



How to use Django Field Choices ?

Last Updated : 03 Nov, 2019

Django Field Choices. According to documentation Field Choices are a sequence consisting itself of iterables of exactly two items (e.g. [(A, B), (A, B) ...]) to use as choices for some field. For example, consider a field semester which can have options as { 1, 2, 3, 4, 5, 6 } only. Choices limits the input from the user to the particular values specified in `models.py`. If choices are given, they're enforced by [model validation](#) and the default form widget will be a select box with these choices instead of the standard text field. Choices can be any sequence object – not necessarily a list or tuple.

The first element in each tuple is the actual value to be set on the model, and the second element is the human-readable name.

For example,

```
SEMESTER_CHOICES = (  
    ("1", "1"),  
    ("2", "2"),  
    ("3", "3"),  
    ("4", "4"),  
    ("5", "5"),  
    ("6", "6"),  
    ("7", "7"),  
    ("8", "8"),  
)
```

Let us create a choices field with above semester in our django project named geeksforgeeks.

```
from django.db import models
```

```
# specifying choices
```

```
SEMESTER_CHOICES = (  
    ("1", "1"),  
    ("2", "2"),  
    ("3", "3"),  
    ("4", "4"),  
    ("5", "5"),  
    ("6", "6"),  
    ("7", "7"),  
    ("8", "8"),  
)
```

```
("1", "1"),  
("2", "2"),  
("3", "3"),  
("4", "4"),  
("5", "5"),  
("6", "6"),  
("7", "7"),
```

Django Views Model Template Forms Jinja Python SQLite Flask Json Postman Interview Ques

declaring a Student Model

```
class Student(models.Model):  
    semester = models.CharField(  
        max_length = 20,  
        choices = SEMESTER_CHOICES,  
        default = '1'  
    )
```

Let us check in admin panel how semester is created.

Home > Geeks > Students > Add student

Add student

Semester:

1 ▾

1

2

3

4

5

6

7

8

One can also collect your available choices into named groups that can be used for organizational purposes:

```
MEDIA_CHOICES = [  
    ('Audio', (  
        ('vinyl', 'Vinyl'),  
        ('cd', 'CD'),  
    )  
),  
    ('Video', (  
        ('vhs', 'VHS Tape'),
```

```
        ('dvd', 'DVD'),  
    ),  
    ('unknown', 'Unknown'),  
]
```

The first element in each tuple is the name to apply to the group. The second element is an iterable of 2-tuples, with each 2-tuple containing a value and a human-readable name for an option. Grouped options may be combined with ungrouped options within a single list (such as the unknown option in this example).

For each model field that has choices set, Django will add a method to retrieve the human-readable name for the field's current value. See [get_FOO_display\(\).](#) in the database API documentation.

Are you ready to elevate your web development skills from foundational knowledge to advanced expertise? Explore our [Mastering Django Framework - Beginner to Advanced Course](#) on GeeksforGeeks, designed for aspiring developers and experienced programmers. This comprehensive course covers everything you need to know about Django, from the basics to advanced features. Gain practical experience through **hands-on projects** and real-world applications, mastering essential Django principles and techniques. Whether you're just starting or looking to refine your skills, this course will empower you to build sophisticated web applications efficiently. Ready to enhance your web development journey? Enroll now and unlock your potential with Django!

N Nave...



Previous Article

Meta Class in Models - Django

Next Article

Overriding the save method - Django
Models

Similar Reads

How to Filter ForeignKey Choices in a Django ModelForm

A ForeignKey field allows one model to relate to another in Django. When this field is represented in a form (e.g., a Django ModelForm), it typically displays a...

7 min read

random.choices() method in Python

The choices() method returns multiple random elements from the list with replacement. You can weigh the possibility of each result with the weights...

2 min read

Built-in Field Validations - Django Models

Built-in Field Validations in Django models are the default validations that come predefined to all Django fields. Every field comes in with built-in validations fro...

3 min read

default - Django Built-in Field Validation

Built-in Field Validations in Django models are the validations that come predefined to all Django fields. Every field comes in with built-in validations fro...

3 min read

blank=True - Django Built-in Field Validation

Built-in Field Validations in Django models are the default validations that come predefined to all Django fields. Every field comes in with built-in validations fro...

4 min read

null=True - Django Built-in Field Validation

Built-in Field Validations in Django models are the default validations that come predefined to all Django fields. Every field comes in with built-in validations fro...

4 min read

error_messages - Django Built-in Field Validation

Built-in Field Validations in Django models are the validations that come predefined to all Django fields. Every field comes in with built-in validations fro...

4 min read

help_text - Django Built-in Field Validation

Built-in Field Validations in Django models are the validations that come predefined to all Django fields. Every field comes in with built-in validations fro...

4 min read

Custom Field Validations in Django Models

This article revolves around how to add custom validation to a particular field. For example to add validation of an email to a CharField by specifying a particular...

3 min read

verbose_name - Django Built-in Field Validation

Built-in Field Validations in Django models are the validations that come predefined to all Django fields. Every field comes in with built-in validations fro...

3 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](#)

Languages

[Python](#)
[Java](#)

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

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

DSA

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

Data Science & ML

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

Interview Preparation

[Competitive Programming](#)
[Top DS or Algo for CP](#)
[Company-Wise Recruitment Process](#)
[Company-Wise Preparation](#)
[Aptitude Preparation](#)
[Puzzles](#)

School Subjects	GeeksforGeeks Videos
Mathematics	DSA
Physics	Python
Chemistry	Java
Biology	C++
Social Science	Web Development
English Grammar	Data Science
Commerce	CS Subjects
World GK	

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved