

Django 備忘

[[Django](#)][[Python](#)]

Model と Form

Form から Model のインスタンスを保存せずに取得

```
new_model = form.save(commit=False)
```

Model インスタンスから Form を生成

```
form = SampleForm(instance=new_model)
```

ウィジェットのオーバーライド

```
class SampleModel(models.Model):
    sample_field = CharField(max_length=10)

class CocomoForm(ModelForm):
    sample_field = CharField(widget=TextInput(attrs={'readonly': 'readonly'}))
    class Meta:
        model = Cocomo
```

バリデーション処理の呼び出し順

1. フィールドクラスの `clean()` メソッド
2. フォームクラスの `clean_{fieldname}()` メソッド。{fieldname} はフィールド名
3. フォームクラスの `clean()` メソッド

フィールドに依存しないエラー

- ・ Form の `clean()` メソッド内でハンドリングしたようなフィールドに依存しないエラーは、`forms.non_field_errors()` で取得できる
 - ・ フォーム定義

```
class CocomoForm(Form):
    def clean(self):
        f1 = self.cleaned_data['f1']
        try:
            f1 = float(f1)
            if f1 == 0.0:
                raise forms.ValidationError('フィールド 1 は 0 以外の値を設定')
        return self.cleaned_data
```

- ・ テンプレート側

```
{{form.non_field_errors}}
```

Textarea Widget に行列を指定する

```
from django.forms import *
class HogeForm(Form):
    data = CharField(widget=Textarea(attrs = dict(rows='40', cols='50')))
```

設定

DJANGO_SETTINGS_MODULE

- どの設定を使っているのかを Django に教える必要があり、環境変数 `DJANGO_SETTINGS_MODULE` を使う。
- `DJANGO_SETTINGS_MODULE` は Python のモジュールパス構文、たとえば `mysite.settings` のようにする。
- 設定モジュールは Python のモジュール検索パスにある必要がある。