

# まとめ

一般的なFormクラスの作成・・・forms.Formを継承して作成する

例) class UserForm(forms.Form):

テンプレートでのFormの表示・・・{{ form }}として、{% csrf\_token %}と共に表示

{{ form.as\_p }}: <p>でラップ。{{ form.as\_ul }}: <ul><li>でラップ。{{ form.as\_table }}: <tr>でラップ

**form = FormClass(request.POST):** POSTリクエストをFormの型に変換

**form.is\_valid()** formの中のフィールドが正しいかバリデーション

フォームのカスタマイズ・・・label=〇〇(ラベルを変更する)。required=True or False(必須かどうかを変更する)。widget=〇〇(表示する形式を変更する)。initial=〇〇(初期値を変更する)。  
max\_length, min\_length(最大の長さ、最小の長さを変更)。attrs={'placeholder':}: プレースホルダーを追加

# まとめ

フォームにID, クラスを追加・・・①. `attrs={'class': '○○', 'id'='××'}`として、フィールド作成時に追加する。②. `__init__`をオーバーライドする、`self.fields['フィールド名'].widget.attrs['id'] = 'ID名'`, `self.fields['フィールド名'].widget.attrs['class'] = 'クラス名'`

フォームのバリデーション・・・①. `clean_フィールド名`のメソッドをクラスの中に追加する。②. フィールド作成時に`validators=[]`を属性に追加する。③. `clean`メソッドを追加して、複数のフィールドに対してバリデーションを追加する

**ModelForm**・・・ModelとFormを連携して、モデルへのデータの挿入・更新などを簡素化する

フォームの各要素を表示する・・・`{{ form.フィールド名 }}`(フィールドの要素を表示)。`{{ form.フィールド名.label }}`(フィールドのラベルを表示)。`{{ form.field.errors }}`(フィールドのエラーを表示)

`{{ field.non_field_errors }}`・・・フィールド単体でないエラーを表示する

`{{ form.errors }}`・・・フォームの持つ全エラー

# まとめ

**HTMLの外だし** ・ ・ ・ {% include '○○.html' with 変数 %} includeで別のHTMLを表示する(withで値を渡す)

**Formset**(from django.forms import formset\_factory) ・ ・ ・ 複数のフォームを一度に表示して利用する

**ModelFormset**(from django.forms import modelformset\_factory) ・ ・ ・ 複数のModelFormを一度に表示して利用する

**ファイルのアップロード** ・ ・ ・ MEDIA\_URL(メディアファイルの公開時のURL)。

MEDIA\_ROOT(ファイルの保存先の指定)

**テンプレートの実装**

<form method="post" enctype="multipart/form-data"> ・ ・ ・ </form>

**ビューの実装**

myfile = request.FILES['myfile']

fs = FileSystemStorage() # ファイル保存用のインスタンス

filename = fs.save(myfile.name, myfile) # ファイルを保存する

uploaded\_file\_url = fs.url(filename) # ファイルのURL

# まとめ

## Modelを用いたファイルのアップロード

### # models.py

```
class Image(models.Model):
```

```
    picture = models.FileField(upload_to='documents/') # アップロードするファイルの格納先を保存する(upload_to='documents/%Y/%m/%d/' とすると、documents/年/月/日で保存できる)
```

### # views.py

```
form = ImageForm(request.POST, request.FILES) # 保存するファイルを第2引数に置く
```

```
if form.is_valid():
```

```
    form.save() # ファイルが保存され、ファイルのパスがDBに書き込まれる
```

```
    return redirect('home')
```

### # 画像を表示する

```

```