to field

through

db table

parent link

to field

on delete

related name

limit choices to

symmetrical

OneToOneField(OtherModel)

Mo	del fields	Common	model field option
AutoField		null	= False
BigIntegerField		blank	= False
BooleanField, NullBooleanFiel	ld	choices	= [(1, "Choice
CharField		db_column	= "column_na
max_length	= 100	db_index	= False
-		db_tablespace default	= "tablespace
CommaSeparatedIntegerField max_length	= 50	error_messages	= value_or_ca = {"name": "l
DateField, DateTimeField	50	help_text	= "long help
		primary_key	= False
DecimalField	= 10	unique	= False
max_digits decimal_places	= 10 = 2	unique_for_date	= "date_field
·	- <u>2</u>	unique_for_month	= "month_fie
EmailField		verbose_name	= "Field Nam
max_length	= 75	validators	= [validator_c
FileField	#£1/0/\//0//0/-2"	Mode	el Meta options
upload_to	= "files/%Y/%m/%d" = file_storage_obj	abstract	= False
storage max_length	= file_storage_obj = 100	app_label	= "applabel"
	100	db_table	= "table_nam
FilePathField path	= "/var/images"	db_tablespace	= "space_nar
match	= /vai/illages = r"\.pdf\$"	get_latest_by	= "field_nam
recursive	= True	managed	= False
max_length	= 100	order_with_respect_to	= "field"
FloatField		ordering	= ["field_nam
FileField		permissions	= [("code", "t
upload_to	= "/uploads"	proxy unique_together	= False = [("field1", "i
storage	= /uploads = file_storage_obj	verbose_name	= "verbose na
max_length	= 100	verbose_name_plural	= "plural verb
ImageField			<u> </u>
upload_to	= "/uploads"	ForeignKe	ey on_delete optio
storage	= file_storage_obj	CASCADE	cascades; de
height_field	= "field_name"	PROTECT	prevent relat
width_field	= "field_name"	SET_NULL	Set to NULL
max_length	= 100	SET_DEFAULT	Set to field's
IntegerField, PositiveIntegerF	ield, PositiveSmallIntegerField	SET(value) / SET(callback)	Set to value
IPAddressField		DO_NOTHING	No action (in
GenericIPAddressField		Model/F	orm field validator
protocol	= "both" "IPv4" "IPv6"	Mandan add Mall data of an all	(1)
•	= False	MaxLengthValidator(max_le MinLengthValidator(min_le	
unpack_ipv4	- 1 alse	MaxValueValidator(max va	
SlugField	= 100	MinValueValidator(min_val	
max_length	= 100	RegexValidator(regex, mess	
SmallIntegerField		URLValidator(verify_exists=	
TextField		validate_email	
TimeField		validate_slug	
URLField		validate_ipv4_address (also	
Model	relationships	validate_comma_separated	_integer_list
ForeignKey(OtherModel)	·	Read the	Docs
related_name	= "things"		st, and browse o
limit_choices_to	= Q(foo="bar",)		dthedocs.org/
to_field	= "key_field_name"	mip.//Teu	g/
on_delete	= (see right)		
ManyToManyField(OtherMode		django.db.models.signals	
related_name	= "things"	pre_init, post_init(sen	der, args, kwargs)
limit_choices_to	= Q(foo="bar",)	pre_save, post_save(se	
	41 C.1.1	nea dalata nast dalat	

= "key field name"

= RelationshipModel

= "table name"

= Q(foo="bar", ...)

= "key field name"

= True,

= False

= "thing"

= (see right)

Read the Docs

Create, host, and browse documentation. http://readthedocs.org/



ForeignKey on_delete options

= "table_name"

= "space_name"

= "field name"

= ["field_name", ...]

= "verbose name"

cascades; default

Set to field's default

No action (in Python)

= [("code", "text label"), ...]

= [("field1", "field2"), ...]

= "plural verbose names"

prevent related deletion

Set to value / callback()

= [(1, "Choice 1"), ...]

= "tablespace name"

= {"name": "Message", ...},

= value or callable

= "long help text"

= "month field"

= "Field Name"

= [validator_obj, ...]

= "column name"

nit, post_init(sender, args, kwargs) ore_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)

request_started, request_finished(sender) got request exception(sender, request)



Find more details in Django's official docs:

http://django.me/<any object here>

QuerySet methods all() annotate(**annotations) dates(field, "year/month/day", "ASC/DESC") defer(*fields) distinct(*fields) (1.4) exclude(**lookups) filter(**lookups) none() only(*fields) order_by(*fields) prefetch_related('field1', 'field2', ...) raw(sql, params) reverse() select for update(nowait=False) select_related('field', 'field2', ..., depth=1) using("db alias") values(*fields) values list(*fields) values_list(field, flat=True) aggregate(**aggregations) bulk_create(list_of_objects) count() create(**attributes) delete() exists()

get(**lookups) get_or_create(**attributes, defaults={}) in_bulk(id_list)

iterator() latest(field) update(**attributes)

Lookups

= "string" exact, iexact contains, icontains = "string" istartswith, iendswith = "string" = list / queryset = date / int / float / decimal = (lower, upper) = int = bool $= r"^reg\. exp\."$

Aggregation/annotation functions

StdDev(field, sample=False) Count(field, distinct=False) Sum(field)

Signals

template rendered(sender, template, context)

user_logged_in, user_logged_out(sender, request, user)

comment_was_flagged(sender, comment, flag,

created, request)

Modular search for Django. Haystack http://haystacksearch.org/



Intelligent migrations. http://south.aeracode.org/

django.test.signals

settings changed(sender, setting, value)

diango.db.backends.signals

connection_created(sender, connection)

django.contrib.auth.signals

diango.contrib.comments.signals

comment will be posted(sender, comment, request) comment_was_posted(sender, comment, request)

dator(regex, message=None, code=None) year, month, day regex, iregex

in

startswith, endswith,

gt, lt, gte, lte

range

Avg(field)

Max(field) Min(field)

Variance(field, sample=false)

get_urls(self)

delete_model(self, request, obj)

get_list_display(self, request)

get_ordering(self, request)

get_prepopulated_fields(self, request, obj=None) get_readonly_fields(self, request, obj=None)

ModelAdmin options & callbacks

delete_confirmation_template = 'delete.html'

= {TextField: {"widget": MyWidget}, ...}

radio fields = {'fk field': admin.HORIZONTAL, ...}

add_view(self, request, form_url=", extra_context={})

change_view(self, request, object_id, extra_context={})

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)

changelist view(self, request, extra context={})

get_list_display_link(self, request, list_display)

get_paginator(self, queryset, per_page, orphans=o,

allow empty first page=True)

delete_selected_confirmation_template = 'del.html'

= [("Name", {"fields": ["field_name", ...]}), ...]

= False

= True

= True

= ["method", callback, ...]

= 'add form.html'

= 'change form.html'

= 'changelist.html'.

= ["field_name", ...]

= ["field name", ...]

= ["m2m_field", ...]

= ["m2m field", ...]

= ModelFormClass

= [InlineClass, ...]

= ['field name', ...]

= ['field name', ...]

= ['field name', ...]

= ['field name']

= {'slug': ['title'], ...}

= ['field_name', ...]

= ['field name']

= ['fk or m2m field', ...]

= Paginator

= 'admin/history.html',

= 200

= 100

= False

= False

= False

= ["fieldname", callable, ...]

= "date field"

actions

actions on bottom

add form template

change form template

change list template

actions_selection_counter

actions on top

date hierarchy

filter horizontal

formfield overrides

list display links

list_max_show_all

list select related

prepopulated_fields

object_history_template

filter vertical

exclude

fieldsets

fields

inlines

list display

list editable

list_per_page

list filter

ordering

paginator

save_as

raw id fields

save on top

search fields

readonly_fields

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)

quervset(self, request)

save formset(self, request, form, formset, change) save_model(self, request, obj, form, change)

save_related(self, request, obj=None)



Brought to you by

DJANGO 1.4 CHEATSHEE

Common	field options
error_messages	= {"code": "Message",}
help_text	= "help text"
initial	= value
label localize	= "field label" = False
required	= raise = True
validators	= [validator,]
widget	= WidgetClass
Forn	n fields
BooleanField, NullBooleanField	d
CharField	
max_length	= 100
min_length	= 10
ChoiceField, MultipleChoiceFie	= [(1, "Choice 1"),]
TypedChoiceField, TypedMultip	
choices	= [(1, "Choice 1"),]
coerce	= callback(value)
empty_value	= ""
ModelChoiceField, ModelMulti	•
queryset empty_label	= Model.objects.all() = u""
DateField	– u
input_formats	= ["%Y-%m-%d",]
DateTimeField input_formats	= ["%Y-%m%d %H:%M",]
DecimalField	
max_value	= Decimal(100)
min_value max_digits	= Decimal(10) = 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	= 100
IPAddressField	
GenericIPAddressField	
protocol	= "both" "IPv4" "IPv6"
unpack_ipv4	= False
RegexField	
regex	= r'\w+'
max_length	= 100
min_length	= 10
SlugField	
max_length min_length	= 100 = 10
	- 10
TimeField input_formats	= ["%H:%M:%S",]
URLField	,
max_length	= 100
min_length	= 10
verify_exists	= False
validator_user_agent	= "Django/1.3"

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

= ["%H:%M:%S", ...]

input_time+_formats



	,		
View shortcuts (django.sl	hortcuts)	
est, template, contex	t_dict={}	,	

render(reque context_instance=RequestContext, content_type="text/html", status=200, current_app=None) redirect(to, permanent=False, *args, **kw) get_object_or_4o4(Model, **lookup) get_list_or_404(Model, **lookup)

removetags: "span div" timesince:from date timeuntil:from_date truncatecharsX:"9" truncatewords:"10" truncatewords_html:"10"

lower wordwrap:"20"	_	
make_list yesno:"yeah,no,unknov	vn"	
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.countero }}		
{{ forloop.revcounter }}		
{{ forloop.revcountero }}		
{{ forloop.first }}		
{{ forloop.last }}		
{{ forloop.parentloop }} {% if condition %}{% elif condition %}{% else %}{% e	nd:f0/1	
{% ifchanged %}{% else %}{% endifchanged %}	iluli /oʻ	
{% include "other/template.html" [with who="Jane" [on	lvll %}	
{% load [foo bar from] other_library %}	(9)] 70)	
{% now "jS FY 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 %}		
{% with alpha=1 beta=2 %}{% endwith %}		
Sontry		

,,	•••••	. ш.р.
4		
	Sen	itry
М		
	_	

Sentry

Track production exceptions as they happen. http://www.getsentry.com/

э.	template	strftime	
	a		a.m. / p.m.
	Α	%р	AM / PM
_	b		jan, feb,
	В		
	С		2008-01-02T10:30:00.000123
		%с	Fri Mar 4 16:43:23 2011
	d	%d	01 - 31
	D	%a	Mon, Tue,
	е		UTC, CST (timezone name)
	E		(alternate long month)
	f		1, 1:30
	F	%B	January, February,
	g	%m	1 - 12
	G		0 - 23
	h	%I	01 - 12
	Н	%H	00 - 23
	i	%M	00 - 59 (minutes)
	1		
	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,
	0		(ISO-8601 week-numbering year)
	0		+0200
	Р		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
	Т	%Z	EST, UTC,
	u		(microseconds)
	U	0/	(unix timestamp)
	w W	%w	o (Sun.) - 6 (Sat.)
	VV	%U	1 - 53
	.,	%u %y	01 - 53
	y Y	%y %Y	99
	•	/01 Χ	1999 03/04/11
		X	
	7	^	16:43:23 0 - 365
	Z		(tz offset, seconds)
	DATE FORMAT		(as defined in settings)
	DATETIME_FOR	МАТ	(as defined in settings)
	SHORT_DATE_F		(as defined in settings)
			- · · · · · · · · · · · · · · · · ·
	SHORT_DATETIME_FORMAT (as defined in settings) diango.test.TestCase		
		2,0.150.	
	client_class		= django.test.client
	fixtures		= ["fix1 ison" "fix2 ison"]

Datetime formatting

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=") assertQuerysetEqual(qs, values, transform=repr, ordered=True)

assertRedirects(resp, url, status_code=302, target_status_code=200, msg_prefix=")

assertTemplateUsed, assertTemplateNotUsed(resp, tmpl) with self.assertNumQueries(num, func): ... with self.assertRaisesMessage(exe, msg): ...

with self.settings(SETTING='new_value'): ...