



---

[Django](#) [Views](#) [Model](#) [Template](#) [Forms](#) [Jinja](#) [Python SQLite](#) [Flask](#) [Json](#) [Postman](#) [Interview Ques](#)

---

## CharField – Django Models

Last Updated : 27 Mar, 2023

---

CharField is a string field, for small- to large-sized strings. It is like a string field in C/C++. CharField is generally used for storing small strings like first name, last name, etc. To store larger text TextField is used. The default form widget for this field is [TextInput](#).

**CharField has one extra required argument:**

`CharField.max_length`

The maximum length (in characters) of the field. The max\_length is enforced at the database level and in Django's validation using [MaxLengthValidator](#).

**Syntax:**

```
field_name = models.CharField(max_length=200, \*\*options)
```

## Django Model CharField Explanation

Illustration of CharField using an Example. Consider a project named geeksforgeeks having an app named geeks.

*Refer to the following articles to check how to create a project and an app in Django.*

- [How to Create a Basic Project using MVT in Django?](#)
- [How to Create an App in Django ?](#)

Enter the following code into models.py file of **geeks** app.

## Python3

```
from django.db import models
from django.db.models import Model
# Create your models here.

class GeeksModel(Model):
    geeks_field = models.CharField(max_length = 200)
```

Add the geeks app to INSTALLED\_APPS

---

## Python3

```
# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'geeks',
]
```

Now when we run makemigrations command from the terminal,

```
Python manage.py makemigrations
```

A new folder named migrations would be created in geeks directory with a file named 0001\_initial.py

---

## Python3

```
# Generated by Django 2.2.5 on 2019-09-25 06:00

from django.db import migrations, models

class Migration(migrations.Migration):
```

```

initial = True

dependencies = [
]

operations = [
    migrations.CreateModel(
        name='GeeksModel',
        fields=[
            ('id',
             models.AutoField(
                 auto_created = True,
                 primary_key = True,
                 serialize = False,
                 verbose_name = 'ID'
             )),
            ('geeks_field',
             models.CharField(
                 max_length = 200,
             )),
        ],
    ),
]

```

Now run,

```
Python manage.py migrate
```

Thus, an `geeks_field` **CharField** is created when you run migrations on the project. It is a field to store small- to large-sized strings.

## How to use CharField ?

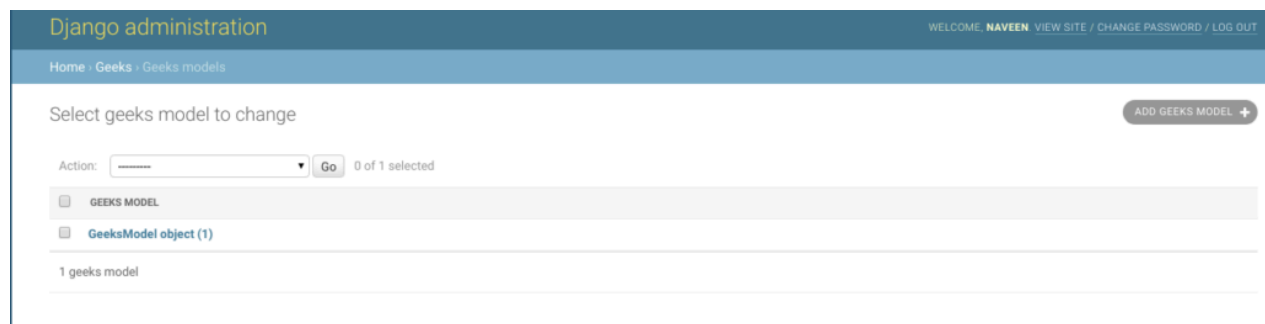
CharField is used for storing small sized strings in the database. One can store First Name, Last Name, Address Details, etc. CharField should be given an argument `max_length` for specifying the maximum length of string it is required to store. In production server, after the Django application is deployed, space is very limited. So it is always optimal to use `max_length` according to the requirement of the field. Let us create an instance of the CharField we created and check if it is working.

## Python3

```
# importing the model
# from geeks app
from geeks.models import GeeksModel

# creating a instance of
# GeeksModel
geek_object = GeeksModel.objects.create(geeks_field ="GFG is Best")
geek_object.save()
```

Now let's check it in admin server. We have created an instance of GeeksModel.



## Field Options

Field Options are the arguments given to each field for applying some constraint or imparting a particular characteristic to a particular Field. For example, adding an argument `null = True` to CharField will enable it to store empty values for that table in a relational database.

Here are the field options and attributes that an CharField can use.

Field Options	Description
<a href="#">Null</a>	If <b>True</b> , Django will store empty values as <b>NULL</b> in the database. Default is <b>False</b> .
<a href="#">Blank</a>	If <b>True</b> , the field is allowed to be blank. Default is <b>False</b> .

Field Options	Description
<b>db_column</b>	The name of the database column to use for this field. If this isn't given, Django will use the field's name.
<b>db_index</b>	If True, a database index will be created for this field
<b>db_tablespace</b>	The name of the database tablespace to use for this field's index, if this field is indexed. The default is the project's <code>DEFAULT_INDEX_TABLESPACE</code> setting, if set, or the <code>db_tablespace</code> of the model, if any. If the backend doesn't support tablespaces for indexes, this option is ignored.
<a href="#"><u>Default</u></a>	The default value for the field. This can be a value or a callable object. If callable it will be called every time a new object is created.
<a href="#"><u>help_text</u></a>	Extra "help" text to be displayed with the form widget. It's useful for documentation even if your field isn't used on a form.
<a href="#"><u>primary_key</u></a>	If True, this field is the primary key for the model.
<a href="#"><u>editable</u></a>	If <b>False</b> , the field will not be displayed in the admin or any other <code>ModelForm</code> . They are also skipped during model validation. Default is <b>True</b> .
<a href="#"><u>error_messages</u></a>	The <code>error_messages</code> argument lets you override the default messages that the field will raise. Pass in a dictionary with keys matching the error messages you want to override.

Field Options	Description
<a href="#">help_text</a>	Extra “help” text to be displayed with the form widget. It’s useful for documentation even if your field isn’t used on a form.
<a href="#">verbose_name</a>	A human-readable name for the field. If the verbose name isn’t given, Django will automatically create it using the field’s attribute name, converting underscores to spaces.
<a href="#">validators</a>	A list of validators to run for this field. See the <a href="#">validators documentation</a> for more information.
<a href="#">Unique</a>	If True, this field must be unique throughout the table.

N Nave...



Previous Article

BooleanField - Django Models

Next Article

DateField - Django Models

Similar Reads

CharField - Django Forms

CharField in Django Forms is a string field, for small- to large-sized strings. It is used for taking text inputs from the user. The default widget for this input is...

5 min read

What's the Difference Between CharField and TextField in Django?

Understanding the difference between CharField and TextField is crucial in web development, especially when using Django. These two are among the most...

5 min read

## Add a Placeholder on a CharField in Django

Django is a popular web framework with Python that builds scalable and robust web applications. One of the common tasks in web development is creating for...

5 min read

## Django Models | Set - 1

Prerequisites : Django Creating apps Models - According to Django Models, A model is the single, definitive source of information about your data. It contains...

2 min read

## Django Models | Set - 2

Model Fields - Model fields define the datatype which will be stored in the provided variable. To store price, the integer type is more suitable. To store heig...

2 min read

## Python | Relational fields in Django models

Prerequisite: Django models Django models represent real-world entities, and it is rarely the case that real-world entities are entirely independent of each other...

4 min read

## BigIntegerField - Django Models

BigIntegerField is a 64-bit integer, much like an IntegerField except that it is guaranteed to fit numbers from -9223372036854775808 to...

4 min read

## AutoField - Django Models

According to documentation, An AutoField is an IntegerField that automatically increments according to available IDs. One usually won't need to use this directl...

4 min read

## BigAutoField - Django Models

BigAutoField is a 64-bit integer, much like an AutoField except that it is guaranteed to fit numbers from 1 to 9223372036854775807. One can create a...

4 min read

## BooleanField - Django Models

BooleanField is a true/false field. It is like a bool field in C/C++. The default form widget for this field is CheckboxInput, or NullBooleanSelect if null=True. The...

4 min read

**Article Tags :** [Python](#) [Django-models](#) [Python Django](#)

**Practice Tags :** [python](#)



Corporate & Communications Address:-  
A-143, 9th Floor, Sovereign Corporate  
Tower, Sector- 136, Noida, Uttar Pradesh  
(201305) | Registered Address:- K 061,  
Tower K, Gulshan Vivante Apartment,  
Sector 137, Noida, Gautam Buddh  
Nagar, Uttar Pradesh, 201305



### Company

[About Us](#)  
[Legal](#)  
[In Media](#)  
[Contact Us](#)  
[Advertise with us](#)  
[GFG Corporate Solution](#)  
[Placement Training Program](#)  
[GeeksforGeeks Community](#)

### DSA

[Data Structures](#)  
[Algorithms](#)

### Languages

[Python](#)  
[Java](#)  
[C++](#)  
[PHP](#)  
[GoLang](#)  
[SQL](#)  
[R Language](#)  
[Android Tutorial](#)  
[Tutorials Archive](#)

### Data Science & ML

[Data Science With Python](#)  
[Data Science For Beginner](#)



DSA for Beginners  
Basic DSA Problems  
DSA Roadmap  
Top 100 DSA Interview Problems  
DSA Roadmap by Sandeep Jain  
All Cheat Sheets

Web Technologies

HTML  
CSS  
JavaScript  
TypeScript  
ReactJS  
NextJS  
Bootstrap  
Web Design

Computer Science

Operating Systems  
Computer Network  
Database Management System  
Software Engineering  
Digital Logic Design  
Engineering Maths  
Software Development  
Software Testing

System Design

High Level Design  
Low Level Design  
UML Diagrams  
Interview Guide  
Design Patterns  
OOAD  
System Design Bootcamp  
Interview Questions

School Subjects

Mathematics  
Physics  
Chemistry  
Biology  
Social Science  
English Grammar  
Commerce  
World GK

Machine Learning  
ML Maths  
Data Visualisation  
Pandas  
NumPy  
NLP  
Deep Learning

Python Tutorial

Python Programming Examples  
Python Projects  
Python Tkinter  
Web Scraping  
OpenCV Tutorial  
Python Interview Question  
Django

DevOps

Git  
Linux  
AWS  
Docker  
Kubernetes  
Azure  
GCP  
DevOps Roadmap

Inteview Preparation

Competitive Programming  
Top DS or Algo for CP  
Company-Wise Recruitment Process  
Company-Wise Preparation  
Aptitude Preparation  
Puzzles

GeeksforGeeks Videos

DSA  
Python  
Java  
C++  
Web Development  
Data Science  
CS Subjects

