Django Admin

==>Custamizing Attributes by models

*** .** in admin.py

```
admin.site.site_header=’ ’ :- change header in admin pannel
admin.site.site_title=’ ’     :-change the title in the admin pannel
admin.site.index_title=’ ’:-
```

==> adding model admin classes

***. in admin.py**

```
class Table(admin.ModelAdmin):
        list_display = [‘ coloum1 ’,’coloum2   ’,’coloum 3    ’]
        list_display_links =  [‘ coloum1 ’,’coloum2   ’,’coloum 3    ’]
        list-editable   = [‘ coloum1 ’,’coloum2   ’,’coloum 3    ’,]
        actions_on_bottom = True/False
        search_fields = [‘ coloum1 ’,’coloum2   ’,’coloum 3    ’]
        list_filter = [‘ coloum1 ’,’coloum2   ’,’coloum 3    ’]

admin.site.register(Table_name,Table)
```

--> Table class apples to Table_name model.

--> **list_display** :- display the columns which are in it, in the admin pannel.

-->**list_display_links**:-It is same as list_display but these fields are links.
If we click on these links .we can chane the data of that row.

--> **list_editable :-** these fields we can edit the particular value .without going to edit page.

NOTE:- list_display_links conn’t be in list_editable.

-->**actions_on_bottom** :-if it is True it display in the bottom along with top.

---> **search_field:-**we can search these values of these attributes.

---> **list_filter:-** it display the recordes order_by the given attributes.

==> adding actions and custamizing Editing pages.

```
class Table(admin.ModelAdmin):
        fields = [‘ coloum1 ’,’coloum2   ’,’coloum 3    ’]
        readonly_fields =[‘column1’]



admin.site.register(Tabl_name,Table)
```

--> **fields** :-these attributes only display on the data editing page.

--> **readonly_fields**:-we cannot edit auto generated values such as
‘id’,’date_auto_now’ ...etc

Django Admin

```
      class Table(admin.ModelAdmin):
            fields = [(‘ coloum1 ’,’coloum2   ’),’coloum 3    ’]
            readonly_fields =[‘column1’]
    admin.site.register(Tabl_name,Table)
```

NOTE:-here column1 and column2 are display in same line in edditing page.

```
class AdminForm(admin.ModelAdmin):
      save_on_top=True
      fieldsets =[('main',{
                        'fields':['balance','name','addr']
                     }
                ),
      ('details',{
          'fields':['status','payment']
            }
      )
      ]
```

Note:-here **fieldsets:-** divide the sections and display the fields.

In the abbove case **main** section contains balance,name,addr fields.And **details** section contains status,payment.

save_on_top :- if it is True it display the save option on the top of the editing page.

==> **Inline editors:-**

```
          from .models import Types,Items
          class types(admin.TabularInline):
              model = Types
          class adminType(admin.ModelAdmin):
              inlines=[types]
      admin.site.register(Items,adminType)
```

Note:- Types contains foreignkey it refers to Items.

Inlines:- it show the Items Data and reffered Types data.