




Model fields	Common model field options	Find more details in Django's official docs: <a href="http://django.me/<any object here>">http://django.me/<any object here>	ModelAdmin options & callbacks	
AutoField BigIntegerField BooleanField, NullBooleanField CharField max_length = 100 CommaSeparatedIntegerField max_length = 50 DateField, 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	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, ...]	QuerySet methods Chainable 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)	ModelAdmin options & callbacks 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 = [{"Name": {"fields": ["field_name", ...]}}], ... filter_horizontal = ["m2m_field", ...] filter_vertical = ["m2m_field", ...] form = ModelFormClass formfield_overrides = {TextFiled: {"widget": MyWidget}, ...} inlines = [InlineClass, ...] list_display = ["field_name", callable, ...] list_display_links = ["field_name", ...] list_editable = ["field_name", ...] list_filter = ["field_name", ...] list_max_show_all = 200 list_per_page = 100 list_select_related = False object_history_template = "admin/history.html", ordering = ["field_name"] paginator = Paginator prepopulated_fields = {'slug': ['title', ...]} radio_fields = {'fk_field': admin.HORIZONTAL, ...} raw_id_fields = ['fk_or_m2m_field', ...] readonly_fields = ["field_name", ...] save_as = False save_on_top = False search_fields = ["field_name"]	
	Model Meta options abstract = False app_label = "applabel" db_table = "table_name" db_tablespace = "space_name" get_latest_by = "field_name" managed = False order_with_respect_to = "field" ordering = ["field_name", ...] permissions = [{"code", "text label"}, ...] proxy = False unique_together = [("field1", "field2"), ...] verbose_name = "verbose name" verbose_name_plural = "plural verbose names"	Not chainable 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)		
	ForeignKey on_delete options 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)	Lookups 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"^\reg\. exp\."		
	Model/Form field validators 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 _ipv6, _ipv4_6) validate_comma_separated_integer_list	Aggregation/annotation functions Avg(field) StdDev(field, sample=False) Count(field, distinct=False) Sum(field) Max(field) Variance(field, sample=False) Min(field)		
	 Read the Docs Create, host, and browse documentation. http://readthedocs.org/			
	Model relationships ForeignKey(OtherModel) related_name = "things" limit_choices_to = Q(foo="bar", ...) to_field = "key_field_name" on_delete = (see right) 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 = (see right)	Signals 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)	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) 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)	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)
			 Haystack Modular search for Django. http://haystacksearch.org/	
			 south Intelligent migrations. http://south.aeracode.org/	

