

Python:**Complete Python****• Demo:**

- i. WhyPython?
 - ii. Who Uses Python today
 - iii. What can we do with Python
 - iv. How Python Developed and Supported
 - v. Python–Technical Strengths.
 - vi. What next?
- Python Interpreter
 - Program Execution–programmer’s view, Python’s view
 - Installation
 - i. Python
 - ii. All Related Software: PyCharm, Anaconda
 - iii. Setup,configure Python in Laptop
 - iv. IDLE– UI, usage, features
 - All Numeric types in Python, coding / Hands-on
 - Python Variables, objects, References, Shared References, coding/Hands-on
 - Garbage Collection of objects
 - All builtin types in python: Strings, Lists, Dictionaries, Tuples, sets, Files
 - Python Statements-coding / Hands-on
 - Assignments, Expressions and Prints
 - i. if-else,if-elif-else,if-else ternary expression
 - ii. while and for loops
 - iii. Comprehensions vs regular
 - iv. Parallel Traversals: map and zip functions
 - v. Other important functions: range, len, enumerate
 - Iterations and Comprehensions-coding/Hands-on
 - Python online Documentation
 - Python Functions–def, nested functions
 - Variable Scopes –basics, LEGB Rules, global, nonlocal
 - Function Arguments-coding/Hands-on
 - i. Arguments and shared references
 - ii. Arguments passing basics
 - iii. Arguments matching basics
 - iv. Arguments matching syntax
 - v. Multiple Results
 - Advanced Function Concepts–Recursive functions, Attributes, Annotations, lambda

- Generators and comprehensions– Generator functions, yield, Generator expression
- **Python Modules**
 - i. Definition, why modules?
 - ii. Typical Python program architecture
 - iii. Import statement-coding/Hands-on
 - iv. How Import works: Findit, Compileit, Runit
 - v. Standard library modules
 - vi. Pycache folder for byte codefiles
 - vii. Module search path
- **Module coding Basics**
 - i. Module creation
 - ii. import statement, from statement, from*statement
 - iii. Module Name spaces, Name space dictionaries: dict__
 - iv. Reloading modules
- **Module Packages**
 - i. Package import basics
 - ii. Why Package imports?
 - iii. Relative import basics
 - iv. Why relative imports?
 - v. Package Name spaces
- **Advanced Module Topics**
 - i. Data hiding in modules -coding/Hands-on
 - ii. Mixed usage modes: name__and main__-coding/Hands-on
 - iii. Theas extension for import and from-coding/Hands-on
- **Introduction to Python Classes-coding/Hands-on**
 - i. Why Classes?
 - ii. Classes, constructors and Instances
 - iii. Method calls
 - iv. Attribute in heritance search
 - v. OOPisab out codereuse
 - vi. Subclassing by Inheritance
 - vii. Polymorphism in Action
 - viii. Class vs instance attributes
 - ix. Storing objects in DB–Pickles & Shelves

- **Coding with Classes-coding/Hands-on**
 - i. Abstract super classes
 - ii. Nested classes
 - iii. Classes vs Modules
 - iv. Name space dictionaries
 - v. LEGB scopesrule revisited

- **Operator Overloading-coding/Hands-on**
 - i. Constructors: init
 - ii. Indexing, Slicing: getitem_ and setitem
 - iii. Attribute Access: getattr and setattr
 - iv. String Representation: repr andstr
 - v. Rightside andIn-PlaceUses: radd and iadd
 - vi. CallExpressions: call
 - vii. Comparisons: lt,gt_andothers
 - viii. BooleanTests: bool andlen
 - ix. Destructors: del

- **Special features of Classes-coding/Hands-on**
 - i. Inheritance: "IS-a"relationship
 - ii. Composition: "HAS-a"relationship
 - iii. Pseudo private class attributes
 - iv. Boundandun bound method objects
 - v. Class objects
 - vi. "Mix-In"classes

- **Advanced Class Topics-coding / Hands-on**
 - i. "Newstyle"class model
 - ii. Diamondinheritancechange
 - iii. MRO: Method Resolution Order
 - iv. Slots: Attribute Declarations
 - v. Properties: Attribute Accessors
 - vi. Staticand Class methods
 - vii. The "super" built-in function

- **Exception Basics-coding / Hands-on**
 - i. Why Exceptions?
 - ii. Default Exception handler
 - iii. Catching Exceptions

iv. RaisingExceptions

- **Coding Exceptions-coding / Hands-on**

- i. Thetry / except / else statement
- ii. Try / finally statement
- iii. Raise statement
- iv. Assert statement
- v. With / as context managers
- vi. Nesting Exception Handlers

- **Exception Objects**

- i. Class based exceptions
- ii. Why Exception hierarchies?
- iii. Built-in Exception Classes
- iv. Custom Exceptions

- **Regular Expressions with Python:**

- i. What are regular expressions?
- ii. Regex module in python
- iii. The match Function
- iv. The search Function
- v. Matching vs searching
- vi. Search and Replace
- vii. Meta characters , advanced patterns

- **Data Libraries:**

- i. Introduction to numpy
- ii. Creating arrays
- iii. Indexing Arrays
- iv. Array Transposition
- v. Universal Array Function
- vi. Array Processing
- vii. Array Input and Output
- viii. Introduction to Pandas, Series, Data frames
- ix. Data reading with Pandas
- x. Data cleaning with Pandas
- xi. Data wrangling with Pandas
- xii. Data selection with Pandas
- xiii. Data extraction with Pandas
- xiv. Introduction to Mat plot lib
- xv. Data Visualization with mat plot lib

- **Mathematics:**

- i. Linear Algebra
- ii. Statistics:
- iii. Probability:
- iv. Differential Calculus:

- **Machine Learning:**

- **Introduction to Machine Learning:**

- i. What is Machine Learning
- ii. Why use Machine Learning
- iii. Examples of ML applications
- iv. Types of ML Systems.
- v. Supervised Learning
- vi. Unsupervised Learning
- vii. Batch vs Online Learning
- viii. Instance-based vs Model-based Learning
- ix. Challenges of Machine Learning
- x. Over fitting vs underfitting training data
- xi. All phases of End to End ML Project.

- **Classification Models**

- i. Binary Classifier
- ii. Performance Measures
 - a) Accuracy
 - b) Confusion Matrix
 - c) Precision and Recall
 - d) ROC Curve
- iii. Multi Class Classification
- iv. Error Analysis
- v. Multi Label Classification
- vi. Multi Output Classification

- **Regression Models**

- i. Linear Regression
- ii. Gradient Descent
 - a) Batch Gradient Descent
 - b) Stochastic Gradient Descent
 - c) Mini-batch Gradient Descent
- iii. Polynomial Regression

- iv. Regularized Linear Models
 - a) Ridge Regression
 - b) Lasso Regression
 - c) Early Stopping
- v. Logistic Regression

- **Support Vector Machines:**
 - I. Linear SVM Classification
 - II. Soft Margin vs Hard Margin Classification
 - III. Nonlinear SVM Classification
 - IV. SVM Regression Models

- **Decision Trees:**
 - i. Introduction to Decision Tree
 - ii. Training Decision Tree
 - iii. Visualizing Decision Tree
 - iv. Estimating Class Probabilities
 - v. The CART Training Algorithm
 - vi. Computational Complexity
 - vii. Gini Impurity vs Entropy
 - viii. Regularization of Hyper parameters
 - ix. Regression, Instability

- **Random Forests:**
 - i. Ensemble Learning
 - ii. Voting Classifiers
 - iii. Bagging and Pasting
 - iv. Bagging & Pasting in sci-kit Learn
 - v. Out of bag valuation
 - vi. Random Patches and Random Subspaces
 - vii. Random Forests
 - viii. Extra Tree and Feature importance
 - ix. Introduction to Boosting
 - x. Ada Boost and Gradient Boost
 - xi. Stacking

- **Dimensionality Reduction:**
 - i. The Curse of dimensionality
 - ii. Approaches for dimensionality reduction
 - iii. Projection and Manifold Learning

iv. PCA

- **Unsupervised Learning Techniques**

- Clustering
- K-Means Algorithm and Limit of K-Means
- Image segmentation using Clustering
- Preprocessing using Clustering
- Semi-supervised Learning using Clustering
- DBSCAN
- Gaussian Mixtures

- **Deep Learning:**

- **Introduction to Artificial Neural Networks with Keras**

- Biological Neuron
- The Perceptron
- Multilayer Perceptron and Back propagation
- Regression MLPs
- Classification MLPs
- Implementing MLPs with Keras
- Building an Image classifier using Sequential API

- **Building Regression ML using Sequential API**

- Building complex Models using Functional API
- Saving and Restoring Model
- Using Callbacks
- Finetuning neural network hyper parameters
- Number of Hidden layers
- Number of neurons per hidden layer
- Learning rate, Batch size and other hyper parameters

- **Training Deep Neural networks**

- Vanishing / Exploding Gradients problems.
 - Glorot and He initialization
 - Non saturating Activation Functions
 - Batch Normalization
 - Gradient Clipping
- Reusing Pretrained Layers
 - Transfer Learning
 - Unsupervised Pretraining
- Faster optimizers

iv. Avoiding over fitting through Regularization

1. L1 & L2 regularization
2. Drop out
3. MC(Monte Carlo)Dropout
4. Max-Norm Regularization

• **Loading and Preprocessing Data with Tensor flow**

- I. Quick tour of Tensorflow
- II. Tensors and Operations
- III. Tensors vs numpy
- IV. The Data API
 1. Chaining Transformations
 2. Shuffling Data
 3. Preprocessing Data
 4. Prefetching Data
- V. Preprocessing Input features
 1. Encoding categorical Features using one hot vectors
 2. Encoding categorical features using Embeddings
 3. Keras Preprocessing Layers
- VI. TF Transform
- VII. TFDS Project

