

ModelForm

Model から Form を生成する .
Model の Field に合わせて自動的に Form 用の Field に変換してくれる .
Meta クラスとして使用する Model , 対象 Field を設定する .
ソースは app フォルダ内に forms.py ファイルを作成し , その中に記述する .

Listing 1 forms.py 記述例

```
1 from django.forms import ModelForm
2
3 from .models import Question
4
5
6 class QuestionForm(ModelForm):
7     class Meta:
8         model = Question
9         fields = ('question_text',)
```

views.py からの呼び出し

form 作成時にユーザから送付されてきた data を渡し , データの妥当性を確認する . このときに instance を渡すことでレコードの編集ができる .
ユーザからのデータは request.GET もしくは request.POST に入っている .
データの検証は is_valid メソッドで行う . 文字列長のオーバーなど , 不正なデータがある場合は False を返し , errors メンバ変数にエラーの内容を辞書で格納する .
save メソッドは commit=False という引数を渡すことで DB への保存を保留できる .

Listing 2 views.py 記述例

```
1 from django.shortcuts import render, redirect, get_object_or_404
2
3 from .models import Question
4 from .forms import QuestionForm
5
6
7 def edit(request, pk=None):
8     if pk is not None:
9         obj = get_object_or_404(Question, pk=pk)
10    else:
11        obj = None
12    if request.method == 'POST':
13        form = QuestionForm(data=request.POST, instance=obj)
14        if form.is_valid():
15            obj = form.save()
16            return redirect('polls:detail', pk=obj.pk)
17    else:
18        form = QuestionForm(instance=obj)
19    return render(request, 'polls/edit.html', {
20        'form': form,
21        'obj': obj,
22    })
```