

Настройка формы и ее полей

Последнее обновление: 20.08.2022



label

Свойство **label** позволяет установить текстовую метку, которая отображается рядом с полями. По умолчанию она отображает название самого поля с большой буквы. Например:

```

1 from django import forms
2
3 class UserForm(forms.Form):
4     name = forms.CharField(label="Имя")
5     age = forms.IntegerField(label="Возраст")
    
```

Эта форма сформирует следующие поля ввода:

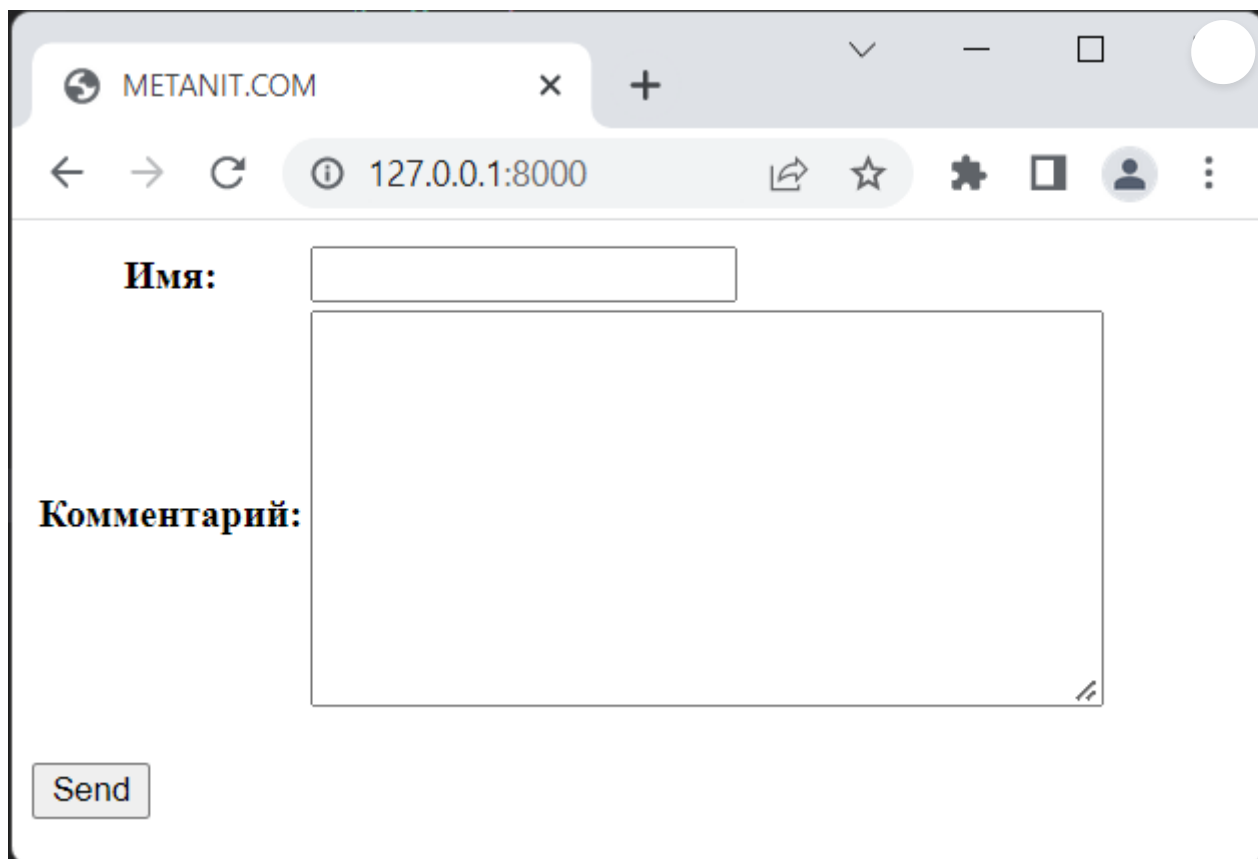
Скриншот веб-браузера, отображающий форму. В адресной строке видно METANIT.COM и IP-адрес 127.0.0.1:8000. Форма содержит два поля ввода: 'Имя:' и 'Возраст:'. Внизу виден фрагмент кнопки 'Send'.

widget

Параметр **widget** позволяет задать виджет, который будет использоваться для генерации разметки html:

```
1 from django import forms
2
3 class UserForm(forms.Form):
4     name = forms.CharField(label="Имя")
5     comment = forms.CharField(label="Комментарий", widget=forms.Textarea)
```

По умолчанию поле CharField использует виджет forms.widgets.TextInput, который создает однострочное текстовое поле. Однако если нам надо создать многострочное текстовое поле, то необходимо воспользоваться виджетом forms.Textarea:



The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000'. The page contains a form with two labels: 'Имя:' and 'Комментарий:'. The 'Имя:' label is followed by a single-line text input field. The 'Комментарий:' label is followed by a multi-line text area. At the bottom left of the form, there is a button labeled 'Send'.

Значения по умолчанию

С помощью параметра **initial** можно установить значения по умолчанию.

```
1 from django import forms
2
3 class UserForm(forms.Form):
4     name = forms.CharField(initial="undefined")
5     age = forms.IntegerField(initial=18)
```

Порядок полей

Поля ввода отображаются на веб-странице в том порядке, в котором они определены в классе формы. С помощью свойства **field_order** можно переопределить порядок, как в классе формы:

```
1 class UserForm(forms.Form):
2     name = forms.CharField()
3     age = forms.IntegerField()
4     field_order = ["age", "name"]
```

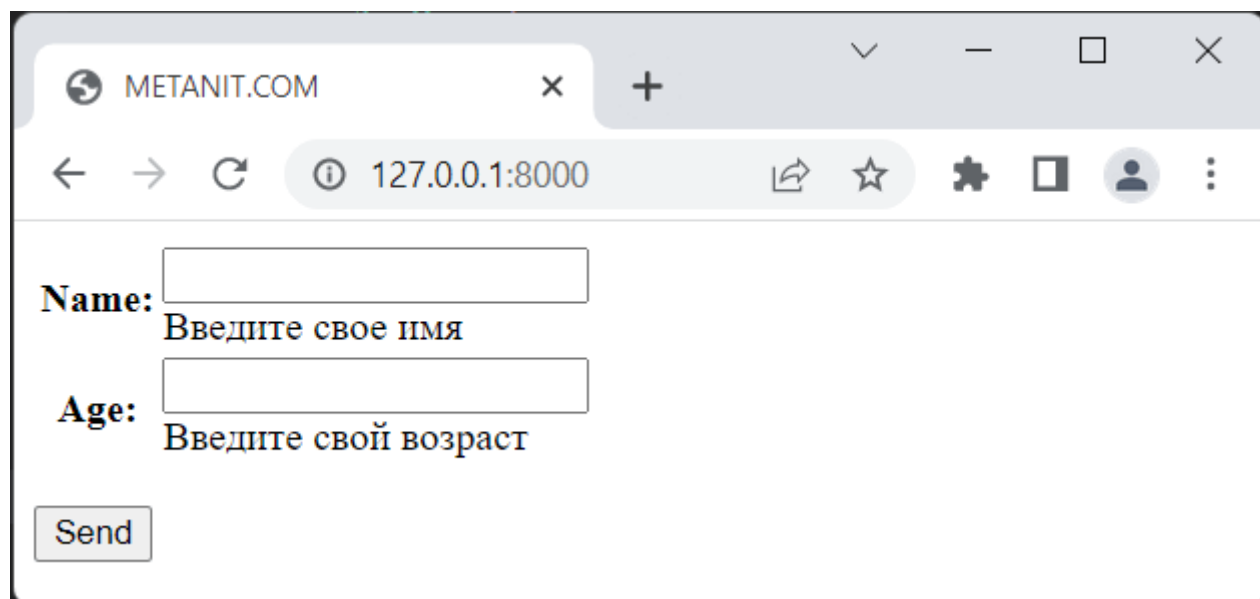
Так и при определении объекта формы в представлении:

```
1 def index(request):
2     userform = UserForm(field_order = ["age", "name"])
3     return render(request, "index.html", {"form": userform})
```

help_text

Параметр **help_text** устанавливает подсказку рядом с полем ввода:

```
1 from django import forms
2
3 class UserForm(forms.Form):
4     name = forms.CharField(help_text="Введите свое имя")
5     age = forms.IntegerField(help_text="Введите свой возраст")
```



The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000'. The page content includes a form with two input fields. The first field is labeled 'Name:' and has a text input box with the help text 'Введите свое имя' displayed below it. The second field is labeled 'Age:' and has a text input box with the help text 'Введите свой возраст' displayed below it. At the bottom left of the form is a button labeled 'Send'.

Настройка вида формы

С помощью специальных методов можно настроить общее отображение формы:

- **as_table()**: отображение в виде таблицы
- **as_ul()**: отображение в виде списка
- **as_p()**: каждое поле формы отображается в отдельном параграфе
- **as_div()**: каждое поле формы отображается в отдельном блоке div

Применение методов:

```
1  <h2>as_table</h2>
2  <form method="POST">
3      {% csrf_token %}
4      <table>
5          {{ form.as_table }}
6      </table>
7      <input type="submit" value="Send" >
8  </form>
9  <h2>as_ul</h2>
10 <form method="POST">
11     {% csrf_token %}
12     <ul>
13         {{ form.as_ul }}
14     </ul>
15     <input type="submit" value="Send" >
16 </form>
17 <h2>as_p</h2>
18 <form method="POST">
19     {% csrf_token %}
20     <div>
21         {{ form.as_p }}
22     </div>
23     <input type="submit" value="Send" >
24 </form>
```

Ⓢ

METANIT.COM

×

+

⌵

—

□

×

←

→

↻

ⓘ

127.0.0.1:8000

☆

⚙

□

👤

⋮

as_table

Name:

Age:

Send

as_ul

• Name:

• Age:

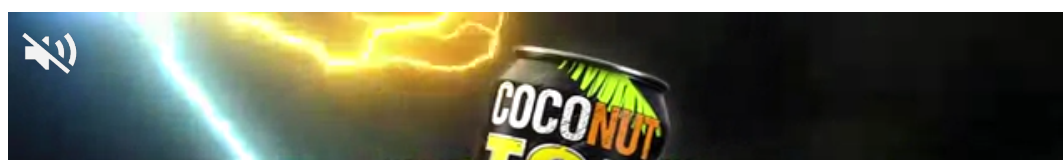
Send

as_p

Name:

Age:

Send



РЕКЛАМА

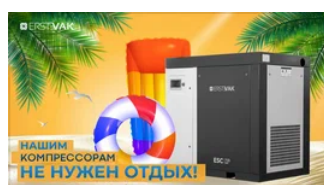
Больше инф
некпамолате



рекламодат

→ tornadoenergy

[Назад](#) [Содержание](#) [Вперед](#)



erstvak.com РЕКЛАМА

Полный аналог кс

Компрессоры в наличии!

ALSO ON METANIT.COM

Python и Tkinter

2 месяца назад • 4 коммент...

Создание кнопок на Python с помощью модуля tkinter, обработка нажатия кнопки

Первое приложение на ASP.NET Core с ...

4 дня назад • 2 комментариев

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

Основные операции с виджетом Text

2 месяца назад • 1 коммент...

Основные операции с виджетом Text в tkinter и Python, добавление, ...

C# и

2 меся

Созда указат .NET,

1 Комментарий

1 Haha Benis ▾



Присоединиться к обсуждению...

Лучшее ▾



2



Ilya Dvoeglazov (STEELMANRUS) • 4 года назад

Порядок полей проще выводить так:

в [views.py](#)

```
def index(request):  
    userform = UserForm()  
    return render(request, "index.html", {"form": userform})
```

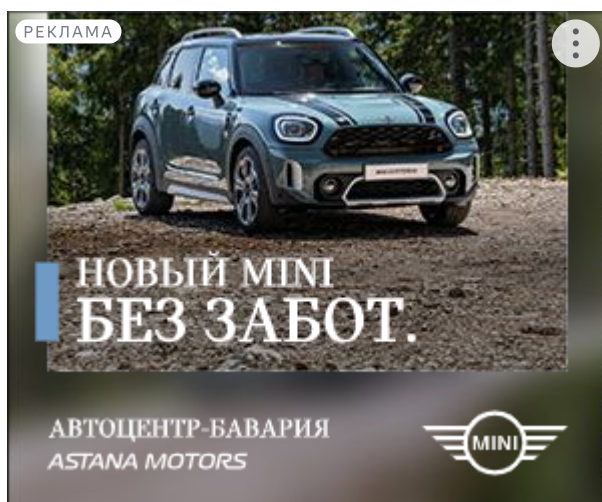
в шаблоне (index.html)

```
{{ form.age }}  
{{ form.name }}
```

Т.е. через точку передаем название атрибута name

^ | ▾ • Ответить • Поделиться ›

✉ Подписаться 🔒 Privacy ⚠ Не продавайте мои данные



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.