@@@ 모델에 대한 정리 @@@

class MyModelName(models.Model):

    #Fields

    my\_field\_name = models.CharField(max\_length=20)

    #Metadata

    class Meta:

        ordering = ['-my\_field\_name']

    #Methods

    def \_\_str(self):

        return self.my\_field\_name

대충 이런 모양새다. class 속성으로 선언하고 인자는 models.Model임

#Fields

필드. 대충 변수 같다.

- 필드 타입  
CharField: 고정길이 문자열. max\_length 필수. 제목 같은데에 쓰면 되는 듯.  
TextField: 겁나 주저리주저리 할 때 쓰는듯. 본문 같은데에 쓰면 되는 듯.  
IntegerField: 설명이 필요한지?  
DateField, DateTimeField: 시간 관리하는거 같음.  
EmailField: 이메일?  
FileField, ImageField: 파일, 이미지  
ForeignKey: 일대다 관계일 경우 사용. 한명의 유저에게서 여러 개의 게시글이 나올 수 있음.  
ManyToManyField: 다대다 관계일 경우 사용. 태그와 게시글의 관계를 생각하면 편할 듯?  
그 외 매우 많은데 일단은 쓸법한 것만 정리함

#Metadata

메타데이터. 순서 제어하는 것 같은데 자세한건 잘 모르겠음.  
ordering =   
fields = (’11, ‘22’): 그냥 저 순서인듯

#Methods

메소드. 설명이 필요한지? 근데

def \_\_str(self):

        return self.my\_field\_name

이거 필수인듯. 이유는 아직 잘 모르겠음

#migrate

모델을 편집하고 나면

py manage.py makemigrations  
py manage.py migrate

해줘야함. 하라면 해 그냥.