Introduction to Python / Django

We are adept at handling Python. The omnipotent Python/Django helps us to convert human intelligence into working web applications. Python/Django, we believe, is the most suitable language for developing web applications for humans. So let’s start learning. :)

Topics

1. Linux basic commands - 2 days
2. HTML, CSS, Javascript, Jquery - 5 Days
3. Python - 5 Days
4. Django - 6 Days
5. Django Advanced - 4 Days
6. Server & Deployment - 2 days.
7. Sample project - 12 Days

Day 1 & 2 Linux

(There will be an introductory session by the Devops)

1. Managing files and file systems

cat, cd, chmod, chown, compress, cp,find, gzip, ln, mkdir, mv, rm, rmdir, rpm, sort ,touch, umount, uncompress, uniq, unzip, zip,cat,tail,head,grep

1. Reference 1
2. Reference 2

2. Managing running programs

bg, fg, kill,, nice, ps, top, watch Reference

3. Console text editors

nano, vim Reference

4. Console Internet and network commands

ftp, host, hostname, ifconfig, netconfig, netstat, ping, scp,sftp, ssh Reference

HTML , CSS, Bootstrap, Javascript, Jquery

(Introductory session will be conducted by senior design engineer)

Day 3 HTML, HTML5

1. HTML Tutorial
2. HTML5 Basics

Day 4 CSS, Bootstrap

1. Css Tutorial
2. Bootstrap Basics
3. Bootstrap Components

Day 5 Javascript

1. Javascript Tutorial

Day 6 jQuery

1. jQuery Tutorial

Day 7 jQuery [Cont...] ,

1. jQuery Tutorial
2. **Ajax** :
   a. http://www.w3schools.com/jquery/jquery_ref_ajax.asp
   b. http://www.w3schools.com/ajax/
   c. http://www.w3schools.com/jquery/ajax_ajax.asp

Python

(Introductory session will be conducted by Python/Django engineers)

You can find example programs here : (*Every day you should try few programs here.*)

References:

- Wikibooks
- Hello Python
- Secnetix
- Python Doc

For each task you can refer Tutorials Point.

Day 8 Identifiers, data types, operations

1. Identifiers - Reserved words - Lines and indentation - Assign values to variables
2. Standard data types (Numbers, String, List, Tuple, Dictionary)
3. Operations on standard data types (Operations like access values, Updating, Deleting) -Mutable vs immutable data types.

Day 9 Operators

1. Operators - Loops(while, for) - Control statements(break, continue, pass). Python Date & Time, Files I/O (raw_input, input, open, close, write, read etc.).

Day 10 Functions

1. Functions (definition, calling, pass by value vs pass by reference) - Modules(import,
2. PYTHONPATH, packages) - Exceptions.

Day 11 Class

1. Class - Objects - Accessing attributes - Garbage collection - Inheritance - Overriding methods - Constructor.

Day 12 Advanced

1. Python Multithreaded Programming
2. Regular Expressions

Django

(Introductory session will be conducted by Python/Django engineers)

Day 13 Overview and Install

1. Get idea of django by reading below links
   a. overview
   b. design-philosophies
   c. introduction
2. Install Django in linux system - Instructions
3. create virtualenv test and install django inside it.
4. Start studying of django (part1-part6) - tutorial01

Day 14 Tutorial

1. Read https://docs.djangoproject.com/en/1.10/intro/tutorial02/
2. Read https://docs.djangoproject.com/en/1.10/intro/tutorial03/

Day 15 Tutorial

1. Read https://docs.djangoproject.com/en/1.10/intro/tutorial04/
2. Read https://docs.djangoproject.com/en/1.10/intro/tutorial05/
3. Read https://docs.djangoproject.com/en/1.10/intro/tutorial06/

Day 16 Models

1. **Models:** Model syntax | Meta options
2. **QuerySets:** Executing queries | QuerySet method reference | Lookup expressions
3. **Model instances:** Instance methods | Accessing related objects

Day 17 Migrations

1. **Migrations:** Introduction to Migrations | Operations reference
2. View Layer:**The basics:** URLconfs | View functions | Shortcuts | Decorators
3. **Class-based views:** Overview

Day 18 Template

1. Template Layer: Syntax overview
2. Forms: Overview | Forms for models
3. settings
4. admin
5. coding-style

Day 19 Django Advanced

Models

1. Managers | Raw SQL
2. Supported databases | Legacy databases
3. Optimize database access

Views

1. Built-in Views | Request/response objects
2. File objects | Managing files | Custom storage
3. Overview | Built-in middleware classes

Day 20 Django Advanced [Cont...]

1. Built-in tags and filters
2. Template API | Custom tags and filters
3. Forms for models | Customizing validation
4. Localization Overview
5. Authentication, Caching

Day 21 Django Advanced [Cont...]

1. Sending emails, Pagination, Flatpages, Signals
2. Logging
3. Adding custom commands
4. Follow pep8 standard
5. sitemap.xml
6. robots.txt
7. SEO friendly urls creation

Day 22 to 35 Sample project (14 days)

Project 1: News site

Normal news site, admins can add news from backend and end users can view news.

Features:

1. Home page with latest news, menu for different category of sites
2. View all news site with pagination
3. News detail page
4. Users can register/ login to the news site.
    a. Facebook login, google login, email login
5. Profile edit should be implemented
6. Password change
7. Forget password
8. Logged in users can comment in the news detail page(Should be implemented using Ajax)
9. End user can search the news using search box
10. In the news detail page end users can share news using social platform facebook, Twitter, google plus, linkedin
11. Sitemap.xml and robots.txt should be implemented
12. Contactus
13. Newsletter

Common guidelines

1. For code version use git and bitbucket
2. Use bootstrap design(Site should be responsive)
3. The development progress should be hosted to heroku on weekly basis
4. The system should be tested with more than 100's of database items