

خب به طور خلاصه شما باید این مراحل رو برید

۱- مدلتون رو بسازید

۲- اطلاعات تون رو ذخیره کنید

۳- اطلاعات تون رو از db بگیرید

## ساخت مدل

خب برای این فقط لازم برید تو `models.py` مدل تونو این شکلی بزیند که خب جای `MyModel` اسم مدل تون رو میزنید و بعد اون پایین هم اطلاعات شو میزنید

```
class MyModel(models.Model):  
    name = models.CharField(max_length = 100)
```

مثلا اگر قد و وزن مدل تونم میخواستید تو `db` ذخیره کنید اینارو هم باید اضافه کنید

```
height = models.IntegerField()  
weight = models.IntegerField()
```

خب تو اینجا چون قد و وزن رشته نیستن عدد هستن به جای `CharField` از `IntegerField` استفاده کردیم. تو [اینجا](#) انواع `Data Type` ها رو میتونید ببینید

## ذخیره دیتا

### ۱. در `views`:

خب این روش بیشتر برای وقتی که میخوایم اطلاعاتی که یوزر با پست بهمون فرستاده رو تو `db` ذخیره کنیم به کار میره برای اینکار میزنیم

```
model = MyModel.objects.create(name=request.POST["name"])
```

## ۲. در ادمین پنل:

خب این روش بیشتر برای وقتی که خودمون میخوایم یه دیتایی رو به DB مون اضافه کنیم بیشتر بکار میره  
ابتدا مدل مون رو تو `admin.py` رو رجیستر میکنیم

```
admin.site.register(MyModel)
```

بعد یه ترمینال جدید باز میکنیم و این دستورات رو می زنیم

```
python manage.py makemigrations
```

```
python manage.py migrate
```

```
python manage.py createsuperuser
```

دستور اول و دوم رو هر زمانی که مدل جدید اضافه کردیم یا مدلی رو تغییر دادیم باید بزنیم

ولی سومی رو فقط برای بار اول برای ساخت سوپر یوزر ( برای لاگین شدن به پنل ادمین ) میسازیم

توجه کنید وقتی دارید پاسورد رو میزنید کارکتر ها مخفی میشن به این خاطر نمیتونید ببینید چی دارید  
تایپ میکنید

خب وقتی یوزر تونو ساختید برید اینجا و لاگین شید

<http://127.0.0.1:8000/admin>

از اینجا هم میتونید دیتاتون رو اضافه کنید

BLOG		
My another models	<a href="#">+ Add</a>	<a href="#">Change</a>
My models	<a href="#">+ Add</a>	<a href="#">Change</a>

برای اینکه دیتای هر مدل رو ببینید رو اسم شون کلیک کنید

برای اینکه اسم هاشون اینجوری نشه

Select my another model to change

Action:   0 of 1 selected

☐

MY ANOTHER MODEL

☐

**MyAnotherModel object (1)**

1 my another model

فیلد `name` + تابع زیر رو به کلاس مدل هاتون اضافه کنید

```
class MyModel(models.Model):  
    name = models.CharField(max_length = 100)  
    def __str__(self):  
        return self.name
```

در هر دو روش یادتون نره هر کجا میخواید از مدل هاتون استفاده کنید باید `import` اش کنید

```
from .models import MyModel
```

## گرفتن دیتا

برای گرفتن همه ی object ها از `MyModel.objects.all()` استفاده می کنیم برای فیلتر کردن و گرفتن یه سری object های خاص از `MyModel.objects.filter(condition)` استفاده میکنیم

چند تا مثال با هم ببینیم:

میخوایم همه رو بگیریم

```
Peoples=MyModel.objects.all()
```

میخوایم فقط اسم مدل هامون رو به صورت یه لیستی بگیریم

```
Names=MyModel.objects.value_list("name",flat=True)
```

میخوایم اون مدل هایی که خانم هستن رو بگیریم

```
khanomHa=MyModel.objects.filter(gender="female")
```

اگر بخوایم یه دونه بگیریم جای فیلتر از `get` استفاده میکنیم

```
yeKhanom=MyModel.objects.get(gender="female")
```

میخوایم اونایی رو که سن شون بالاتر از ۱۸ هستش رو بگیریم

```
MyModel.objects.filter(age__gt=18)
```

می بینید تهش `__gt` اضافه شده خب `gt` مخفف `greater than` هستش

توجه کنید دو تا آندرلاین بین شون میذاریم

- 1- gte: greater than equal >=
- 2- lte: less than equal <=
- 3- gt: greater than >
- 4- lt: less than <
- 5- icontains: Case-insensitive contains

یه مثال دیگه : میخوایم اونایی که تو اسمشون یا `sa` هستش رو بگیریم (برای سرچ بدرد میخوره)

```
MyModel.objects.filter(name__icontains="sa")
```

برای sort کردن شون بر اساس یه property مثلا سن از order\_by استفاده میکنیم

برای مثال این همه رو بر اساس سن شون از کوچک به بزرگ مرتب میکنه

```
MyModel.objects.order_by("age")
```

برای اینکه از بزرگ به کوچک مرتب کنه منفی میذاریم پشتش:

```
MyModel.objects.order_by("-age")
```