

Типы полей формы

Последнее обновление: 20.08.2022



В формах Django мы можем использовать следующие классы для создания полей форм:

- **BooleanField**: создает поле `<input type="checkbox">`. Возвращает значение Boolean: True - если флажок отмечен и False - если флажок не отмечен.
- **NullBooleanField**: создает следующую разметку:

```

1 <select>
2 <option value="1" selected="selected">Unknown</option>
3 <option value="2">Yes</option>
4 <option value="3">No</option>
5 </select>

```

- **CharField**: предназначен для ввода текста и создает следующую разметку:

```

1 <input type="text">

```

Принимает следующие параметры:

- `max_length`: максимальная длина вводимого текста
- `min_length`: максимальная длина вводимого текста

- `strip`: при значении `True` (по умолчанию) начальные и конечные пробелы удаляются
- `empty_value`: значение, применяемое для представления пустого поля

- **EmailField**: предназначен для ввода адреса электронной почты и создает следующую разметку:

```
1 <input type="email">
```

Применяет те же параметры, что и `CharField`.

- **GenericIPAddressField**: предназначен для ввода IP-адреса в формате IPv4 или IPv6 и создает следующую разметку:

```
1 <input type="text">
```

- **RegexField (regex="регулярное_выражение")**: предназначен для ввода текста, который должен соответствовать определенному регулярному выражению. Создает текстовое поле:

```
1 <input type="text">
```

- **SlugField()**: предназначен для ввода текста, который условно называется "slug", то есть последовательность символов в нижнем регистре, чисел, дефисов и знаков подчеркивания. Создает текстовое поле:

```
1 <input type="text">
```

- **URLField()**: предназначен для ввода ссылок. Создает следующее поле:

```
1 <input type="url">
```

- **UUIDField()**: предназначен для ввода UUID (универсального уникального идентификатора). Создает следующее поле:

```
1 <input type="text">
```

- **ComboField(fields=[field1, field2,...])**: аналогичен обычному текстовому полю за тем исключением, что требует, чтобы вводимый текст соответствовал требованиям тех полей, которые передаются через параметр `fields`. Создает следующее поле:

```
1 <input type="text">
```

- **MultiValueField(fields=[field1, field2,...])**: предназначен для создания сложных компоновок, состоящих из нескольких полей.

- **FilePathField(path="каталог файлов")**: создает список select, который содержит все папки и файлы в определенном каталоге:

```
1 <select>
2   <option value="file1">folder/file1</option>
3   <option value="file2">folder/file2</option>
4   <option value="file3">folder/file3</option>
5   //.....
6 </select>
```

Параметры поля

- path: абсолютный путь к каталогу
 - recursive: если равно False (значение по умолчанию), то элемент отображает файлы только непосредственно из этой папки. Если равно True, то также отображается содержимое вложенных папок
 - match: регулярное выражение, которому должны соответствовать файлы для отображения
 - allow_files: представляет значение True (значение по умолчанию) или False. Указывает, должны ли быть включены файлы по определенному пути
 - allow_folders: представляет значение True (значение по умолчанию) или False. Указывает, должны ли быть включены папки по определенному пути
- **FileField()**: предназначен для выбора файла. Создает следующее поле:

```
1 <input type="file">
```

- **ImageField()**: предназначен также для выбора файла, но при этом добавляет ряд дополнительных возможностей. Создает следующее поле:

```
1 <input type="file">
```

- **DateField()**: предназначен установки даты. В создаваемое поле вводится текст, который может быть сконвертирован в дату, например, 2017-12-25 или 11/25/17. Создает следующее поле:

```
1 <input type="text">
```

- **TimeField()**: предназначен ввода времени, например, 14:30:59 или 14:30. Создает следующее поле:

```
1 <input type="text">
```

- **DateTimeField()**: предназначен ввода даты и времени, например, 2017-12-25 14:30:59 или 11/25/17 14:30. Создает следующее поле:

```
1 <input type="text">
```

- **DurationField()**: предназначен временного промежутка. Вводимый текст должен соответствовать формату "DD HH:MM:SS", например, 2 1:10:20 (2 дня 1 час 10 минут 20 секунд). Создает следующее поле:

```
1 <input type="text">
```

- **SplitDateTimeField()**: создает два текстовых поля для ввода соответственно даты и времени:

```
1 <input type="text" name="_0" >  
2 <input type="text" name="_1" >
```

- **IntegerField()**: предназначен для ввода чисел. Создает следующее поле:

```
1 <input type="number">
```

Параметры поля:

- `max_value`: максимально возможное значение
- `min_value`: минимально возможное значение
- `step_size`: шаг приращения числа

- **DecimalField()**: предназначен для ввода чисел. Создает следующее поле:

```
1 <input type="number">
```

Параметры поля:

- `max_value`: максимально возможное значение
- `min_value`: минимально возможное значение
- `max_digits`: максимальное количество цифр в числе
- `decimal_places`: максимальное количество знаков после запятой
- `step_size`: шаг приращения числа

- **FloatField()**: предназначен для ввода чисел. Создает следующее поле:

```
1 <input type="number">
```

Параметры поля:

- `max_value`: максимально возможное значение
- `min_value`: минимально возможное значение
- `step_size`: шаг приращения числа
- **ChoiceField(choices=кортеж_кортежей)**: генерирует список select, каждый из его элементов формируется на основе отдельного кортежа. Например, следующее поле:

```
1 languages = forms.ChoiceField(choices=((1, "English"), (2, "German"), (3, "Frei
```

будет генерировать следующую разметку:

```
1 <select name="languages" id="id_languages">
2     <option value="1">English</option>
3     <option value="2">German</option>
4     <option value="3">French</option>
5 </select>
```

- **TypedChoiceField(choices=кортеж_кортежей, coerce=функция_преобразования, empty_value=None)**: также генерирует список select на основе кортежа. Однако дополнительно принимает функцию преобразования, которая преобразует каждый элемент. И также принимает параметр `empty_value`, который указывает на значение по умолчанию.
- **MultipleChoiceField(choices=кортеж_кортежей)**: также генерирует список select на основе кортежа, как и `forms.ChoiceField`, добавляя к создаваемому полю атрибут `multiple="multiple"`. То есть список поддерживает множественный выбор.
- **TypedMultipleChoiceField(choices=кортеж_кортежей, coerce=функция_преобразования, empty_value=None)**: аналог `TypedChoiceField` для списка с множественным выбором.
- **JSONField()**: поле, которое принимает данные в формате json.

Принимает два дополнительных параметра:

- `encoder`: класс производный от `json.JSONEncoder` для сериализации данных в формат JSON. Значение по умолчанию - `json.JSONEncoder`.
- `decoder`: класс производный от `json.JSONDecoder` для десериализации данных из формата JSON. Значение по умолчанию - `json.JSONDecoder`.

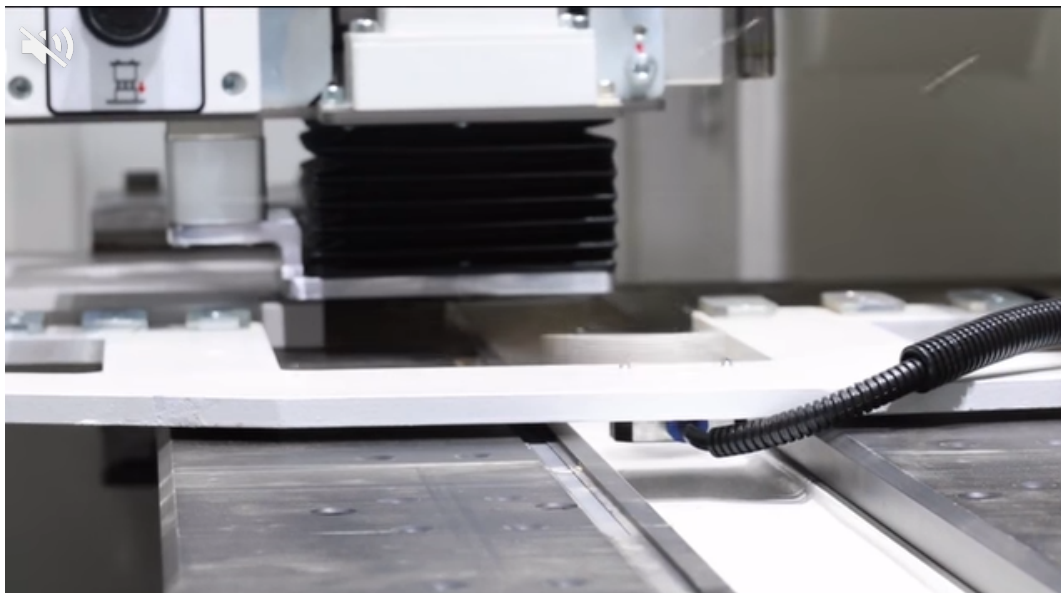
Виджеты Django

Выше рассмотренные поля при генерации разметки используют определенные виджеты из пакета **forms.widgets**. Например, класс CharField использует виджет forms.widgets.TextInput, а ChoiceField использует forms.widgets.Select. Но есть ряд виджетов, которые по умолчанию не используются полями форм, но тем не менее мы их можем задействовать:

- **PasswordInput**: генерирует поле для ввода пароля `<input type="password" >`
- **HiddenInput**: генерирует скрытое поле `<input type="hidden" >`
- **MultipleHiddenInput**: генерирует набор скрытых полей
- **TextArea**: генерирует многострочное текстовое поле `<textarea></textarea>`
- **RadioSelect**: генерирует список переключателей (радиокнопок) `<input type="radio" >`
- **CheckboxSelectMultiple**: генерирует список флажков `<input type="checkbox" >`
- **TimeInput**: генерирует поле для ввода времени (например, 12:41 или 12:41:32)
- **SelectDateWidget**: генерирует три поля select для выбора дня, месяца и года
- **SplitHiddenDateTimeWidget**: использует скрытое поле для хранения даты и времени
- **FileInput**: генерирует поле для выбора файла

Например, CharField по умолчанию использует виджет TextInput для создания однострочного текстового поля. Но что, если мы хотим создать многострочное текстовое поле? Для этой цели применим виджет **TextArea**:

```
1 comment = forms.CharField(widget=forms.Textarea)
```



РЕКЛАМА

**Присадочны
3 головы по**
Выгодная замена
Шеф-монтаж в п

 nanxing-stanki



→ [rodinlabs.kz](#) РЕКЛАМА

3D-печать металлом

3D-печать, изготовление прото-

ALSO ON METANIT.COM

Аутентификация и авторизация на ...

3 месяца назад • 2 коммент...

Аутентификация и авторизация в хабах и клиентах в SignalR в ...

Первое приложение на ASP.NET Core с ...

4 дня назад • 2 комментария

Начало работы с ASP.NET Core, установка .NET SDK, первое приложение на ...

ASP.NET и SignalR и C#

3 месяца назад • 4 коммент...

Клиентское приложение на .NET MAUI для SignalR в ASP.NET Core, ...

C# и

2 мес

Созда указат .NET,



Присоединиться к обсуждению...

Лучшее ▾



2

**Rostislav Malyshev** • 4 years ago

А как подключить PasswordInput и прочие функции, описанные в конце?

1



• Ответить • Поделиться ›

**Eugene Zabolotny** → Rostislav Malyshev • 4 years ago

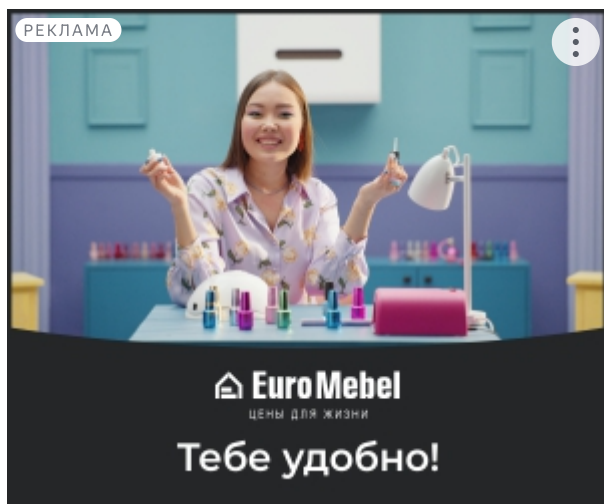
В следующей главе.



• Ответить • Поделиться ›

✉ Подписаться 🔒 Privacy ⚠ Не продавайте мои данные

РЕКЛАМА



Помощь сайту

YooMoney:

410011174743222

Перевод на карту

Номер карты:

4048415020898850

Номер карты:

4890494751804113

[Вконтакте](#) | [Телеграм](#) | [Twitter](#) | [Канал сайта на youtube](#) | [Помощь сайту](#).

Контакты для связи: metanit22@mail.ru

Copyright © metanit.com, 2012-2022. Все права защищены.

