Python Programming Course Syllabus

DETAILS

Python was designed to be easy to understand and fun to use (its name came from Monty Python so a lot of its beginner tutorials reference it). Fun is a great motivator, and since you'll be able to build prototypes and tools quickly with Python, many find coding in Python a satisfying experience. Thus, Python has gained popularity for being a beginner-friendly language, and it has replaced Java as the most popular introductory language at Top U.S. Universities.

Hands-on Programming Practice

Techno Knowledge Center

www.tkc.drdineshsharma.com

Why Learn Python?

Python is a general-purpose language, which means it can be used to build just about anything, which is made easy with the right tools/libraries. Professionally, Python is great for backend web development, data analysis, artificial intelligence, and scientific computing. Many developers have also used Python to build productivity tools, games, and desktop apps, so there are plenty of resources to help you learn how to do those as well.

Python can be used for a wide variety of applications, and as you learn the basics of Python, you'll be able to create almost anything you want. Many great developers contribute daily to the Python community by creating Python libraries. These libraries can help you get started so that you don't have to write code to reinvent the wheel. So for example, if you want to do complex image processing, the Python Imaging Library will help you get started. Want to create games? PyGame is a Python game engine. If data science is your thing, SciPy is the library for you.

"Why" Python has emerged as one of the most popular programming languages on the market:

- ***QUICK TO SETUP***: **PYTHON IS EASY TO DOWNLOAD, EVEN FOR THE NEWBIE; CAREFUL DOCUMENTATION TAKES YOU THROUGH THE DOWNLOAD AND SETUP STEPS IN EITHER WINDOWS, MAC, OR LINUX ENVIRONMENTS. TONS OF SUPPORT AND DOCUMENTATION MAKE PYTHON LEARNING FAIRLY MANAGEABLE. IF YOU WANT TO JUMP RIGHT INTO PYTHON WITHOUT ANY NEED FOR DOWNLOAD ON YOUR MACHINE, JUST.**

- ***PYTHON IS FAST***: **PYTHON HAS DEVELOPED A REPUTATION AS A SOLID, HIGH-PERFORMANCE LANGUAGE. LOTS HAS BEEN DONE IN RECENT YEARS TO GET TO THIS POINT. THE PYPY PROJECT AIMS TO SPEED UP PYTHON AS A WHOLE (AND IS DOING A GREAT JOB OF IT). AND NUMBA IS ANOTHER TOOL THAT CAN OFFER AMAZING SPEEDUPS BY IMPLEMENTING HIGH PERFORMANCE FUNCTIONS WRITTEN DIRECTLY IN PYTHON.**

- ***PYTHON HAS BROAD SUPPORT***: **THE APPLICATIONS FOR PYTHON ARE BROAD AND VARIED; IT'S USED BY INDIVIDUALS AND BIG INDUSTRY PLAYERS ALIKE IN EVERYTHING FROM SYSTEMS AUTOMATION, TESTING, AND ETL TO GAMING, CGI AND WEB DEVELOPMENT. DISNEY USES PYTHON TO HELP POWER THEIR**

***CREATIVE PROCESS*. AND MOZILLA RELEASES TONS OF OPEN SOURCE PACKAGES BUILT IN PYTHON. BANK OF AMERICA USES PYTHON TO BUILD NEW PRODUCTS AND INTERFACES WITHIN THE BANK'S TECHNOLOGY INFRASTRUCTURE.**

- ***EASE OF USE*: PYTHON GETS A LOT OF ACCOLADES FOR BEING EASY TO LEARN, AND RIGHTFULLY SO. THE LEARNING CURVE IS VERY GRADUAL. OTHER LANGUAGES CAN BE QUITE STEEP. PYTHON PLACES A HEAVY EMPHASIS ON READABILITY, AS SHOWN BY ITS COMPARISON WITH OTHER OBJECT-ORIENTED LANGUAGES.**

Python powers Django, a complete and open source web application framework. Frameworks - like Ruby on Rails - can be used to simplify the development process. It has a massive support base thanks to the fact that it is open source and community developed. Millions of like-minded developers work with the language on a daily basis and continue to improve the functionality.
The latest version of Python continues to receive enhancements and updates as time progresses. This is a great way to network with other developers.
Finally, Python is widely used, including by a number of big companies like Google, Pinterest, Instagram, Disney, Yahoo!, Nokia, IBM, and many others. The Raspberry Pi - which is a mini computer and DIY lover's dream - relies on Python as its main programming language. You're probably wondering why either of these things matter, and that's because once you learn Python, you'll never have a shortage of ways to utilize the skill. Not to mention, since a lot of big companies rely on the language, you can make good money as a Python developer

COURSE OVERVIEW

- Introduction to Python
- Installing Python
- IDE: Pycharm, Jupyter
- Installing Pycharm
- Installing Jupyter
- Writing Your First Python Program
- Data types in Python
- Operators in Python
- Input and Output
- Control Statements
- Arrays in Python
- Strings and Characters

- Functions
- Lists and Tuples
- Dictionaries
- Introduction to Object Oriented Programming System
- Classes and Objects
- Files in Python
- Date and time
- and much more
- Web development Framework: Django
- Data Science Frameworks: Numpy, Panda, Matplotlib
- Hands on Projects.
- Learn GitHub for FREE. (BONUS)

DETAILED SYLLABUS BREAKDOWN

INTRODUCTION TO PYTHON

- Overview
- What is Python?
- Why should you learn Python?

GETTING STARTED WITH PYTHON PROGRAMMING LANGUAGE

- Creating variables and assigning values
- Block Indentation
- IDLE - Python GUI
- Installation of Python
- Installing external modules using pip

CODING STANDARDS IN PYTHON & BEST PRACTICES TO FOLLOW

- Understanding what the international coding standards are
- Why to follow coding standards
  How following coding standards improve security & accuracy of code
- Best Practices followed by developers

PYTHON DATA TYPES

- String Data Type
- Set Data Types
- Numbers data type
- List Data Type
- Dictionary Data Type
- Tuple Data Type

INDENTATION

- Simple example
- How Indentation is Parsed
- Indentation Errors
- Comments and Documentation
- Single line, inline and multiline comments

DATE AND TIME IN PYTHON

- Parsing a string into a timezone aware datetime object
- Computing time differences
- Basic datetime objects usage
- Simple date arithmetic
- Date Formatting
- Time between two date-times
- Outputting datetime object to string
- Parsing string to datetime object

ENUM

- Creating an enum
- Iteration

SETS IN PYTHON

- Operations on sets
- Get the unique elements of a list
- Set of Sets
- Set Operations using Methods and Builtins
- Sets versus multisets

OPERATORS

- Bitwise
- Boolean
- Operator Precedence

SCOPE AND BINDING OF VARIABLE

- Nonlocal Variables
- Global Variables
- Local Variables
  The del command
- Binding Occurrence

CONDITIONALS AND LOOPS IN PYTHON

- Conditionals
- Loops

ARRAYS

- Basic Introduction to Arrays
- Access individual elements through indexes
- Different Methods of Array
- Multidimensional arrays
- Lists in lists
- Lists in lists in lists in..

DICTIONARY

- Introduction to Dictionary
- Avoiding KeyError Exceptions
- Iterating Over a Dictionary
- Merging dictionaries
- Accessing keys and values
- Creating a dictionary
- Dictionaries Example

LISTS AND STRING

- List methods and supported operators
- Accessing list values
- Checking if list is empty
- Iterating over a list
- Length of a list
- Reversing list elements
- List comprehensions
- List slicing

LINKED LISTS

- Single linked list example
- Write a simple Linked List Node in python

FILTER

- Basic of filter
- Filter without function
- Filter as short-circuit check

FUNCTIONS AND MODULES

- Defining and calling simple functions
- Defining a function with an arbitrary number of arguments
- Lambda (Inline/Anonymous) Functions
- Returning more than one value
- Recursive functions

FUNCTIONAL PROGRAMMING IN PYTHON

- Lambda Function
- Map Function
- Reduce Function
- Filter Function

STRING

- String Formatting
- String Methods

IMPORTING MODULES

- Importing a module
- The__all__special variable
- Import modules from an arbitrary filesystem location
- Importing all names from a module
- Programmatic importing
- PEP8 rules for Imports
- Importing specific names from a module
- __import__() function

JSON MODULE

- Storing data in a file
- Retrieving data from a file
- Formatting JSON output
- Calling `json.tool` from the command line to pretty-print JSON output
- Creating JSON from Python dict
- Creating Python dict from JSON

DEBUGGING AND EXCEPTION HANDLING

- Using print for debugging
- Use the Command Line
- Module Test Code
- Handling Exceptions: try / except – Advanced Topic

MATHS MODULES – NUMPY

- Using NumPy
- Addressing and Slicing Arrays
- The math Module
- The NumPy Module
- Creating Arrays and Some Examples of Basic Manipulation
- Linear Algebra

PANDAS

- Look Ups, Selections and Indexing, Filling Methods, Series operation, Handling NaN values
- Mapping, Data Frames, Reading Files, Plotting
- Joins, Correlation, Histograms

FILE INPUT AND OUTPUT

- Line Terminators – The \n character
- Writing to File
- Reading from File
- Saving an Array to File
- Loading an Array from File

PYTHON DATABASE CONNECTIVITY WITH MYSQL

At the end of the course, participants will be able to:

- Evaluate problems and analyze data using Python
- Analysis of data using Histograms
- Creating application using python like login page, registration page etc.
- Participant will able to do data analysis using different data set e.g. IRIS