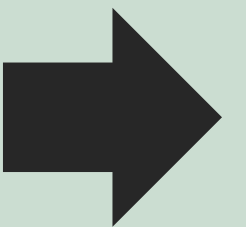# Difference between models.TextChoices and Charfield (Selection concept)

CharField:

This field stores short-medium length characters or text strings, which makes it suitable to store an attribute like a name.

- **CharField has a (max_length) parameter you must specify every time you use the field. But when you do not specifythe field length, it defaults to 255 characters.**

Example ➡

```python
from django.db import models
class product(models.Model):
    name = models.CharField(max_length=20)
```

Field.choices:

A mapping or iterable in the format described below
to use as choices for this field. 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.

If mapping is given, the key element is the actual value to be set on the model,
and the second element is the human-readable name.

```
YEAR_IN_SVHOOL_CHOICES = {
"FR": "Freshman",
"SO": "Sophomore",
"JR": "Junior",
"SR": "Senior",
"GR": "Graduate",
}
```

Selection Methods: (CharField)

```python
from django.db import models


class Student(models.Model):
    FRESHMAN = "FR"
    SOPHOMORE = "SO"
    JUNIOR = "JR"
    SENIOR = "SR"
    GRADUATE = "GR"
    YEAR_IN_SCHOOL_CHOICES = {
        FRESHMAN: "Freshman",
        SOPHOMORE: "Sophomore",
        JUNIOR: "Junior",
        SENIOR: "Senior",
        GRADUATE: "Graduate",
    }
    year_in_school = models.CharField(
        max_length=2,
        choices=YEAR_IN_SCHOOL_CHOICES,
        default=FRESHMAN,
    )

    def is_upperclass(self):
        return self.year_in_school in {self.JUNIOR, self.SENIOR}
```

Selection Methods: (model.TextChoices)

Django provides enumeration types that you can subclass to define choices in a concise way:

```python
class Student(models.Model):
    class YearInSchool(models.TextChoices):
        FRESHMAN = "FR", _("Freshman")
        SOPHOMORE = "SO", _("Sophomore")
        JUNIOR = "JR", _("Junior")
        SENIOR = "SR", _("Senior")
        GRADUATE = "GR", _("Graduate")

    year_in_school = models.CharField(
        max_length=2,
        choices=YearInSchool,
        default=YearInSchool.FRESHMAN,
    )

    def is_upperclass(self):
        return self.year_in_school in {
            self.YearInSchool.JUNIOR,
            self.YearInSchool.SENIOR,
        }
```