



Django

Python with DJANGO [80 Hrs]

Instructor: Pondit Team

E-Mail:

Phone:

Overview

আইটি ইন্ডাস্ট্রির জন্য দক্ষ জনশক্তি তৈরিতে পন্ডিট স্পন্সরশীপ প্রদান করছে। যার পরিপ্রেক্ষিতে ইউনিভার্সিটির ছাত্রছাত্রীরা একটি পরীক্ষায় ভালো রেজাল্টের মাধ্যমে ১০০% পর্যন্ত ফ্রি কোর্স করার সুযোগ পাচ্ছেন।

পরীক্ষার মাধ্যমে আপনি ১০০% পর্যন্ত স্পন্সরশীপ পাবেন তবে আপনাকে অবশ্যই পরীক্ষায় ৮০% মার্ক্স পেতে হবে। ১০০% স্পন্সরশীপ না পেলে পরীক্ষার রেজাল্টের উপর ভিত্তি করে ৭৫% অথবা ৫০% ডিসকাউন্ট পাবেন।

কেন আমরা পাইথন শিখবো?

১. ডেটা সাইন্স: ডেটা সাইন্স এর জন্য পাইথন খুব সহজে ব্যবহারযোগ্য একটি ল্যাঙ্গুয়েজ। কেননা পাইথন এর লাইব্রেরি ও ফ্রেমওয়ার্ক গুলি (Pybrain, Numpy etc) ডেটা সাইন্স ও আর্টিফিশিয়াল ইন্টেলিজেন্স এ ব্যবহার করা সহজ।

২. মেশিন লার্নিং : বিগত কয়েক বছর ধরে মেশিন লার্নিং ও আর্টিফিশিয়াল ইন্টেলিজেন্স এর চর্চা পুরো পৃথিবী জুড়ে অনেক বেশি সমাদৃত। মেশিনকে কৃত্রিম বুদ্ধিমত্তা প্রদানের জন্য যে ইন্সট্রাকশনগুলো দেয়া প্রয়োজন তা পাইথন ল্যাঙ্গুয়েজ দ্বারা খুব সহজেই প্রোগ্রাম করা যায়।

৩. ইমেজ প্রসেসিং : ইমেজ প্রসেসিং এর জন্য সবচেয়ে সহজে ইমপ্লিমেন্টেশন করার ল্যাঙ্গুয়েজ হলো পাইথন। পাইথন এর scikit-image, NumPy, SciPy, PIL/Pillow, OpenCV, Mahotas, pgmagick, Pycairo লাইব্রেরি গুলো হায়ার লেভেল ইমেজ প্রসেসিং এর জন্য খুব সহজে ব্যবহারযোগ্য।

৪. ওয়েব ডেভেলপমেন্ট : যদিও সারাবিশ্বে ওয়েব ডেভেলপমেন্ট এর ফ্রন্ট এন্ডের জন্য HTML, CSS, Javascript এবং ব্যাক এন্ডে Laravel ও Bootstrap সবচেয়ে বেশি প্রচলিত, কিন্তু পাইথনের Django ও Flask লাইব্রেরি ও ফ্রেমওয়ার্ক ব্যবহার করে ফ্রন্ট এন্ড ও ব্যাক এন্ড দুটোই খুব সহজে ডেভেলপ করা যায়। Instagram, Pinterest, Bitbucket, and Dropbox এর মত সফটওয়্যার গুলি পাইথনকে তাদের মেইন কোডিং ল্যাঙ্গুয়েজ হিসেবে ব্যবহার করে।

৫. অটোমেশন : প্রযুক্তির উন্নতির সাথে সাথে অটোমেশন এর জন্য আমাদের চাহিদা আরো বাড়ছে। আর অটোমেশন ডিভাইস এর প্রোগ্রামিং এর জন্য সবচেয়ে সহজে ইমপ্লিমেন্টেশন করা যায় পাইথন ল্যাঙ্গুয়েজ দিয়ে।

৬. বিগ ডেটা ও IoT: বিগ ডেটা ও IoT বর্তমানের আইটি দুনিয়ায় সবচেয়ে বেশি আলোচিত একটি সেক্টর। এখানেও প্রোগ্রামিং এর প্রধান ল্যাঙ্গুয়েজ হলো পাইথন।

তাই তরুণ প্রজন্মের যারা প্রোগ্রামিং ওয়ার্ল্ডে নিজের অবস্থান শক্ত করতে ইচ্ছুক, তাদের জন্য পাইথন

Prerequisites

- আপনার একটি কম্পিউটার এবং জ্যাঞ্জো শেখার আগ্রহের প্রয়োজন হবে!
- যেকোন ভাষায় কোডিং এর কিছু অভিজ্ঞতা থাকা ভালো

Course Information

- Duration : 80 Hours
- Day : শনিবার, রবিবার এবং মঙ্গলবার
- Time : রাত ৯.০০টা থেকে ১১.০০টা পর্যন্ত
- Reg. Start : Jul 14, 2022
- Reg. End : Oct 17, 2022
- Class Start : Oct 18, 2022

শিখে ফেলা খুবই যুগোপযোগী সিদ্ধান্ত ।

Audience

- সম্পূর্ণ নতুনদের জন্য
- যারা অলরেডি পাইথন নিয়ে কাজ করছেন তারও চাইলে একটু দেখে নিতে
- যারা Big Data, Data Science এবং IOT নিয়ে কাজ করতে চান

Evaluation

কোর্সের সফল ভাবে সমাপ্তির জন্য পাইথন জ্যাঙ্গো ব্যবহার করে একটি প্রজেক্ট অবশ্যই করতে হবে।

Curriculum

1. Python

- 1.1. Python introduction
- 1.2. Why python?
- 1.3. Interpreters vs Compilers
- 1.4. Install python
- 1.5. Working with Python shell
- 1.6. Running python files
- 1.7. Integrated Development Environments
- 1.8. Values and data types
- 1.9. Variable names and keywords
- 1.10. Assigning Values to Variables
- 1.11. Numbers
- 1.12. Conditions
- 1.13. Simple if Statements
- 1.14. if-else Statements
- 1.15. if-elif-else
- 1.16. Python loops
- 1.17. While loop
- 1.18. For loop
- 1.19. Using break to Exit a Loop
- 1.20. Using continue in a Loop
- 1.21. Strings
- 1.22. Accessing Values in Strings
- 1.23. String length
- 1.24. Traversal a string with a for loop
- 1.25. Searching a char in string
- 1.26. String comparison
- 1.27. String methods
- 1.28. Lists
- 1.29. Accessing Value in a List
- 1.30. Index Positions
- 1.31. Lists are mutable
- 1.32. Traversing a list
- 1.33. List slices
- 1.34. Accessing Values in Lists
- 1.35. Updating Lists
- 1.36. Delete List Elements

- 1.37. Built-in List Functions & Methods
- 1.38. Tuples
- 1.39. Accessing Values in Tuples
- 1.40. Updating Tuples
- 1.41. Tuples Operations
- 1.42. Lists and tuples
- 1.43. Comparing tuples
- 1.44. Built-in Tuple Functions
- 1.45. Dictionaries
- 1.46. Accessing Values in Dictionary
- 1.47. Updating Dictionary
- 1.48. Looping and dictionaries
- 1.49. Dictionaries and hash map
- 1.50. Removing Key-Value Pairs
- 1.51. A Dictionary in a Dictionary
- 1.52. Functions
- 1.53. Defining a Function
- 1.54. Passing Arguments
- 1.55. Return Values
- 1.56. Classes
- 1.57. Working with Classes and Instances
- 1.58. Inheritance
- 1.59. File handling
- 1.60. Reading file
- 1.61. Writing file
- 1.62. Exceptions

2. Django

- 2.1. Django introduction
- 2.2. Basic HTTP, TCP
- 2.3. How does web works, HTTP Verbs
- 2.4. Web server
- 2.5. Necessity of web framework
- 2.6. Brief History of Django
- 2.7. Setup Django
- 2.8. Django request response cycle
- 2.9. MVC and MVT design patterns
- 2.10. Database setup
- 2.11. Create Project
- 2.12. Create App
- 2.13. Django router
- 2.14. Django models, and admin site
- 2.15. Different type of fields
- 2.16. Django ForeignKey
- 2.17. Django ManyToManyField
- 2.18. Django model save method
- 2.19. Default User Moder
- 2.20. Django Abstractuser Model
- 2.21. Model Constraints
- 2.22. Django Views
- 2.23. Function-Based Views

- 2.24. Class-Based Views
- 2.25. Generic Views
- 2.26. Django Template
- 2.27. Creating Template
- 2.28. Rendering a Template
- 2.29. Extending Templates
- 2.30. Template Tags and Filters Tags
- 2.31. Django queries
- 2.32. Creating objects
- 2.33. Retrieving objects
- 2.34. Retrieving a single object with get
- 2.35. Limiting QuerySets
- 2.36. Complex lookups with Q objects
- 2.37. Update objects
- 2.38. Bulk create
- 2.39. Bulk update
- 2.40. Django join query
- 2.41. Form Processing
- 2.42. Processing the Submission
- 2.43. Custom Validation Rules
- 2.44. Creating Forms from Models
- 2.45. File upload
- 2.46. Customize the Django Admin
- 2.47. Show fields using list_display
- 2.48. Adding Filters
- 2.49. Adding Search options
- 2.50. Make a field Read Only
- 2.51. Middleware
- 2.52. Built in middleware
- 2.53. Custom middleware
- 2.54. Third Party middleware

3. Django-Rest-Framework

- 3.1. Statelessness
- 3.2. REST
- 3.3. CORS
- 3.4. Class-based Views
- 3.5. Mixins
- 3.6. Generic class-based views
- 3.7. Serialization
- 3.8. Model Serializers
- 3.9. Validation in Serializers
- 3.10. ViewSet
- 3.11. list method
- 3.12. create method
- 3.13. retrieve method
- 3.14. update method
- 3.15. partial_update method
- 3.16. destroy method
- 3.17. Action URLs
- 3.18. GenericViewSet

- 3.19. ReadOnlyModelViewSet
- 3.20. Permission classes
- 3.21. Filtering
- 3.22. Generic Filtering
- 3.23. Filtering and object lookups
- 3.24. DjangoFilterBackend
- 3.25. OrderingFilter
- 3.26. Pagination

4. Final Project

