

## Basic model data types and fields list

Field Name	Description
<u><a href="#">AutoField</a></u>	It is an IntegerField that automatically increments.
<u><a href="#">BigAutoField</a></u>	It is a 64-bit integer, much like an AutoField except that it is guaranteed to fit numbers from 1 to 9223372036854775807.
<u><a href="#">BigIntegerField</a></u>	It is a 64-bit integer, much like an IntegerField except that it is guaranteed to fit numbers from -9223372036854775808 to 9223372036854775807.
<u><a href="#">BinaryField</a></u>	A field to store raw binary data.
<u><a href="#">BooleanField</a></u>	A true/false field. The default form widget for this field is a CheckboxInput.
<u><a href="#">CharField</a></u>	A field to store text-based values.
<u><a href="#">DateField</a></u>	A date, represented in Python by a datetime.date instance
<u><a href="#">DateTimeField</a></u>	It is used for date and time, represented in Python by a datetime.datetime instance.
<u><a href="#">DecimalField</a></u>	It is a fixed-precision decimal number, represented in Python by a Decimal instance.
<u><a href="#">DurationField</a></u>	A field for storing periods of time.
<u><a href="#">EmailField</a></u>	It is a CharField that checks that the value is a valid email address.
<u><a href="#">FileField</a></u>	It is a file-upload field.
<u><a href="#">FloatField</a></u>	It is a floating-point number represented in Python by a float instance.
<u><a href="#">ImageField</a></u>	It inherits all attributes and methods from FileField, but also validates that the uploaded object is a valid image.
<u><a href="#">IntegerField</a></u>	It is an integer field. Values from -2147483648 to 2147483647 are safe in all databases supported by Django.

Field Name	Description
<u>GenericIPAddressField</u>	An IPv4 or IPv6 address, in string format (e.g. 192.0.2.30 or 2a02:42fe::4).
<u>NullBooleanField</u>	Like a BooleanField, but allows NULL as one of the options.
<u>PositiveIntegerField</u>	Like an IntegerField, but must be either positive or zero (0).
<u>PositiveSmallIntegerField</u>	Like a PositiveIntegerField, but only allows values under a certain (database-dependent) point.
<u>SlugField</u>	Slug is a newspaper term. A slug is a short label for something, containing only letters, numbers, underscores or hyphens. They're generally used in URLs.
<u>SmallIntegerField</u>	It is like an IntegerField, but only allows values under a certain (database-dependent) point.
<u>TextField</u>	A large text field. The default form widget for this field is a Textarea.
<u>TimeField</u>	A time, represented in Python by a datetime.time instance.
<u>URLField</u>	A CharField for a URL, validated by URLValidator.
<u>UUIDField</u>	A field for storing universally unique identifiers. Uses Python's UUID class. When used on PostgreSQL, this stores in a uuid datatype, otherwise in a char(32).

## Relationship Fields

Django also defines a set of fields that represent relations.

Field Name	Description
<u>ForeignKey</u>	A many-to-one relationship. Requires two positional arguments: the class to which the model is related and the on_delete option.
<u>ManyToManyField</u>	A many-to-many relationship. Requires a positional argument: the class to which the model is related, which works exactly the same as it does for ForeignKey, including recursive and lazy relationships.

Field Name	Description
<u>OneToOneField</u>	A one-to-one relationship. Conceptually, this is similar to a ForeignKey with unique=True, but the “reverse” side of the relation will directly return a single object.