







DJANGO 1.5 CHEATSHEET


Model fields	Common model field options	<div> Grove.io Enterprise chat, backed by IRC, built by Revsys. http://grove.io/</div>	ModelAdmin options & callbacks
AutoField BigIntegerField BooleanField , NullBooleanField CharField max_length = 100 CommaSeparatedIntegerField max_length = 50 DateTimeField , DateTimeField DecimalField max_digits = 10 decimal_places = 2 EmailField max_length = 75 FileField upload_to = "files/%Y/%m/%d" storage = file_storage_obj max_length = 100 FilePathField path = "/var/images" match = r"\\.pdf\$" recursive = True max_length = 100 FloatField FileField upload_to = "/uploads" storage = file_storage_obj max_length = 100 ImageField upload_to = "/uploads" storage = file_storage_obj height_field = "field_name" width_field = "field_name" max_length = 100 IntegerField , PositiveIntegerField , PositiveSmallIntegerField IPAddressField GenericIPAddressField protocol = "both" "IPv4" "IPv6" unpack_ipv4 = False SlugField max_length = 100 SmallIntegerField TextField TimeField URLField	<div>null = False blank = False choices = [(1, "Choice 1"), ...] db_column = "column_name" db_index = False db_tablespace = "tablespace_name" default = value_or_callable error_messages = {"name": "Message", ...}, help_text = "long help text" primary_key = False unique = False unique_for_date = "date_field" unique_for_month = "month_field" verbose_name = "Field Name" validators = [validator_obj, ...]</div> <div>Model Meta options</div> <div>abstract = False app_label = "applabel" db_table = "table_name" db_tablespace = "space_name" get_latest_by = "field_name" index_together = [(<i>"field1"</i>, <i>"field2"</i>), ...] managed = False order_with_respect_to = "field" ordering = [<i>"field_name"</i>, ...] permissions = [(<i>"code"</i>, <i>"text label"</i>), ...] proxy = False unique_together = [(<i>"field1"</i>, <i>"field2"</i>), ...] verbose_name = "verbose name" verbose_name_plural = "plural verbose names"</div> <div>ForeignKey on_delete options</div> <div>CASCADE cascades; default PROTECT prevent related deletion SET_NULL Set to NULL SET_DEFAULT Set to field's default SET(value) / SET(callback) Set to value / callback() DO_NOTHING No action (in Python)</div> <div>Model/Form field validators</div> <div>MaxLengthValidator(max_length) MinLengthValidator(min_length) MaxValueValidator(max_value) MinValueValidator(min_value) RegexValidator(regex, message=None, code=None) URLValidator(verify_exists=False) validate_email validate_slug validate_ipv4_address (also <i>_ipv6_</i>, <i>_ipv46_</i>)</div>	<div><div><div>Chainable</div><div><div>all()</div><div>annotate(**annotations)</div><div>dates(field, "year/month/day", "ASC/DESC")</div><div>defer(*fields)</div><div>distinct(*fields) (1.4)</div><div>exclude(**lookups)</div><div>filter(**lookups)</div><div>none()</div><div>only(*fields)</div><div>order_by(*fields)</div><div>prefetch_related('field1', 'field2', ...)</div><div>raw(sql, params)</div><div>reverse()</div><div>select_for_update(nowait=False)</div><div>select_related('field', 'field2', ..., depth=1)</div><div>using("db_alias")</div><div>values(*fields)</div><div>values_list(*fields)</div><div>values_list(field, flat=True)</div></div><div><div>Not chainable</div><div><div>aggregate(**aggregations)</div><div>bulk_create(list_of_objects)</div><div>count()</div><div>create(**attributes)</div><div>delete()</div><div>exists()</div><div>get(**lookups)</div><div>get_or_create(**attributes, defaults={})</div><div>in_bulk(id_list)</div><div>iterator()</div><div>latest(field)</div><div>update(**attributes)</div></div></div></div><div>Lookups</div><div>exact, iexact = "string" contains, icontains = "string" startswith, endswith, istartswith, iendswith = "string" in = list / queryset gt, lt, gte, lte = date / int / float / decimal range = (lower, upper) year, month, day = int isnull = bool regex, iregex = r"^(?P<reg>\\. exp\\.)"</div><div>Aggregation/annotation functions</div><div>Avg(field) StdDev(field, sample=False) Count(field, distinct=False) Sum(field) Max(field) Variance(field, sample=False) Min(field)</div></div>	<div>actions = ["method", callback, ...] actions_on_bottom = False actions_on_top = True actions_selection_counter = True add_form_template = 'add_form.html' change_form_template = 'change_form.html' change_list_template = 'changelist.html', date_hierarchy = "date_field" delete_confirmation_template = 'delete.html' delete_selected_confirmation_template = 'del.html' exclude = ["field_name", ...] fields = ["field_name", ...] fieldsets = [(<i>"Name"</i>, {<i>"fields"</i>: [<i>"field_name"</i>, ...]}), ...] filter_horizontal = ["m2m_field", ...] filter_vertical = ["m2m_field", ...] form = ModelFormClass formfield_overrides = {<i>(TextField: {<i>"widget"</i>: MyWidget}), ...</i>] inlines = [InlineClass, ...] list_display = [<i>"fieldname"</i>, callable, ...] list_display_links = [<i>"field_name"</i>, ...] list_editable = [<i>"field_name"</i>, ...] list_filter = [<i>"field_name"</i>, ...] list_max_show_all = 200 list_per_page = 100 list_select_related = False object_history_template = 'admin/history.html', ordering = [<i>"field_name"</i>] paginator = Paginator prepopulated_fields = {<i>"slug"</i>: [<i>"title"</i>], ...} radio_fields = {<i>"fk_field"</i>: admin.HORIZONTAL, ...} raw_id_fields = [<i>"fk_or_m2m_field"</i>, ...] readonly_fields = [<i>"field_name"</i>, ...] save_as = False save_on_top = False search_fields = [<i>"field_name"</i>]</div> <div>add_view(self, request, form_url="", extra_context={}) changelist_view(self, request, extra_context={}) change_view(self, request, object_id, extra_context={}) delete_model(self, request, obj) delete_view(self, request, object_id, extra_context={}) formfield_for_choice_field(self, db_field, request, **kw) formfield_for_foreignkey(self, db_field, request, **kw) formfield_for_manytomany(self, db_field, request, **kw) get_list_display(self, request) get_list_display_link(self, request, list_display) get_ordering(self, request) get_paginator(self, queryset, per_page, orphans=0, allow_empty_first_page=True) get_prepopulated_fields(self, request, obj=None) get_readonly_fields(self, request, obj=None) get_urls(self) has_add_permission(self, request) has_change_permission(self, request, obj=None) has_delete_permission(self, request, obj=None) history_view(self, request, object_id, extra_context={}) message_user(self, request, message) queryset(self, request) save_formset(self, request, form, formset, change) save_model(self, request, obj, form, change) save_related(self, request, obj=None)</div>
<div>Model relationships</div> <div>ForeignKey(OtherModel) related_name = "things" limit_choices_to = Q(foo="bar", ...) to_field = "key_field_name" on_delete = (<i>see right</i>) ManyToManyField(OtherModel) related_name = "things" limit_choices_to = Q(foo="bar", ...) to_field = "key_field_name" symmetrical = True, through = RelationshipModel db_table = "table_name" OneToOneField(OtherModel) parent_link = False related_name = "thing" limit_choices_to = Q(foo="bar", ...) to_field = "key_field_name" on_delete = (<i>see right</i>)</div>	<div><div> Read the Docs Create, host, and browse documentation. http://readthedocs.org/</div></div> <div>django.db.models.signals pre_init, post_init(sender, args, kwargs) pre_save, post_save(sender, instance, using) pre_delete, post_delete(sender, instance, using) m2m_changed(sender, instance, action, reverse, model, pk_set, using) class_prepared(sender) post_syncdb(sender, app, created_models, verbosity, interactive) django.core.signals request_started, request_finished(sender) got_request_exception(sender, request)</div>	<div>django.test.signals settings_changed(sender, setting, value) template_rendered(sender, template, context) django.db.backends.signals connection_created(sender, connection) django.contrib.auth.signals user_logged_in, user_logged_out(sender, request, user) user_login_failed(sender, credentials) django.contrib.comments.signals comment_will_be_posted(sender, comment, request) comment_was_posted(sender, comment, request) comment_was_flagged(sender, comment, flag, created, request)</div>	<div><div> Haystack Modular search for Django. http://haystacksearch.org/</div><div><div> south Intelligent migrations. http://south.aeracode.org/</div></div></div>

DJANGO 1.5 CHEATSHEET

Common field options	
error_messages	= {"code": "Message", ...}
help_text	= "help text"
initial	= value
label	= "field label"
localize	= False
required	= True
validators	= [validator, ...]
widget	= WidgetClass
Form fields	
BooleanField, NullBooleanField	
CharField	
max_length	= 100
min_length	= 10
ChoiceField, MultipleChoiceField	
choices	= [(1, "Choice 1"), ...]
TypedChoiceField, TypedMultipleChoiceField	
choices	= [(1, "Choice 1"), ...]
coerce	= callback(value)
empty_value	= ""
ModelChoiceField, ModelMultipleChoiceField	
queryset	= Model.objects.all()
empty_label	= u"-----"
DateField	
input_formats	= ["%Y-%m-%d", ...]
DateTimeField	
input_formats	= ["%Y-%m%d %H:%M", ...]
DecimalField	
max_value	= Decimal(100)
min_value	= Decimal(10)
max_digits	= 10
decimal_places	= 2
EmailField	
FileField, ImageField	
FilePathField	
path	= "/home/images"
recursive	= True
match	= r"\.pdf"
FloatField	
max_value	= 100.0
min_value	= 10.0
IntegerField	
max_value	= 100
min_value	= 10
IPAddressField	
GenericIPAddressField	
protocol	= "both" "IPv4" "IPv6"
unpack_ipv4	= False
RegexField	
regex	= r'\w+'
max_length	= 100
min_length	= 10
SlugField	
max_length	= 100
min_length	= 10
TimeField	
input_formats	= ["%H:%M:%S", ...]
URLField	
max_length	= 100
min_length	= 10
verify_exists	= False
validator_user_agent	= "Django/1.3"
SplitDateTimeField	
input_date_formats	= ["%Y-%m-%d", ...]
input_time+_formats	= ["%H:%M:%S", ...]

Django Debug Toolbar	
 Debug better, faster. http://django.me/djdt	
Form widgets	
All widgets	
attrs	= {'class': 'fancy', ...}
CheckboxInput	
check_test	= callback(value)
CheckboxSelectMultiple	
ClearableFileInput	
DateInput	
format	= "%Y-%m-%d"
DateTimeInput	
format	= "%Y-%m-%d %H:%M"
FileInput	
HiddenInput, MultipleHiddenInput	
MultiWidget	
NullBooleanSelect	
PasswordInput	
render_value	= False
RadioSelect	
Select	
SelectDateWidget	
years	= [2010, 2011, ...]
SelectMultiple	
SplitDateTimeWidget	
Textarea	
TextInput	
TimeInput	
format	= "%H:%M:%S"

 Celery Distributed, asynchronous task queue. http://celeryproject.org/	
HttpRequest	HttpResponse
__iter__()	__init__(content="", mimetype=None, status=200, content_type=None)
body	__delitem__(header)
build_absolute_uri(path)	__getitem__(header)
COOKIES	__setitem__(header, val)
encoding	delete_cookie(key, path="/", domain=None)
GET, POST, REQUEST	flush()
FILES	has_header(header)
get_full_path()	set_cookie, set_signed_cookie(key, value, max_age=None, expires=None, path="/", domain=None, secure=None, httponly=True)
get_host()	tell()
get_signed_cookie(key)	write(content)
is_ajax()	
is_secure()	
META	
method	
path	
path_info	
read(size=None)	
readline(), readlines()	
session	
urlconf	
user	
View shortcuts (django.shortcuts)	
render(request, template, context_dict={}, context_instance=RequestContext, content_type="text/html", status=200, current_app=None)	
redirect(to, permanent=False, *args, **kw)	
get_object_or_404(Model, **lookup)	
get_list_or_404(Model, **lookup)	

Tastypie	
 Creating delicious APIs for Django apps since 2010. http://tastypieapi.org/	
Template filters	
add:"2"	pluralize
addslashes	pluralize:"es"
capfirst	pluralize:"y,ies"
center:"15"	pprint
cut:" "	random
date:"JS F Y H:i"	removetags:"span div"
default:"nothing"	rjust:"10"
default_if_none:"nothing"	safe
dictsort:"key"	safeseq
dictsortreversed:"key"	slice:"10:20"
divisibleby:"4"	slugify
escape	stringformat:"s"
escapejs	striptags
filesizeformat	time:"H:i"
first	timesince
fix_ampersands	timesince:from_date
floatformat	timeuntil
floatformat:"3"	timeuntil:from_date
force_escape	truncatecharsX:"9"
get_digit:"2"	truncatewords:"10"
iriencode	truncatewords_html:"10"
join:"/"	unordered_list
length	upper
length_is:"4"	urlencode
linebreaks	urlencode:"/+"
linebreaksbr	urlize
linenumbers	urlizetrunc:"15"
ljust:"10"	wordcount
lower	wordwrap:"20"
make_list	yesno:"yeah,no,unknown"
phone2numeric	
Template tags	
{% block name %}...{% endblock %}	
{% csrf_token %}	
{% cycle "row1" "row2" [as varname] [silent] %}	
{% debug %}	
{% extends "base.html" %}	
{% filter force_escape lower %}...{% endfilter %}	
{% firstof var1 var2 "fallback" %}	
{% for i in list [reversed] %}...{% empty %}...{% endfor %}	
{% forloop.counter %}	
{% forloop.counter0 %}	
{% forloop.revcounter %}	
{% forloop.revcounter0 %}	
{% forloop.first %}	
{% forloop.last %}	
{% forloop.parentloop %}	
{% if condition %}...{% elif condition %}...{% else %}...{% endif %}	
{% ifchanged %}...{% else %}...{% endifchanged %}	
{% include "other/template.html" [with who="Jane" ... [only]] %}	
{% load [foo bar ... from] other_library %}	
{% now "JS F Y H:i" %}	
{% regroup people by gender as gender_list %}	
{% spaceless %}...{% endspaceless %}	
{% templatetag openblock / closeblock / openvariable / closevariable / openbrace / closebrace / opencomment / closecomment %}	
{% url path.to.view arg1 arg2 arg3=v1 arg4=v2 [as the_url] %}	
{% widthratio this_value max_value 100 %}	
{% verbatim %} ... {% endverbatim %}	
{% with alpha=1 beta=2 ... %}...{% endwith %}	

Datetime formatting		
template	strftime	
a		a.m. / p.m.
A	%p	AM / PM
b		jan, feb, ...
B		
c		2008-01-02T10:30:00.000123
	%c	Fri Mar 4 16:43:23 2011
d	%d	01 - 31
D	%a	Mon, Tue, ...
e		UTC, CST (timezone name)
E		(alternate long month)
f		1, 1:30
F	%B	January, February, ...
g	%m	1 - 12
G		0 - 23
h	%l	01 - 12
H	%H	00 - 23
i	%M	00 - 59 (minutes)
j		1 - 31
	%j	001 - 365
l	%A	Monday, Tuesday, ...
L		(leap year?)
m		01 - 12
M	%b	Jan, Feb, ...
n		1 - 12
N		Jan., Feb., March, ...
o		(ISO-8601 week-numbering year)
O		+0200
P		1 a.m., noon, 2:30 p.m.
r		Thu, 21 Dec 2000 16:01:07 +0200
s	%S	00 - 59 (seconds)
S		st, nd, rd, th
t		28 - 31
T	%Z	EST, UTC, ...
u		(microseconds)
U		(unix timestamp)
w	%w	0 (Sun.) - 6 (Sat.)
W		1 - 53
	%U	01 - 53
y	%y	99
Y	%Y	1999
	x	03/04/11
	X	16:43:23
z		0 - 365
Z		(tz offset, seconds)
DATE_FORMAT		(as defined in settings)
DATETIME_FORMAT		(as defined in settings)
SHORT_DATE_FORMAT		(as defined in settings)
SHORT_DATETIME_FORMAT		(as defined in settings)
django.test.TestCase		
client_class	= django.test.client	
fixtures	= ["fix1.json", "fix2.json", ...]	
urls	= "my.app.urls"	
multi_db	= False	
assertContains, assertNotContains(resp, text, count=None, status_code=200, msg_prefix="", html=False)		
assertFieldOutput(cls, valid, invalid, field_args=None, field_kwargs=None, empty_value="")		
assertFormError(resp, form, field, errors, msg_prefix="")		
assertHTMLEqual, assertHTMLNotEqual(html1, html2, msg="")		
assertXMLEqual, assertXMLNotEqual(xml1, xml2, msg="")		
assertQuerysetEqual(qs, values, transform=repr, ordered=True)		
assertRedirects(resp, url, status_code=302, target_status_code=200, msg_prefix="")		
assertTemplateUsed, assertTemplateNotUsed(resp, tmplt)		
with self.assertNumQueries(num, func): ...		
with self.assertRaisesMessage(exe, msg): ...		
with self.settings(SETTING='new_value'): ...		

Find more details in Django's official docs:
<http://django.me/<any object here>>