Part 2: Web Development with Python

Table of Contents

Introduction 1.1

Getting Started 1.2

- Python Review 1.2.1
- Development Environments 1.2.2
- SQLite 1.2.3
- pip 1.2.4
- virtualenv 1.2.5
- Web Browsers 1.2.6
- Version Control 1.2.7

Interlude: Modern Web Development 1.3

- Front-end, Back-end, and Middleware 1.3.1
- Model-View-Controller 1.3.2

Flask: Quick Start 1.4

- Overview 1.4.1
- Installation 1.4.2
- Hello World 1.4.3
- App Flow 1.4.4
- Dynamic Routes 1.4.5
- Response Object 1.4.6
- Debug Mode 1.4.7

Interlude: Database Programming 1.5

- SQL and SQLite Basics 1.5.1
- Creating Tables 1.5.2
- Inserting Data 1.5.3
- Searching 1.5.4
- Updating and Deleting 1.5.5
- Working with Multiple Tables 1.5.6

- SQL Functions 1.5.7
- Example Application 1.5.8
- SQL Summary 1.5.9
- Chapter Summary 1.5.10

Flask Blog App 1.6

- Project Structure 1.6.1
- Model 1.6.2
- Controller 1.6.3
- Views 1.6.4
- Templates 1.6.5
- Sanity Check 1.6.6
- User Login 1.6.7
- Sessions 1.6.8
- Show Posts 1.6.9
- Add Posts 1.6.10
- Style 1.6.11
- Conclusion 1.6.12

Interlude: Debugging in Python 1.7

- Workflow 1.7.1
- Breakpoints 1.7.2
- Post Mortem Debugging 1.7.3

Flask: FlaskTaskr, Part 1 - Quick Start 1.8

- Getting Started 1.8.1
- Configuration 1.8.2
- Database 1.8.3
- Controller 1.8.4
- Templates and Styles 1.8.5
- Sanity Check 1.8.6
- Tasks 1.8.7
- Add, Update, and Delete Tasks 1.8.8

Tasks Template 1.8.9
Add Tasks Form 1.8.10
Sanity Check 1.8.11
Flask: FlaskTaskr, Part 2 - SQLAlchemy and User Management 1.9
Database Management 1.9.1
User Registration 1.9.2
Authentication 1.9.3
Database Relationships 1.9.4
Managing Sessions 1.9.5
Flask: FlaskTaskr, Part 3 - Error Handling and Testing 1.10
Error Handling 1.10.1
Testing 1.10.2
Getting Started 1.10.3
Interlude: Intro to HTML and CSS 1.11
HTML 1.11.1
CSS 1.11.2
Chrome Developer Tools 1.11.3
Flask: FlaskTaskr, Part 4 - Styles, Test Coverage, and Permissions 1.12
Templates and Styling 1.12.1
Test Coverage 1.12.2
Nose Testing Framework 1.12.3
Permissions 1.12.4
Flask: FlaskTaskr, Part 5 - Blueprints 1.13
What are Blueprints? 1.13.1
Refactoring 1.13.2
Flask: FlaskTaskr, Part 6 - New features and Error Handling 1.14
New Features 1.14.1
Password Hashing 1.14.2
Custom Error Pages 1.14.3
Flask: FlaskTaskr, Part 7 - Deployment 1.15

- Deployment 1.15.1
- Automated Deployments 1.15.2

Flask: FlaskTaskr, Part 8 - RESTful API 1.16

- Building a RESTful API 1.16.1

Interlude: Flask Boilerplate Template and Workflow 1.17

Flask: FlaskTaskr, Part 9: Continuous Integration and Delivery 1.18

- Workflow 1.18.1
- Continuous Integration Tools 1.18.2
- Travis CI Setup 1.18.3
- Intermission 1.18.4
- Feature Branch Workflow 1.18.5
- Fabric 1.18.6
- Recap 1.18.7
- Conclusion 1.18.8

Flask: Behavior-Driven Development with Behave 1.19

- Behavior-Driven Development 1.19.1
- Project Setup 1.19.2
- Introduction to Behave 1.19.3
- Feature Files 1.19.4
- First Feature 1.19.5
- Second Feature 1.19.6
- Third Feature 1.19.7
- Conclusion 1.19.8
- Update Steps 1.19.9

Interlude: Web Frameworks, Compared 1.20

- Overview 1.20.1

web2py: Quick Start 1.21

- Installation 1.21.1
- Hello World 1.21.2
- Deploying on PythonAnywhere 1.21.3

seconds2minutes App 1.21.4
Interlude: APIs 1.22
Retrieving Web Pages 1.22.1
Web Services Defined 1.22.2
Working with XML 1.22.3
Working with JSON 1.22.4
Working with Web Services 1.22.5
My API Films 1.22.6
web2py: Sentiment Analysis 1.23
Sentiment Analysis 1.23.1
Sentiment Analysis Expanded 1.23.2
Movie Suggester 1.23.3
web2py: py2manager 1.24
Setup 1.24.1
Database 1.24.2
URL Routing 1.24.3
Initial Views 1.24.4
Templates 1.24.5
Profile Page 1.24.6
Add Projects 1.24.7
Add Companies 1.24.8
Homepage 1.24.9
More Grids 1.24.10
Notes 1.24.11
Error Handling 1.24.12
Interlude: Web Scraping and Crawling 1.25
Libraries 1.25.1
HackerNews (scrapy.Spider) 1.25.2
Scrapy Shell 1.25.3
Wikipedia (scrapy.Spider) 1.25.4

- Socrata (CrawlSpider and Item Pipeline) 1.25.5
- Web Interaction 1.25.6

web2py: REST Redux 1.26

- Basic REST 1.26.1
- Advanced REST 1.26.2

Django: Quick Start 1.27

- Installation 1.27.1
- Hello, World! 1.27.2
- Workflow 1.27.3

Interlude: Introduction to JavaScript and jQuery 1.28

- Getting Started 1.28.1
- Handling the Event 1.28.2
- Updating the DOM 1.28.3

Django Bloggy: A blog app (part one) 1.29

- Setup 1.29.1
- Model 1.29.2
- Setup an App 1.29.3
- Django Shell 1.29.4
- Unit Tests for Models 1.29.5
- Django Admin 1.29.6
- Custom Admin View 1.29.7
- Templates and Views 1.29.8
- Friendly Views 1.29.9
- Django Migrations 1.29.10
- View Counts 1.29.11
- Styles 1.29.12
- Popular Posts 1.29.13

Django Bloggy: A blog app (part two) 1.30

- Forms 1.30.1
- Even Friendlier Views 1.30.2

- Stretch Goals 1.30.3
- Django Workflow 1.31
- Bloggy Redux: Introducing Blongo 1.32
  - MongoDB 1.32.1
  - Talking to Mongo 1.32.2
  - Django Setup 1.32.3
  - Setup an App 1.32.4
  - Add Data 1.32.5
  - Update Project 1.32.6
  - Test 1.32.7
  - Test Script 1.32.8
  - Conclusion 1.32.9
- Django: Ecommerce Site 1.33
  - Rapid Web Development 1.33.1
  - Prototyping 1.33.2
  - Project Setup 1.33.3
  - Landing Page 1.33.4
  - Bootstrap 1.33.5
  - About Page 1.33.6
  - Contact App 1.33.7
  - User Registration with Stripe 1.33.8
  - Next Steps 1.33.9
- Appendix A: Installing Python 1.34
  - Check Current Version 1.34.1
  - Install Python 1.34.2
  - Verify Install 1.34.3
- Appendix B: Supplementary Materials 1.35
  - Working with FTP 1.35.1
  - Working with SFTP 1.35.2
  - Sending and Receiving Email 1.35.3

Acknowledgements 1.36