

# まとめ

# Templateの使用方法

`render('HTMLファイルの配置先', context={'キー': '渡す値'})`

## Template

`<h1>{{ value }}</h1>`



## 設定ファイル(settings.py)にディレクトリを指定する

```
TEMPLATES = [ {  
    'BACKEND': 'django.template.backends.django.DjangoTemplates',  
    'DIRS': [], # templateとして利用するディレクトリのパスを記述  
    -- 以下略 --
```

# DTLの制御文

## 処理の記述方法

{% %}: if, forなどの式(制御文)を記載

{{ }}: 値をアウトプットする

{# #}: コメント文

## for文

```
{% for value in mylist %}  
  <p>{{ value }}</p>  
{% endfor %}
```

## コメント文

```
{# コメント文は画面上に表示されません #}
```

## if文

```
{% if value in mylist %}  
  <p>something</p>  
{% elif 式 %}  
  :  
{% else %}  
  <p>Hmmm</p>  
{% endif %}
```

# templateの継承

```
{% extends "base.html" %}
```

```
{% block content %}
```

```
{% endblock %}
```

# テンプレートフィルタ

```
{{ 変数等 | フィルタ関数 }}
```

```
{% filter ○○ %}
```

フィルターを適用

```
{% endfilter %}
```

# template間の画面遷移

## 画面遷移

```
<a href="{% url 'app:home' val1='val1' val2='val2' %}">Link</a>
```

```
app_name = 'app'  
urlpatterns = [  
    path('home/<val1>/<val2>/', views.home, name='home'),  
]
```

## 静的コンテンツの使用

```
{% load static %}  

```

静的コンテンツを配置するフォルダの変更

```
STATICFILES_DIRS = [  
    'staticフォルダのパス'  
]
```