



Python and Django Web App Development

Course description

In this course you will learn how to Create a fully functional web site using the Full-Stack with Django 1.11, how to use HTML to create website content, Use CSS to create beautifully styled sites, how to take advantage of Bootstrap to quickly style sites, Use JavaScript to interact with sites on the Front-End, Learn how to use jQuery to quickly work with the DOM, Understand HTTP requests, Create fantastic landing pages, Learn the power of Python to code out your web applications, Use Django as a back end for the websites, Implement a full Models-Views-Templates structure for your site

Python and Django Web App Development online training curriculum

Getting Started

- Lesson Objective: The Interactive Interpreter
- Lab exercise- Interactive Interpreter
- Getting Started Follow up

Simple Functions and Test Driven Labs

- Lesson Objective
- Writing simple functions
- **Lab exercise**
- Completing Test Driven Labs
- **Lab exercise**
- Test Driven Labs Follow up

Types and Variables

- Types
- Strings
- Integers
- Integer Division
- Floats
- Complex

Variables

- Defining
- Dynamic Typing
- Strong Typing
- Managing your types
- Internals
- **Lab exercise**
- Types and Variables Follow-up

Simple Expressions

- Boolean Evaluation
- Truthiness
- Branching (if / elif / else)
- Block Structure and Whitespace
- **Lab exercise**
- Simple Expressions Follow-up

Advanced: Containers

- Lists
- **Lab Exercise**

➤ Strings Revisited

Lab Exercise

➤ Tuples

Lab Exercise

➤ Dictionaries

Lab Exercise

➤ Sets

Lab Exercise

➤ Collection Transitions

Lab Exercise

➤ Advanced Types Follow-up: A Bit More Iteration

Lab Exercise

➤ Break and continue

Lab Exercise

➤ Loop-Else

- **Lab Exercise**
- More Iteration Follow-up

Functions

- Defining
- Arguments
- Mutable Arguments and Binding of Default Values
- Accepting Variable Arguments
- Unpacking Argument Lists
- Scope
- Functions Follow-up

Exceptions

- Basic Error Handling
- Raising and re-raising exceptions
- Exceptions Follow-up

Code Organization

Namespaces

- Importing modules

Creating Modules

- Preventing execution on import
- Code Organization Follow-up

Working with Files

- File I/O
- Iteration
- Context managers
- Duck Typing
- Working with Files Follow-up

Interacting with the Outside World

- Interacting with the Outside World Follow-up
- Regular Expressions
- Re-module
- Regular Expressions Follow-up

Functional Programming

- Functions as Objects
- Higher-Order Functions
- Sorting
- An Example of Higher-Order Functions

➤ Anonymous Functions

➤ Nested Functions

Closures

- Lexical Scoping
- Useful Function Objects i.e. Operator

Decorators

- Functional Programming Follow-up

➤ Advanced Iteration

➤ List Comprehensions

Generator Expressions

- Generator Functions
- Iteration Helpers i.e. itertools
- chain()
- izip()
- Advanced Iterations Follow-up

➤ Debugging Tools

➤ Logging

➤ PPrint

pdb

- Debugging Tools Follow-up

Django

Introduction to Django

- What is Django?
- Django and Python
- Django's take on MVC: Model, View and Template
- DRY programming: Don't Repeat Yourself
- How to get and install Django
- Getting started with Django
- About the 3 Core Files:
 - models.py
 - urls.py
 - views.py
- Setting up database connections
- Managing Users & the Django admin tool
- Installing and using 'out of the box' Django features

Django Forms

- Form classes
- Validation
- Authentication
- Advanced Forms processing techniques

➤ Django & REST APIs

➤ Django REST framework

➤ Django-piston