

1. Record Nr.	UNINA9910959263403321
Autore	Newman Scott
Titolo	Django 1.0 template development : a practical guide to Django template development with custom tags, filters, multiple templates, caching, and more / / Scott Newman ; reviewers: Jan V. Smith, Dave Fregon, Patrick Chan
Pubbl/distr/stampa	Birmingham, UK, : Packt Publishing, c2008
ISBN	1-281-95348-2 9786611953485 1-84719-571-7
Edizione	[1st ed.]
Descrizione fisica	1 online resource (272 p.)
Collana	From technologies to solutions
Altri autori (Persone)	SmithJan V FregonDave ChanPatrick
Soggetti	Web sites - Authoring programs
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Bibliographic Level Mode of Issuance: Monograph
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	Intro -- Table of Contents -- Django 1.0 Template Development -- Django 1.0 Template Development -- Credits -- About the Author -- About the Reviewers -- Preface -- What this book covers -- What you need for this book -- Who this book is for -- Conventions -- Reader feedback -- Customer support -- Downloading the example code for the book -- Errata -- Piracy -- Questions -- 1. An Introduction to the Django Template System -- What are templates? -- Understanding the need for templates -- Overview of the Django template system -- Separating code from presentation -- Helping designers and developers collaborate -- Increasing maintainability -- Template syntax -- Modularity and reusability -- Flexibility -- Limitations -- Critics of the system -- Exploring how Django handles requests -- Understanding the template system syntax -- Context variable -- Variables -- Filters -- Tags -- Comments -- Code note: Python dictionaries -- How invalid variables are handled -- Creating our demo application -- Starting our application -- Adding templates to our application -- Adding variables to the view -- Moving the logic into a

separate template file -- Using template filters -- Using template tags to perform logical tests -- Adding comments -- Summary -- 2. Views, URLs, and Generic Views -- An overview -- Creating the application -- Create the data model -- Create the admin file -- Configure the URLs -- Add data in the admin application -- Mapping URLs to views -- Handling unmatched URL patterns -- Splitting up the URL configurations -- Creating views -- Accepting the request object -- Responding with an HTTP response -- Responding with an exception -- Putting the views together -- Building the basic view -- Cleaning up the error handling -- Adding the template files -- Adding the template to the view -- Creating the list view and template.

Using generic views to shorten development time -- Simple generic views -- Loading a template directly -- Redirecting URLs -- List/detail generic views -- Replacing the list view -- Replacing the detail view -- Using the other generic views -- Comparing views and generic views -- Summary -- 3. Template Context -- The context explained -- Practicing working with the context -- Using locals for prototyping -- Adding, changing, and removing items in the context -- Using the context values in your templates -- -- Preventing method execution from templates -- Handling invalid context variables -- Cleaning up the view -- Context rendering shortcuts -- Using `render_to_response()` -- Using `render_to_string()` -- Context processors -- Exploring the default context processors -- Auth -- Debug -- Media -- `il8n` -- Configuring your project to use context processors -- Configuring your views to use context processors -- Using `render_to_response` with `RequestContext` -- Using the context processors in our project -- Writing your own context processor -- Summary -- 4. Using the Built-In Tags and Filters -- Built-in filter reference -- `add` -- `addslashes` -- `capfirst` -- `center` -- `cut` -- `date` -- `default` -- `default_if_none` -- `dictsort` -- `dictsortreversed` -- `divisibleby` -- `escape` -- `escapejs` -- `filesizeformat` -- `first` -- `fix_ampersands` -- `floatformat` -- `force_escape` -- `get_digit` -- `iriencode` -- `join` -- `last` -- `length` -- `length_is` -- `linebreaks` -- `linebreaksbr` -- `linenumbers` -- `ljust` -- `lower` -- `make_list` -- `phone2numeric` -- `pluralize` -- `pprint` -- `random` -- `removetags` -- `rjust` -- `safe` -- `slice` -- `slugify` -- `stringformat` -- `striptags` -- `time` -- `timesince` -- `timeuntil` -- `title` -- `truncatewords` -- `truncatewords_html` -- `unordered_list` -- `upper` -- `urlencode` -- `urlize` -- `urlizetrunc` -- `wordcount` -- `wordwrap` -- `yesno` -- Built-in tag reference -- `autoescape` -- `block` -- `comment` -- `cycle` -- `debug` -- `extends` -- `filter` -- `firstof` -- `for` -- `forloop` -- `if` -- `ifchanged` -- `ifequal` -- `ifnotequal` -- `include` -- `load` -- `now` -- `regroup` -- `spaceless` -- `ssi` -- `templatetag` -- `url` -- `widthratio` -- `with` -- Summary -- 5. Loading and Inheriting Templates -- Configuring the template system -- Finding a home for the template files -- Working with the template loaders -- Loading templates manually -- Choosing a template loader -- Using the filesystem loader -- Using the application directories loader -- About the eggs template loader -- Using the loaders together -- Loading your template files -- Setting up the error handling templates -- Creating the error templates -- Testing the error templates -- Breaking templates into reusable pieces -- Extending templates with inheritance -- Using the block tag -- Extending templates -- Adding inheritance to the press application -- Using multiple block tags -- Adding template inheritance to our press release list -- Inheriting from multiple child templates -- Appending to blocks -- Template strategy -- Creating content placeholders -- Extra JS -- Extra style -- Extra head content -- Extra body tag attributes -- Using include files -- Using include -- Using SSI -- Summary -- 6. Serving Multiple Templates -- Considering the different approaches --

Serving mobile devices -- Adapting content -- Setting up our example -- Serving printable pages -- Creating site themes -- Testing the template overrides -- Serving different templates by domain name -- Serving different sites with the development web server -- Redirecting users to the mobile site (optional) -- Detecting mobile devices -- Writing the middleware -- Checking only once -- Installing the middleware -- Summary -- 7. Custom Tags and Filters -- Examining the built-in tags and filters -- Template filters -- Template tags -- Writing your own template filters.

Setting up a test application -- Creating a home for our filter library -- Template filter syntax -- Loading template libraries -- U.S. currency filter -- Replace profanities filter -- Filters that expect strings -- In-list filter -- Writing your own template tags -- Creating another sample application -- Adding the template library -- Template tag syntax -- A simple tag example -- The compilation function -- The template node subclass -- Registering our custom tag -- All work and no play tag -- Passing a template variable to a tag -- Modifying the context through a tag -- Summary -- 8. Pagination -- An Overview -- Verifying our application setup -- Verifying the application -- Verifying the configuration -- Verifying the URL configuration -- Verifying the model -- Verifying the view -- Adding test records -- Exploring pagination using the Django shell -- Examining database performance -- Allowing for empty result sets -- Preventing orphaned records -- Using pagination in your views -- Creating the view -- Retrieving the current position from the URL -- Putting navigation into the templates -- Pagination with generic views -- Setting up our generic list view -- Generically calling the last page -- Summary -- 9. Customizing the Admin Look and Feel -- Overriding the admin templates -- Leveraging the template loader -- Locating the admin template files -- Exploring the admin template files -- Inspecting the base.html template -- Inspecting the base_site.html template -- Inspecting the index.html template -- Inspecting the change_list.html template -- Inspecting the change_form.html template -- Customizing the admin header -- Replacing the page title -- Changing the header text -- Adding a new link box to the admin -- Overriding the admin index file -- Creating the include file -- Customizing the admin color scheme -- Identifying styles to change.

Using the extrastyle block -- Summary -- 10. Caching Your Pages -- An overview -- Do you need caching? -- How caching works -- Exploring the available cache systems -- Filesystem caching -- Database caching -- Memcached -- Local memory caching -- Dummy caching -- Setting up your cache system -- Configuring the cache backend -- Database caching -- Filesystem caching -- Local memory caching -- Dummy caching -- Memcached -- Adding additional backend arguments -- Setting up for the examples -- Caching individual views -- Adding caching -- Caching pieces of templates -- Low-level caching -- Caching your whole site -- Preventing data from being cached -- General caching strategies -- Working with outside caches -- Summary -- 11. Internationalization -- Exploring i18n -- Creating an example application -- Configuring your project for i18n -- Installing libraries for i18n translation -- Marking strings as translatable -- Creating message files -- Enabling automatic language preference -- How Django determines language preference -- Summary -- Index.

Sommario/riassunto

This book is designed for readers who learn by doing and employs many examples and screenshots to let the reader dig in and start coding. This book isn't designed to be a reference; instead it has a practical, example-driven approach that teaches you by following along

with the examples in the chapters. When you have completed this book, you will fully understand how the template system works, how to extend it when you have specialized needs, and how to optimize the performance and usability of your content. This book is for web developers and template authors who want to fully understand and utilize the Django template system. The reader should have completed the introductory tutorials on the Django project's website and some experience with the framework will be very helpful. Basic knowledge of Python and HTML is assumed.
