizing, Cropping, Rotation, Flipping

### Image Enhancement

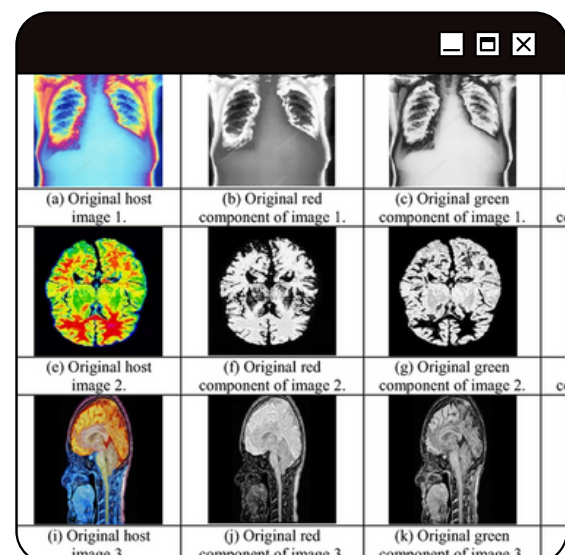
- Brightness, Contrast, Histogram Equalization

### Filters and Noise Removal

- Gaussian, Median, Sharpening

### Edge Detection Techniques

- Sobel, Canny





### MODULE 3: BASIC CONCEPTS OF THE CARTESIAN COORDINATE SYSTEM

**Objective:** To understand how Cartesian coordinates apply in image positioning and spatial data.

- What is a Cartesian Plane?
- Origin, Axes, Quadrants
- Points and Coordinates
- Distance Formula, Midpoint
- Coordinate Systems in Digital Imaging
- Pixel Coordinates vs Physical Coordinates

### MODULE 4: FUNDAMENTALS OF DIGITAL IMAGES

**Objective:** To explore the basic building blocks and properties of digital images.

- What is a Digital Image?
- Image Resolution
- Pixel Depth (Bit Depth)
- Image Size vs File Size
- Grayscale vs Color Images
- Sampling and Quantization
- Image Histogram and Intensity Levels

### MODULE 5: DIGITAL IMAGE CHARACTERISTICS

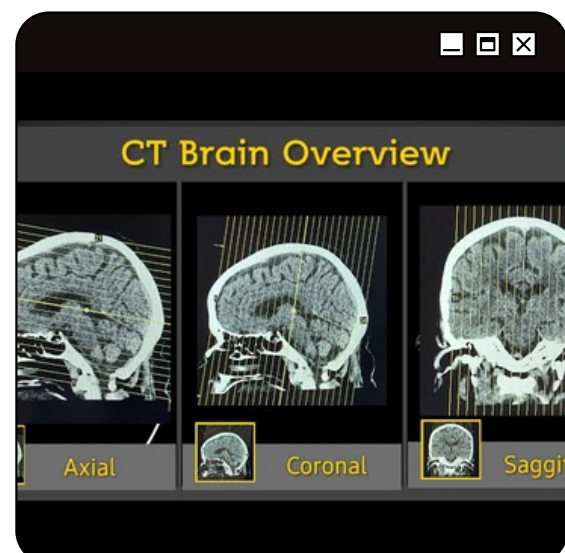
**Objective:** To define the measurable properties of digital images.

- Spatial Resolution
- Contrast Resolution
- Signal-to-Noise Ratio (SNR)
- Dynamic Range
- Image Fidelity and Artifacts
- Image Compression (Lossy vs Lossless)

### MODULE 6: INTRODUCTION TO MEDICAL IMAGING

**Objective:** To understand the role and basics of medical imaging.

- What is Medical Imaging?
- History and Evolution
- Importance in Diagnosis and Treatment
- Market Value and Industry Scope





### Medical Imaging Modalities:

- X-Ray, MRI, CT, PET, Ultrasound
- Comparison Table

## MODULE 7: IMAGE FORMATS IN MEDICAL IMAGING

**Objective:** To identify and differentiate between various image formats used in medical imaging.

Common Medical Image Formats:

1. Bitmap (BMP)
2. Portable Network Graphics (PNG)
3. Joint Photographic Experts Group (JPEG)
4. DICOM (Digital Imaging and Communications in Medicine)
5. NIFTI (Neuroimaging Informatics Technology Initiative)
6. TIFF (Tagged Image File Format)
7. STL (Stereolithography for 3D Printing)

Format Comparison:

- Compression type
- Metadata support
- Use cases in medical field

## MODULE 8: ESSENTIAL PYTHON LIBRARIES FOR IMAGE ANALYSIS

**Objective:** To introduce important Python libraries for image visualization and analysis.

Matplotlib

- Basic plotting
- Displaying images
- Histograms and Intensity graphs

NumPy

- Array operations on image data
- Matrix representation

OpenCV

- Reading and writing medical images
- Image transformations

Pydicom

- Reading DICOM files
- Extracting metadata





## Nibabel

- Working with NIfTI images

## SimpleITK / ITK

- Advanced medical image analysis
- Image registration and segmentation

## MODULE 9: GETTING STARTED WITH DJANGO

**Objective:** Understand the basics of web development and Django, and set up the development environment.

- Introduction to Web Development
- What is Django and why use it?
- Installing Python and Django
- Creating a virtual environment (venv/pipenv/poetry)
- Django project and app structure
- Creating your first Django project and app

## MODULE 10: BUILDING THE BACKEND

**Objective:** Learn how Django handles backend logic using the MTV (Model-Template-View) architecture.

Topics:

- Understanding the MTV architecture
- Writing views and connecting them with URLs
- Creating models and making migrations
- Working with Django ORM and QuerySets
- Customizing the Django admin panel

## MODULE 11: DATABASE INTEGRATION

**Objective:** Learn how to integrate and work with different relational databases in Django.

Topics:

- SQLite (default for development)
- PostgreSQL / MySQL setup
- Database configuration in settings.py







- Defining model relationships (OneToOne, ForeignKey, ManyToMany)
- Using raw SQL with ORM

## MODULE 12: FRONTEND DEVELOPMENT WITH DJANGO

**Objective:** Design and build dynamic and styled user interfaces using Django templates and frontend technologies.

- Django template engine
- Template inheritance and template tags
- Rendering dynamic content
- Styling with CSS/Bootstrap
- Adding JavaScript for interactivity
- Managing static files

## MODULE 13: SECURITY & USER MANAGEMENT

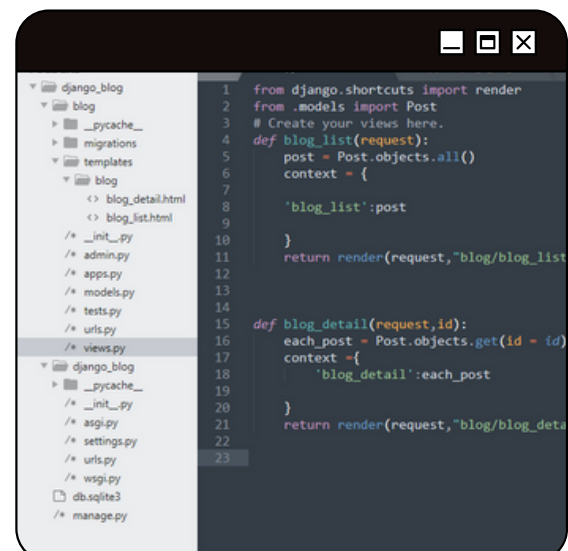
**Objective:** Implement secure authentication, user management, and protect your application from common threats.

- Django's built-in authentication system
- User registration, login, and logout
- Password management (change/reset)
- CSRF protection and middleware
- Form validation and input sanitization
- User permissions, groups, and access control

## MODULE 14: BUILDING RESTFUL APIS WITH DRF

**Objective:** Use Django REST Framework (DRF) to create RESTful APIs for mobile and frontend apps. Topics.

- Introduction to Django REST Framework
- Serializers and viewsets
- Creating CRUD APIs
- Token and JWT-based authentication
- API filtering, pagination, and versioning
- Using routers and API testing





## MODULE 15: STATIC & MEDIA FILE HANDLING

**Objective:** Properly manage and serve static assets and user-uploaded media in Django projects.

- Serving static files with collectstatic
- Configuring and uploading media files
- Using MEDIA\_ROOT and STATIC\_ROOT
- Integration with AWS S3 using django-storages
- Serving static files in production with WhiteNoise

## MODULE 16: DEPLOYMENT & HOSTING

**Objective:** Deploy Django projects to production servers and cloud platforms.

- Understanding WSGI and ASGI
- Using Gunicorn or uWSGI with Django
- Setting up Nginx as a reverse proxy
- Deploying on Heroku, PythonAnywhere, or DigitalOcean
- Using environment variables and secrets management
- Setting up SSL and HTTPS

## MODULE 17: DEVELOPER TOOLS & BEST PRACTICES

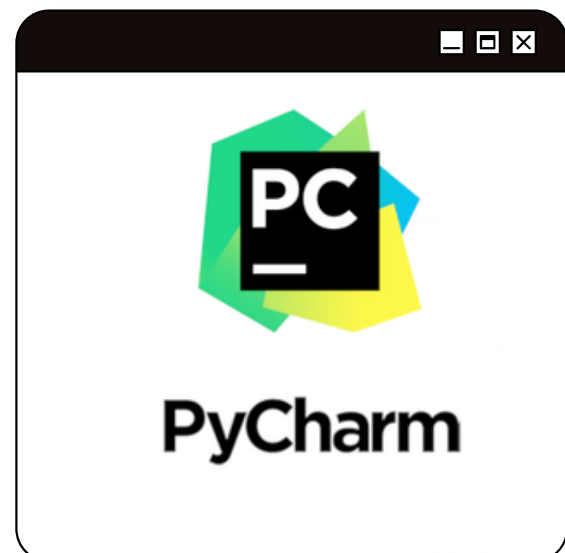
**Objective:** Use professional tools and follow best practices to improve code quality and team collaboration.

- Version control using Git and GitHub
- Working with IDEs: VSCode / PyCharm
- Managing settings using .env files
- Writing modular, reusable, and clean code
- Debugging tools and log management
- Code reviews and documentation

## PROJECT DEVELOPMENT

### 1. Medical Imaging using Data Science and Machine Learning

**DURATION : 180 DAYS**

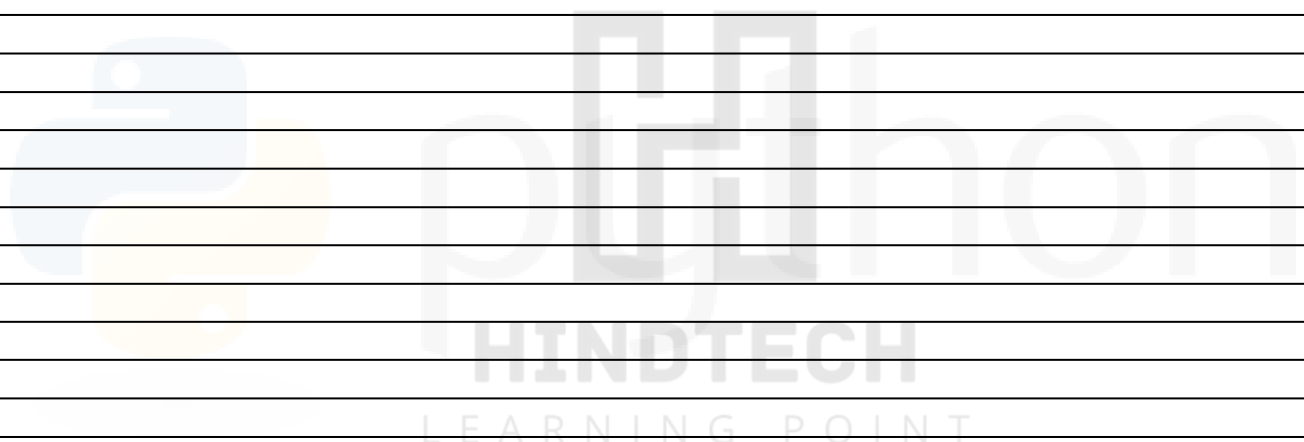




**OUTCOME:**

1. **Master Python Programming:** Understand and apply core Python programming concepts, including data structures, control flow, functions, and object-oriented programming relevant to technical and scientific applications.
2. **Understand and Apply Image Processing Techniques:** Gain hands-on experience in basic image processing operations such as image enhancement, filtering, transformation, and edge detection using Python libraries like OpenCV and Matplotlib.
3. **Apply Cartesian Coordinate Concepts in Imaging:** Understand the Cartesian coordinate system and its application in positioning, pixel indexing, and geometric image operations.
4. **Comprehend the Fundamentals of Digital Images:** Understand digital image structure, resolution, grayscale levels, image formats, and characteristics that impact quality and diagnostics.
5. **Gain Knowledge of Medical Imaging and Modalities:** Learn about various medical imaging technologies (CT, MRI, X-ray, PET, Ultrasound) and how image processing is used in healthcare.
6. **Explore Image Preprocessing Techniques:** Understand techniques such as noise reduction, contrast enhancement, image filtering, and normalization to improve image quality and prepare data for analysis in medical imaging.
7. **Understand Image Segmentation and Feature Extraction:** Learn how to isolate regions of interest (e.g., tumors, organs) and extract meaningful features (e.g., shape, texture, intensity) that are crucial for diagnosis, classification, and treatment planning.

**PROJECT VIEW**

The logo for HindTech Learning Point is centered at the top of the page. It features a stylized 'H' icon composed of blue and yellow segments. To the right of the icon, the word 'hindtech' is written in a large, light blue, sans-serif font. Below 'hindtech', the words 'LEARNING POINT' are written in a smaller, all-caps, light blue, sans-serif font.



## Hindtech Learning Point

A Trusted IT Institute Brand In Lucknow

Admission Open For

### Summer Training

CSE/IT

Diploma, B.Tech/B.E, BCA, MCA

45 - 50 Days Program

### Apprenticeship Training

CSE/IT

B.Tech, BCA, MCA, PGDCA, Dip.

6 to 8 Months Program

### Industrial Training

CSE/IT

B.Tech, BCA, MCA, PGDCA, Dip.

3 to 5 Months Program

### UI/UX Designing

CSE/IT or interested student

B.Tech, BCA, MCA, PGDCA, Dip.

45 - 50 Days Program

1  
Number  
ONE

Excellent Performance  
by Hindtech Learning Point  
in IT Training



Since 2020, 1000+ students placed in top IT firms | Parmar Plaza, BKT, Bargadi, UP 226201 | Call: 6307738600 / 7905320279 | www.hindtechlearningpoint.com



**HINDTECH**  
IT SOLUTIONS

### Training

Python Apprenticeship Training Program 6 to 8 Months

### Amount

₹21,000

[www.hindtechitsolutions.com](http://www.hindtechitsolutions.com) [hr@hindtechitsolutions.com](mailto:hr@hindtechitsolutions.com)

+91 7905320279

[www.hindtechlearningpoint.com](http://www.hindtechlearningpoint.com) [hindtechlearningpointlko@gmail.com](mailto:hindtechlearningpointlko@gmail.com)

+91 6307738600



Building No 10/703, Ground Floor, near Arvindo Park Road, Sector 10, Indira Nagar, Lucknow, Uttar Pradesh 226016

### Branch 1



PARMAR PLAZA, 1st Floor, Bakshi Ka Talab, Bargadi Magath, Uttar Pradesh 226201

### Branch 2

Scan Me:



Scan this QR code for details



**HINDTECH**  
LEARNING POINT

