

Course duration

- 48.0 hours

Course Benefits

- Learn how to install Django in a virtual environment.
- Create your first Django website.
- Learn to use Git and GitHub to manage Django projects.
- Learn to work with Django models that map to the database.
- Learn how to create template pages using Django's template language.
- Learn to create your own custom template filters and tags.
- Learn to manage static files (e.g., CSS, JS, and images) in Django.
- Learn to create users using a custom user model.
- Learn to work with Django's built-in admin site.
- Learn to send emails with Django.
- Learn to process and validate forms.
- Deep-dive into Django models.
- Learn to handle user registration and authentication.
- Learn to add feedback messages to let users know about changes.
- Learn to allow users to securely upload files to your server.
- Learn to keep user-uploaded files private.
- Learn to use Amazon Simple Storage Service (Amazon S3) to manage files.
- Learn how to make queries against the database using Django.
- Learn to add custom properties to models.
- Learn to use Ajax to securely make changes to the database without refreshing the page.
- Learn to create constraints to control the data entered into the database.
- Learn to load data with fixtures.
- Learn to add pagination to Django templates.
- Learn to find optimization opportunities the Django Debug Toolbar.
- Learn to speed up your queries by prefetching data.
- Learn to deploy your Django website to Heroku.
- Learn to customize Django admin for staff users.

Available Delivery Methods

Self-Paced

Learn at your own pace with 24/7 access to an On-Demand course.

Course Outline

1. Setting Up Your Computer
 1. Demo and Exercise Files
 2. Recommended Software
 3. Running Python
2. Getting Started with Django
 1. Welcome to the Server-side
 2. Google Chrome DevTools: Network Tab
 3. Creating the Virtual Environment
 4. Exercise: Creating a Virtual Environment and Installing Django
 5. Exploring the Django Source Code
 6. Creating a New Project
 7. Exercise: Hello, Django!
 8. Settings
 9. Django Apps
 10. Creating a New App
 11. Exercise: First Django App
 12. Exercise: Working with Templates
 13. Exercise: Creating an About Us Page
3. Git, GitHub, and an App with a Model
 1. djangojokes.com
 2. Version Control Systems, Git, and GitHub
 3. Exercise: Setting Up VS Code, Git, and GitHub
 4. Exercise: Creating the Project
 5. Exercise: Creating a pages App
 6. Exercise: Creating the jokes App
 7. Models
 8. Exercise: Migrating
 9. Exercise: Creating a Model
 10. Types of Views
 11. Exercise: Creating a ListView
 12. Exercise: Creating a DetailView
 13. GET and POST Requests
 14. Exercise: Creating and Updating Jokes
 15. Deleting Objects
 16. Exercise: Deleting Jokes
4. Template Filters and Tags
 1. Exercise: Setting Up the Project
 2. `get_context_data()`
 3. Exercise: Adding Template Context
 4. Template Filter Basics
 5. Exercise: Adding Filters
 6. Most Useful Template Filters

7. Template Filter Quick Reference
8. Template Tag Basics
9. Most Useful Template Tags
10. Template Tag Quick Reference
11. Custom Filters
12. Exercise: Creating Custom Filters
13. Custom Tags
14. Exercise: Creating Custom Tags
5. Static Files
 1. Static File Basics
 2. Exercise: Adding Static Files to Django Jokes
 3. Review of the Settings
6. Django Admin and the User Model
 1. The Database
 2. Exercise: Setting Up PostgreSQL
 3. The Default User Model
 4. Exercise: Creating a Custom User Model
 5. Referencing the User Model
 6. Exercise: Getting Started with Django Admin
 7. Exercise: Adding the jokes Table to Django Admin
 8. Exercise: Installing the Django Admin Documentation Generator
7. URLs and Slugs
 1. Slugs
 2. Exercise: Creating a Slug-generating Function
 3. Exercise: Changing Jokes to Use Slugs
8. Sending Email with SendGrid
 1. Transactional Email Services
 2. Exercise: Getting a SendGrid Account
 3. Exercise: Creating an Email Utility Function
9. Forms and Widgets
 1. Form Processing
 2. Understanding Form Fields
 3. Exercise: Creating a Job Application Form
 4. Core Field Arguments
 5. Exercise: Building out the Application
 6. Typed Choice Fields
 7. Changing Widgets
 8. Exercise: Improving the Job Application Form
 9. Validators
 10. Exercise: Crispy Forms
 11. Exercise: Making the Joke Form Crispy
10. ModelForms
 1. Models
 2. Model Fields
 3. Model Field Arguments
 4. ModelForms
 5. Exercise: Creating a ModelForm

6. Many-to-One Relationships
7. Exercise: Adding Joke Categories
8. Many-to-Many Relationships
9. Exercise: Adding Joke Tags
10. Exercise: Updating the Joke Templates
11. Exercise: Tying Job Applications to Models
11. User Registration
 1. The User-Registration Process
 2. Exercise: Getting Started with django-allauth
 3. django-allauth Templates
 4. Exercise: Making the Authentication Pages Crispy
 5. Exercise: Updating the _base.html Template
 6. Email Templates
 7. Exercise: Custom User Registration
 8. Exercise: Creating a My Account Page
 9. Exercise: Associating Users with Jokes
10. Mixins
11. Exercise: Restricting Joke Creating and Updating
12. Messages Framework
 1. Message Framework
 2. Exercise: Adding Messages
 3. Exercise: Using SuccessMessageMixin
 4. Exercise: Adding Success Messages to the Joke Views
 5. Exercise: Adding a Success Message to JokeDeleteView
13. Media Files
 1. Media File Basics
 2. Exercise: Adding Media Settings
 3. Exercise: Uploading a Resume
 4. Exercise: Checking File Type
 5. Exercise: Keeping Resume Files Private
 6. Exercise: Adding an ImageField to the Model
 7. Exercise: Displaying the Avatar
 8. Amazon Simple Storage Service (Amazon S3)
 9. Exercise: Setting Up Amazon S3
 10. Exercise: Static Settings in Development
14. Making Queries, Ajax, and View Functions
 1. Useful Prerequisites
 2. Managers and QuerySets
 3. Indexing and Slicing QuerySets
 4. Checking for Existence
 5. Creating, Updating, and Saving Model Instances
 6. QuerySet Methods that Don't Return QuerySets
 7. One-to-Many Relationships
 8. Many-to-Many Relationships
 9. Spanning Relationships
 10. Joke Voting
 11. Exercise: Adding a JokeVote Model

12. Exercise: Registering the JokeVote Model
13. Exercise: Adding Properties to the Joke Model
14. Exercise: Rating Jokes Using Ajax
15. Exercise: Adding a vote() View Function
16. Exercise: Updating the URLConf
17. Exercise: Updating the Templates
18. Model Constraints
19. Letting JavaScript Know if the User is Logged In
15. Pagination and Ordering
 1. Fixtures
 2. Exercise: Loading Data from a Fixture
 3. Exercise: Pagination
 4. Exercise: Ordering
16. Advanced Querying and Search
 1. Field Lookups
 2. Q Objects
 3. Exercise: Filtering Results by Category, Tag, or Creator
 4. Exercise: Implementing Search
 5. Aggregation
 6. Exercise: Getting Joke Ratings
 7. Code Optimization
 8. Exercise: Installing and Using the Django Debug Toolbar
 9. Brief Introduction to Annotation
 10. Viewing SQL Used in QuerySets
17. Deploying your Website to Production
 1. Where to Host Your Django Site
 2. Exercise: Deploying to Heroku
18. Customizing Django Admin
 1. Django Admin Views
 2. Exercise: Customizing Titles and Headers
 3. ModelAdmin Attributes that Affect the List View
 4. Exercise: Creating a Custom ModelAdmin Base Class
 5. Exercise: Customizing the Django Admin List View
 6. Lists vs. Tuples
 7. ModelAdmin Attributes that Affect Forms
 8. Exercise: Customizing Django Admin Form Views
 9. Customizing Fieldsets
 10. Exercise: Modifying the CustomUserAdmin Class
 11. Adding Calculated Read-only Fields
 12. Exercise: Adding a Link to the Change Password Form
 13. Exercise: Unregistering Models
 14. Exercise: Django Admin Groups and Permissions
19. Additional Topics
 1. Additional Learning
 2. Additional Projects

Class Materials

Each student will receive a comprehensive set of materials, including course notes and all the class examples.

Class Prerequisites

Experience in the following *is required* for this Django class:

- Python

Experience in the following *would be useful* for this Django class:

- HTML
- CSS
- Bootstrap

Prerequisite Courses

Courses that can help you meet these prerequisites:

- [Introduction to Python 3 Training](#)
- [Advanced Python 3 Training](#)