

**Advanced Django**

**Course Number:** DGO-102
**Duration:** 2 days

**Overview**

Accelebrate's Advanced Django training teaches existing Python developers how to create more sophisticated Web applications using the popular Django framework. This course covers advanced topics such as administration, session management, authentication, and caching.

**Note:** This course can be taught with 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.] at <https://docs.djangoproject.com/en/2.0/releases/>).

**Prerequisites**

Students should already have a working knowledge of Python, HTML5, and CSS. Students should have some knowledge of Django.

**Materials**

All attendees receive comprehensive course materials covering all topics in the course.

**Software Needed on Each Student PC**

* Python 3.x installed (Anaconda bundle recommended)
* Django framework (pip install django)
* Django REST framework (pip install djangorestframework)
* Bootstrap framework (available from [http://getbootstrap.com](http://getbootstrap.com/))
* A text editor or IDE (PyCharm Community Edition is an excellent free editor if you don’t have a strong preference)

**Objectives**

* Take advantage of the built-in Admin interface
* Extend the template system
* Perform advanced database queries
* Implement documentation with Sphinx
* Create testable, reusable apps
* Migrate data
* Provide a RESTful interface to Django models
* Build responsive web applications that adapt dynamically to clients ranging from desktop computers with large screens to tablets and phones
* Cache requests to improve throughput

**Outline**

* Introduction
* Beyond Simple Queries
	+ Using fields in queries
	+ Executing raw SQL
	+ Non-Django DBMS access
* Enhancing Models
	+ Custom methods
	+ Complex relationships
	+ Overriding standard methods
* Advanced Template Usage
	+ Multiple inheritance
	+ Alternate delimiters
	+ Creating custom filters
	+ Automatic HTML escaping
	+ Template loading
* Session Management
	+ Enabling sessions
	+ Types of session backends
	+ Session cookies
	+ Accessing sessions from views
* Documentation with Sphinx
	+ Description of Sphinx
	+ reStructured Text
	+ Configuring sphinx
	+ Creating doc files
	+ Using autodoc
	+ Generating docs
* Creating Reusable Apps
	+ Packaging
	+ Choosing an appropriate name
	+ Deploying
	+ Using setuptools
* Migrations
	+ Managed vs unmanaged tables
	+ Handling schema changes
	+ Using makemigrations and migrate
	+ Tweaking migrations
	+ Migrating data
* Serving Data with a RESTful Interface
	+ The Django REST framework
	+ Serialization
	+ Requests and Responses
	+ Function-based views
	+ Class-based views
	+ What about django-tastypie?
* Building Responsive Web Sites
	+ Choosing a JS library
	+ Creating API endpoints
	+ Fixing template delimiter conflicts
* Using the Cache
	+ Types of caches
	+ Setting up the cache
	+ Per-site and per-view caching
	+ Low-level API
	+ Cache security
* Conclusion