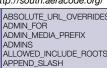
Brought to you by

	Common model field options		Model fields			Find more details in Django's official docs:		ModelAdmin options & callbacks	
	null	= False	AutoField		_	http://diango.n	ne/ <any here="" object=""></any>	actions	= ["method", callback,]
om		= False	BigIntegerField			gom		actions_on_bottom	= False
S.O		= [(1, "Choice 1"),] = "column name"	BooleanField, NullBooleanFie	eld		QueryS	et methods	actions_on_top actions selection counter	= True = True
/Sy		= False	CharField			all()		add form template	= 'add form.html'
ē		= "tablespace_name"	max_length	= 100		annotate(**annotations)		change_form_template	= 'change_form.html'
):d		= value_or_callable	CommaSeparatedIntegerFiel			dates(field, "year/month/d	ay", "ASC/DESC")	change_list_template	= 'changelist.html',
Ħ		= {"name": "Message",}, = "long help text"	max_length	= 50		defer(*fields)		date_hierarchy delete confirmation templa	= "date_field"
ġ.		= False	DateField			distinct()		delete selected confirmation	
g)		= False	DateTimeField		o l	exclude(**lookups) filter(**lookups)		exclude	= ["field_name",]
		= "date_field" = "month field"	DecimalField	= 10	ap	none()		fields	= ["field_name",]
8		= "Field Name"	max_digits decimal places	= 10	Chainable	only(*fields)		fieldsets = [("Name", {"fields": ["fields")	I name" }}
돌		= [validator_obj,]	EmailField	_	ပ်	order_by(*fields) raw(sql, params)		filter_horizontal	= ["m2m_field",]
We			max_length	= 75		reverse()		filter_vertical	= ["m2m_field",]
_	Model Me	eta options	FileField			select_related('field', 'field	d2',, depth=1)	form	= ModelFormClass
	abstract	= False	upload_to	= "files/%Y/%m/%d"		using("db_alias")		formfield_overrides = {TextField: {"widget": My	Widget\ \
		= "applabel"	storage	= file_storage_obj		values(*fields) values list(*fields)		inlines	= [InlineClass,]
		= "table_name" = "space name"	max_length	= 100		values_list(field, flat=True)		list_display	= ["field_name", callable,]
		= "field name"	FilePathField path	= "/var/images"		Talaco_not(nois, nat mao)		list_display_links	= ['field_name',]
	managed	= False	match	= r"\.pdf\$"		get(**lookups)		list_editable list_filter	= ['field_name',] = ['field_name',]
		= "field"	recursive	= True		create(**attributes)		list per page	= 100
		= ["field_name",]	max_length	= 100	<u>o</u>	get_or_create(**attributes count()	s, defaults={})	list_select_related	= False
		= [("code", "text label"),] = False	FloatField		ab	in bulk(id list)		object_history_template	= 'admin/history.html',
		= [("field1", "field2"),]	FileField		chainable	iterator()		ordering paginator	= ['field_name'] = Paginator
		= "verbose name"	upload_to storage	= "/uploads" = file_storage_obj	i i	latest(field)		prepopulated_fields	= {'slug': ['title'],}
	verbose_name_plural	= "plural verbose names"	max length	= 1100 = 100	Not	<pre>aggregate(**aggregations exists()</pre>)	radio_fields = {'fk_field': admi	
	Model rela	ationships	ImageField			update(**attributes)		raw_id_fields	= ['fk_or_m2m_field',]
-		udonompo	upload_to	= "/uploads"		delete()		readonly_fields save_as	= ['field_name',] = False
	ForeignKey(OtherModel)		storage	= file_storage_obj		Lo	okups	save_on_top	= False
		= "things" = Q(foo="bar",)	height_field width field	= "field_name" = "field_name"			окира	search_fields	= ['field_name']
		= "key field name"	max_length	= 100		ct, iexact	= "string"		
	on_delete	= (on_delete option)	~	Field, PositiveSmallIntegerField		tains, icontains tswith, endswith	= "string"	add_view(self, request, form_u	ırl=", extra_context={})
	ManyToManyField(OtherMode		IPAddressField			rtswith, iendswith	= "string"	changelist_view(self, request,	
		= "things"	SlugField		in		= list / queryset	change_view(self, request, ob delete model(self, request, ob	
		= Q(foo="bar",) = "key field name"	max_length	= 100		t, gte, Ite	= date / int / float / decimal	delete_view(self, request, objections)	
		= True,	SmallIntegerField		rang	ge r, month, day	= (lower, upper) = int	formfield_for_choice_field(se	elf, db_field, request, **kw)
		= RelationshipModel	TextField		isnu		= bool	formfield_for_foreignkey(self	
		= "table_name"	TimeField		rege	ex, iregex	= r"^reg\. exp\."	formfield_for_manytomany(s get_paginator(self, queryset,)	
	OneToOneField(OtherModel)	= False	URLField					allow_empty_fi	
		= "thing"	verify_exists	= False		Aggregation/ar	nnotation functions	get_readonly_fields(self, requ	est, obj=None)
		= Q(foo="bar",)			Avg	(field)	StdDev(field, sample=False)	get_urls(self) has add permission(self, rec	u (oot)
		= "key_field_name"	ForeignKey o	n_delete options		unt(field, distinct=False)	Sum(field)	has_change_permission(self	
	on_delete	= (on_delete option)	CASCADE	cascades; default		x(field) n(field)	Variance(field, sample=false)	has_delete_permission(self,	request, obj=None)
			PROTECT	prevent related deletion	IVIII	(IIGIG)		history_view(self, request, obj	
	south	/ Haystack	SET_NULL SET DEFAULT	Set to NULL Set to field's default		Read the Do	cs	message_user(self, request, r queryset(self, request)	nessage)
	Intelligent migrations.	Modular search for Diango	SET_DEFAULT SET(value) / SET(callback)	Set to field's default Set to value / callback()			and browse documentation.	save_formset(self, request, fo	rm, formset, change)
	http://south.aeracode.org/	http://haystacksearch.org/	DO_NOTHING	No action (in Python)		http://readth		save_model(self, request, obj,	form, change)
	ABSOLUTE_URL_OVERRIDES					_UPLOAD_TEMP_DIR	MONTH_DAY_FORMAT		TE_FORMAT
	ADMIN_FOR	CSRF_FAILURE_\		L_SEPARATOR		T_DAY_OF_WEEK	NUMBER_GROUPING		TETIME_FORMAT
	ADMIN_MEDIA_PREFIX ADMINS	DATABASES {"ENGINE"		T_CHARSET T CONTENT TYPE		JRE_DIRS MAT MODULE PATH	PASSWORD_RESET_TI PREPEND WWW	MEOUT_DAYS SITE_ID STATIC RO	OT
	ADMINO	{ ENGINE	DEFAUL	I_OONTENT_TTPE	FUR	VIAI_IVIODULE_PAIT	LUELEIND_AAAAA	STATIC_RU	O1



ALLOWED INCLUDE ROOTS AUTHENTICATION BACKENDS AUTH_PROFILE_MODULE CACHES {"BACKEND" "KEY_FUNCTION" "KEY_PREFIX" "LOCATION" "OPTIONS" "TIMEOUT" "VERSION" CACHE_MIDDLÉWARE_ALIAS

CACHE_MIDDLEWARE_KEY_PREFIX

CACHE MIDDLEWARE SECONDS

CSRF COOKIE DOMAIN

"HOST" "NAME" "OPTIONS" "PASSWORD" "PORT" "USER"} TEST_CHARSET TEST_COLLATION TEST_DEPENDENCIES TEST MIRROR TEST_NAME TEST USER DATABASE_ROUTERS CACHE_MIDDLEWARE_ANONYMOUS_ONLY DATE_FORMAT DATE_INPUT_FORMATS DATETIME FORMAT DATETIME_INPUT_FORMATS

DEFAULT FILE STORAGE DEFAULT_FROM_EMAIL DEFAULT INDEX TABLESPACE DEFAULT_TABLESPACE DISALLOWED_USER_AGENTS EMAIL_BACKEND EMAIL_FILE_PATH EMAIL_HOST EMAIL_HOST_PASSWORD EMAIL_HOST_USER EMAIL_PORT EMAIL_SUBJECT_PREFIX EMAIL_USE_TLS FILE_CHARSET FILE_UPLOAD_HANDLERS FILE_UPLOAD_MAX_MEMORY_SIZE FILE UPLOAD PERMISSIONS

IGNORABLE 404 ENDS IGNORABLE_404_STARTS INSTALLED APPS INTERNAL_IPS LANGUAGE_CODE LANGUAGE_COOKIE_NAME LANGUAGES LOCALE_PATHS LOGGING LOGGING CONFIG LOGIN_REDIRECT_URL LOGIN URL LOGOUT_URL MANAGERS MEDIA_ROOT MEDIA URL MIDDLEWARE_CLASSES

PROFANITIES LIST ROOT URLCONF SECRET_KEY SEND_BROKEN_LINK_EMAILS SERIALIZATION_MODULES SERVER_EMAIL SESSION_COOKIE_AGE SESSION_COOKIE_DOMAIN SESSION_COOKIE_HTTPONLY SESSION_COOKIE_NAME SESSION_COOKIE_PATH SESSION_COOKIE_SECURE SESSION_ENGINE SESSION_EXPIRE_AT_BROWSER_CLOSE

SESSION FILE PATH

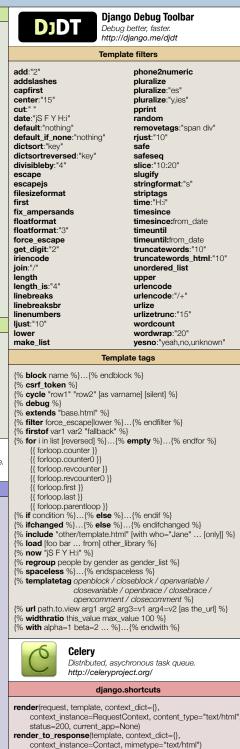
SESSION_SAVE_EVERY_REQUEST

RMAT STATIC URL RESTRUCTUREDTEXT_FILTER_SETTINGS TEMPLATE_CONTEXT_PROCESSORS TEMPLATE DEBUG TEMPLATE_DIRS TEMPLATE_LOADERS TEMPLATE_STRING_IF_INVALID TEST_RUNNER THOUSAND_SEPARATOR TIME_FORMAT TIME_INPUT_FORMATS TIME_ZONE URL_VALIDATOR_USER_AGENT USE_ETAGS USE_I18N USE_L10N USE_THOUSAND_SEPARATOR YEAR_MONTH_FORMAT

REVSYS

DJANGO 1.3 CHEATSHEET

	Common	field options	Forr	n widgets
We know Django. http://revsys.com/	error_messages help_text initial label localize required validators	= {"code": "Message",} = "help text" = value = "field label" = False = True = [validator,]	All widgets attrs CheckboxInput check_test CheckboxSelectMultiple ClearableFileInput	= {'class': 'fancy',} = callback(value)
±	widget	= WidgetClass	DateInput format	= "%Y-%m-%d"
angc	Forn	n fields	DateTimeInput	#0/\\ 0/
ğ	BooleanField		format FileInput	= "%Y-%m-%d %H:%M"
Sugar P	NullBooleanField CharField		HiddenInput, MultipleHidde	nInput
Se l	max_length	= 100 = 10	MultiWidget	
	min_length ChoiceField	= 10	NullBooleanSelect PasswordInput	
	choices	= [(1, "Choice 1"),]	render_value	= False
	TypedChoiceField choices	= [(1, "Choice 1"),]	RadioSelect Select	
	coerce empty_value	= callback(value) = ""	SelectDateWidget	
	MultipleChoiceField	_	years SelectMultiple	= [2010, 2011,]
	choices	= [(1, "Choice 1"),]	SplitDateTimeWidget	
	TypedMultipleChoiceField choices	= [(1, "Choice 1"),]	Textarea	
	coerce empty_value	= callback(value) = ""	TextInput TimeInput	
	DateField		format	= "%H:%M:%S"
	input_formats DateTimeField	= ["%Y-%m-%d",]	Model/Form	n field validators
	input_formats	= ["%Y-%m%d %H:%M",]	MaxLengthValidator(max_le	
	DecimalField max_value min_value max_digits decimal_places	= Decimal(100) = Decimal(10) = 10 = 2	MinLengthValidator(min_length) MaxValueValidator(max_value) MinValueValidator(min_value) RegexValidator(regex, messaure) URLValidator(verify_exists=Fa	e) e) age=None, code=None)
	EmailField FileField		validate_email validate_slug validate_ipv4_address	,
	FilePathField path	= "/home/images"	validate_comma_separated	l_integer_list
	recursive match	= True = r"\.pdf"		e apps and tools for your Django site.
	max_value min_value	= 100.0 = 10.0		opackages.com/
	ImageField	10.0		Signals
	IntegerField max_value min_value IPAddressField	= 100 = 10	django.db.models.signals pre_init(sender, args, kw. post_init(sender, args, kv. pre_save(sender, instance	wargs) ee, using)
	RegexField		post_save(sender, instan pre_delete(sender, instan	
	regex max_length min_length	= r'\w+' = 100 = 10	post_delete(sender, insta m2m_changed(sender, i	nstance, action, reverse,
	SlugField	= 10	class_prepared(sender)	
	max_length min_length	= 100 = 10	post_syncdb(sender, ap verbosity, i	
	TimeField		django.core.signals request started(sender)	•
	input_formats URLField max length	= ["%H:%M:%S",] = 100	request_started(sender) request_finished(sender) got_request_exception)
	min_length verify_exists	= 10 = 10 = False	django.test.signals template_rendered(send	der, template, context)
	validator_user_agent	= "Django/1.3"	django.db.backends.signale	
	SplitDateTimeField input_date_formats input_time+_formats	= ["%Y-%m-%d",] = ["%H:%M:%S",]	django.contrib.auth.signals	
	ModelChoiceField queryset empty_label	= Model.objects.all() = u""	user_logged_in(sender, user_logged_out(sender django.contrib.comments.s	r, request, user) s ignals
	ModelMultipleChoiceField queryset	= Model.objects.all()		ed(sender, comment, request) sender, comment, request) (sender, comment, flag, created, request)



redirect(to, permanent=False, *args, **kw)
get_object_or_404(Model, **lookup)
get_list_or_404(Model, **lookup)

			Fiormatung
	template	strftime	
	а		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,
		70 d	
	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
	,	%i	001 - 365
	1	%A	Monday, Tuesday,
	Ĺ	707 (
			(leap year?)
	m	0/1-	01 - 12
	M	%b	Jan, Feb,
	n		1 - 12
	N		Jan., Feb., March,
	0		+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	/0 <u>/</u>	
			(microseconds)
	U	0/	(unix timestamp)
	W	%w	0 (Sun.) - 6 (Sat.)
	W		1 - 53
		%U	01 - 53
	У	%y	99
	Y	%Y	1999
		X	03/04/11
		X	16:43:23
	z		0 - 365
	Z		(tz offset, seconds)
	DATE_FORMAT	Т	(as defined in settings)
	DATETIME_FO		(as defined in settings)
	SHORT_DATE		(as defined in settings)
		TIME_FORMAT	(as defined in settings)
	SHORT_DATE	TIIVIL_I ONIVIAI	(as delined in Settings)
		Request	HttpDocponeo
	HttpF		HttpResponse
	iter()		init(content=",
	iter() build_absolut		init(content=", mimetype=None,
	iter() build_absolut COOKIES		init(content=", mimetype=None, status=200,
	iter() build_absolut COOKIES encoding		init(content=", mimetype=None, status=200, content_type=None)
	iter() build_absolut COOKIES encoding FILES		init(content=", mimetype=None, status=200, content_type=None) delitem(header)
	iter() build_absolut COOKIES encoding FILES GET	e_uri (path)	init(content=", mimetype=None, status=200, content_type=None) delitem(header) getitem(header)
	iter() build_absolut COOKIES encoding FILES GET get_full_path(e_uri (path)	init(content=", mimetype=None, status=200, content_type=None) delitem(header) setitem(header, val)
	iter() build_absolut COOKIES encoding FILES GET	e_uri (path)	init(content=", mimetype=None, status=200, content_type=None) delitem(header) getitem(header)
	iter() build_absolut COOKIES encoding FILES GET get_full_path(get_host() is_ajax() is_secure()	e_uri (path)	init(content=", mimetype=None, status=200, content_type=None) delitem(header) setitem(header, val) delete_cookie(key,
	iter() build_absolut COOKIES encoding FILES GET get_full_path(get_host() is_ajax() is_secure() META	e_uri (path)	init(content=", mimetype=None, status=200, content_type=None) delitem(header) setitem(header, val) delete_cookie(key, path="/", domain=None) flush()
	iter() build_absolut COOKIES encoding FILES GET get_full_path(get_host() is_ajax() is_secure() META mETA	e_uri (path)	init(content=", mimetype=None, status=200, content_type=None) delitem(header) getitem(header, val) delete_cookie(key, path="/", domain=None) flush() has_header(header)
	iter() build_absolut COOKIES encoding FILES GET get_full_path(get_host() is_ajax() is_secure() META method path	e_uri (path)	init(content=", mimetype=None, status=200, content_type=None)delitem(header)getitem(header)setitem(header, val) delete_cookie(key, path="/", domain=None) flush() has_header(header) set_cookie(key, value,
	iter() build_absolut COOKIES encoding FILES GET get_full_path(get_host() is_ajax() is_secure() META method path path_info	e_uri (path)	init(content=", mimetype=None, status=200, content_type=None)delitem(header)getitem(header,setitem(header, val) delete_cookie(key, path="/", domain=None) flush() has_header(header) set_cookie(key, value, max_age=None,
	iter() build_absolut COOKIES encoding FILES GET get_full_path(get_host() is_ajax() is_secure() META method path path_info POST	.e_uri(path)	init(content=", mimetype=None, status=200, content_type=None)delitem(header)getitem(header, val) delete_cookie(key, path="/", domain=None) flush() has_header(header) set_cookie(key, value, max_age=None, expires=None,
	iter() build_absolut COOKIES encoding FILES GET get_full_path(get_host() is_ajax() is_secure() META method path path_info POST raw_post_dat	e_uri(path)	init(content=", mimetype=None, status=200, content_type=None)delitem(neader)getitem(header)setitem(header, val) delete_cookie(key, path="/", domain=None) flush() has_header(header) set_cookie(key, value, max_age=None, expires=None, path="/",
,	iter() build_absolut COOKIES encoding FILES GET get_full_path(get_host() is_ajax() is_secure() META method path path_info POST raw_post_dat read(size=Non	e_uri(path)	init(content=", mimetype=None, status=200, content_type=None)delitem(header)getitem(header)setitem(header, val) delete_cookie(key, path="/", domain=None) flush() has_header(header) set_cookie(key, value, max_age=None, expires=None, path="/", domain=None,
,	iter() build_absolut COOKIES encoding FILES GET get_full_path(get_host() is_ajax() is_secure() META method path path_info POST raw_post_dat read(size=Non readline()	e_uri(path)	init(content=", mimetype=None, status=200, content_type=None)delitem(header)getitem(header)setitem(header, val) delete_cookie(key, path="/", domain=None) flush() has_header(header) set_cookie(key, value, max_age=None, expires=None, path="/", domain=None, secure=None, secure=None,
,	iter() build_absolut COOKIES encoding FILES GET get_full_path(get_host() is_ajax() is_secure() META method path_info POST raw_post_dat read(size=Non readline()	e_uri(path)	init(content=", mimetype=None, status=200, content_type=None)delitem(neader)getitem(header)setitem(header, val) delete_cookie(key, path="/", domain=None) flush() has_header(header) set_cookie(key, value, max_age=None, expires=None, path="/", domain=None, secure=None, httponly=False)
,	iter() build_absolut COOKIES encoding FILES GET get_full_path(get_host() is_ajax() is_secure() META method path path_info POST raw_post_dat read(size=Non readline() REQUEST	e_uri(path)	init(content=", mimetype=None, status=200, content_type=None)delitem(header)getitem(header)setitem(header, val) delete_cookie(key, path="/", domain=None) flush() has_header(header) set_cookie(key, value, max_age=None, expires=None, path="/", domain=None, secure=None, httponly=False) tell()
,	iter() build_absolut COOKIES encoding FILES GET get_full_path(get_host() is_ajax() is_secure() META method path path_info POST raw_post_dat read(size=Non readlines() REQUEST session	e_uri(path)	init(content=", mimetype=None, status=200, content_type=None)delitem(neader)getitem(header)setitem(header, val) delete_cookie(key, path="/", domain=None) flush() has_header(header) set_cookie(key, value, max_age=None, expires=None, path="/", domain=None, secure=None, httponly=False)
,	iter() build_absolut COOKIES encoding FILES GET get_full_path(get_host() is_ajax() is_secure() META method path path_info POST raw_post_dat read(size=Non readline() REQUEST	e_uri(path)	init(content=", mimetype=None, status=200, content_type=None)delitem(header)getitem(header)setitem(header, val) delete_cookie(key, path="/", domain=None) flush() has_header(header) set_cookie(key, value, max_age=None, path="/", domain=None, secure=None, httponly=False) tell()

Datetime formatting