def **str**(self): is a python method which is called when we use print/str to convert object into a string. It is predefined , however can be customised. Will see step by step.Suppose below is our code.

class topics():

def \_\_init\_\_(self,topics):

print "inside init"

self.topics=topics

my\_top = topics("News")

print(my\_top)

Output:

inside init

<\_\_main\_\_.topics instance at 0x0000000006483AC8>

So while printing we got reference to the object. Now consider below code.

class topics():

def \_\_init\_\_(self,topics):

print "inside init"

self.topics=topics

def \_\_str\_\_(self):

print "Inside \_\_str\_\_"

return "My topics is " + self.topics

my\_top = topics("News")

print(my\_top)

Output:

inside init

Inside \_\_str\_\_

My topics is News

So, here instead of object we are printing the object. As we can see we can customize the output as well. Now, whats the importance of it in a django models.py file?

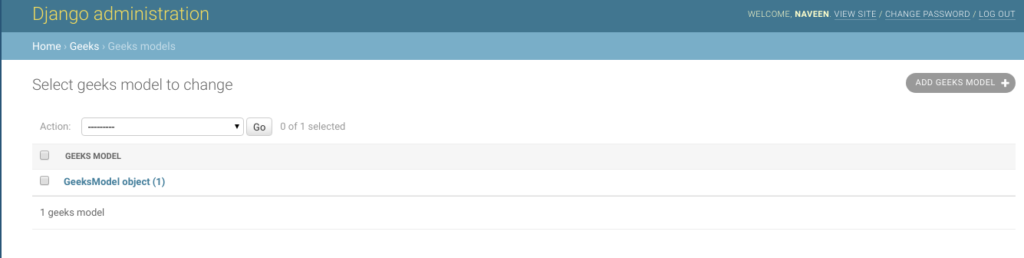
When we use it in models.py file, go to admin interface, it creates object as "News", otherwise entry will be shown as **main**.topics instance at 0x0000000006483AC8 which won't look that much user friendly.

The \_\_str\_\_ function is used add a string representation of a model's object, so that is

to tell Python what to do when we convert a model instance into a string.

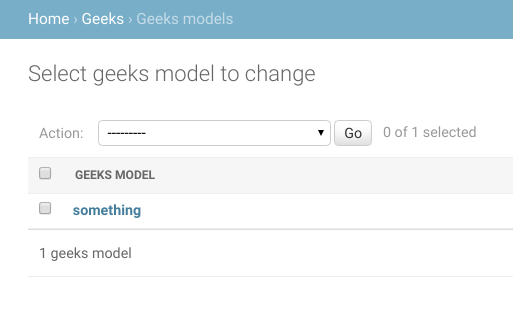
And if you dont mention it then it will take it by default the USERNAME\_FIELD for that purpose.

The \_\_str\_\_ method just tells Django what to print when it needs to print out an instance of the any model. It is also what lets your admin panel, go from this

[](https://i.stack.imgur.com/jlkx5.png)

Note: how objects are just plainly numbered

to this

[](https://i.stack.imgur.com/vMJnz.png).

Note: proper object names here

You could choose what to show in the admin panel, as per your choice. Be it a field value or a default value or something else.