

**Introduction to Python and Django**

**Course Number:** DGO-106
**Duration:** 5 days

**Overview**

Accelebrate's Introduction to Python and Django training course starts by teaching students the basics of Python, then moves on to teach students how to develop Web applications using the Django framework. This class covers the basics of creating basic applications using the MVC (model-view-controller) design pattern, as well as more advanced topics such as administration, session management, authentication, and automated testing.

**Note:**This course can be taught with Django 4.x or later on compatible versions of Python (as documented in the earliest release notes for your general Django release [1.11, 2.0, etc.]. See the [Django Release Notes](https://docs.djangoproject.com/en/2.0/releases/) here.

**Prerequisites**

Students should already be comfortable using the operating system (Linux, Unix, Windows, Solaris, macOS, etc.) on which they will be running Python. While not mandatory, basic skills with at least one other programming language are desirable. All students should have a working knowledge of HTML5 and CSS.

**Materials**

All attendees receive comprehensive course materials.

**Software Needed on Each Student PC**

* Python 3.x installed (Anaconda bundle recommended)
* Django 4.x or later framework
* Other Python modules installed
* A text editor or IDE (PyCharm Community Edition is an excellent free editor if you don’t have a strong preference)

**Objectives**

* Develop full-stack web sites based on content stored in an RDMS
* Use python data types appropriately
* Define data models
* Understand the architecture of a Django-based web site
* Create Django templates for easy-to-modify views
* Map views to URLs
* Take advantage of the built-in Admin interface
* Provide HTML form processing

**Outline**

* Introduction
* The Python Environment
	+ Starting Python
	+ Using the interpreter
	+ Running a Python script
	+ Python scripts on Unix/Windows
	+ Editors and IDEs
* Getting Started
	+ Using variables
	+ Built-in functions
	+ Strings
	+ Numbers
	+ Converting among types
	+ Writing to the screen
	+ Command line parameters
* Flow Control
	+ About flow control
	+ White space
	+ Conditional expressions
	+ Relational and Boolean operators
	+ While loops
	+ Alternate loop exits
* Lists and Tuples
	+ About sequences
	+ Lists and list methods
	+ Tuples
	+ Indexing and slicing
	+ Iterating through a sequence
	+ Sequence functions, keywords, and operators
	+ List comprehensions
	+ Nested sequences
* Working with Files
	+ File overview
	+ The with statement
	+ Opening a text file
	+ Reading a text file
	+ Writing to a text file
	+ Raw (binary) data
* Dictionaries and Sets
	+ About dictionaries
	+ Creating dictionaries
	+ Iterating through a dictionary
	+ About sets
	+ Creating sets
	+ Working with sets
* Functions
	+ About sequences
	+ Function parameters
	+ Global variables
	+ Global scope
	+ Returning values
	+ Sorting data
* Errors and Exception Handling
	+ Syntax errors
	+ Exceptions
	+ Using try/catch/else/finally
	+ Handling multiple exceptions
	+ Ignoring exceptions
* Using Modules
	+ The import statement
	+ Module search path
	+ Zipped libraries
	+ Creating Modules
	+ Function and Module aliases
* Classes
	+ About o-o programming
	+ Defining classes
	+ Constructors
	+ Instance methods and data
	+ Class/static methods and data
	+ Inheritance
* Django Architecture
	+ Sites and apps
	+ Shared configuration
	+ Minimal Django layout
	+ Built-in flexibility
* Configuring a Project
	+ Executing manage.py
	+ Starting the project
	+ Database setup
	+ The development server
* Adding an Application
	+ Generate the application files
	+ Defining models
	+ Related objects
	+ SQL Migration
	+ App configuration
	+ Accessing models
* Login for Nothing and Admin for free
	+ Setting up the admin user
	+ Running the admin site
	+ Tweaking the admin interface
	+ Changing the admin index page
* Basic Views (AKA Controllers)
	+ What is a view
	+ HttpResponse
	+ URL route configuration
	+ Shortcut: get\_object\_or\_404()
* Basic Views (AKA Templates)
	+ About templates
	+ Django template syntax
	+ Static files
	+ Loading templates
	+ The url tag
	+ Shortcut: template.render()
* Querying the Models
	+ QuerySets
	+ Field lookups
	+ Chaining filters
	+ Slicing QuerySets
	+ Related fields
	+ Q objects
* Working with Templates
	+ Variable lookups
	+ Comments
	+ Inheritance
	+ Filters
	+ Escaping HTML
	+ Custom filters
* Forms
	+ Forms overview
	+ GET and POST
	+ The Form class
	+ Processing the form
	+ Widgets
	+ Validation
	+ Forms in templates
	+ Beyond the basics
* Automated Testing
	+ Why create tests?
	+ When to create tests
	+ Using Django's test framework
	+ Using the test client
	+ Running tests
	+ Checking code coverage
* Conclusion