



통합 구현(Spring, Django)

# Django 템플릿 알아보기



한국기술교육대학교  
온라인평생교육원

## 학습내용

- 템플릿 기본 문법
- 템플릿 상속
- 정적 파일 처리

## 학습목표

- Django에서 템플릿을 구성하는 방법에 대해 설명할 수 있다.
- Django에서 템플릿을 상속하는 방법에 대해 설명할 수 있다.
- Django에서 정적 파일을 처리하는 방법을 파악할 수 있다.

# 템플릿 기본 문법

## 1 템플릿 태그

### 1> 템플릿 태그

Django의 템플릿 태그는 html코드 내에서 동적 화면 구성을 위해 쓰임

if, for 등의 프로그래밍 문법이 있고 python과는 다름

템플릿 태그를 먼저 해석하여 문법을 실행하고 최종적으로는 html 파일로 나타냄

### 2> 기본 문법

#### 1 {{ 변수 }}

View에서 전달하는 변수를 화면에 보이게 할 때 쓰임

객체일 경우 도트(.) 표현식도 가능

```
<h1>{{ section.title }}</h1>
```

# 템플릿 기본 문법

## 1 템플릿 태그

### 2 기본 문법

2 {% for A in B %}

기본 반복문으로 컬렉션 B 안의 원소를  
하나씩 꺼내어 A에 대입한 후  
사용할 때 쓰임

반복문 종료 후에는  
{% endfor %}로 닫아주어야 함

```
{% for choice in question.choice_set.all %}
<span> {{choice.choie_text}} </span><br/>
{% endfor %}
```

3 {% if 변수 %} {% elif 변수 %} {% else %} {% endif %}

기본 조건문으로 조건 변수가 true(혹은 not empty)인지를 판단하여 구문을 실행함

>와 < 등의 연산자도 사용할 수 있음

조건문 종료 후에는 {% endif %}로 닫아주어야 함

## 템플릿 기본 문법

### 1 템플릿 태그

#### 2 기본 문법

3 {% if 변수 %} {% elif 변수 %} {% else %} {% endif %}

```
{% if athlete_list %}
    Number of athletes: {{ athlete_list|length }}
{% elif athlete_in_locker_room_list %}
    Athletes should be out of the locker room soon!
{% else %}
    No athletes.
{% endif %}
```

4 {% # %}

- 기본 한 줄 주석으로 템플릿 코드가 주석 안에 있어도 정상적으로 주석처리 됨

```
{# Hello #}, World !
```

5 {% comment %} {% endcomment %}

- 기본 여러 줄 주석으로 템플릿 코드가 주석 안에 있어도 정상적으로 주석처리 됨

```
{% comment %}, World !
Hello, Python!
Hello, Django!
{% endcomment %}
```

## 템플릿 기본 문법

### 2 템플릿 필터

#### 템플릿 필터

- 템플릿 필터는 객체나 변수에 추가적인 옵션을 적용하여 결과 값을 추려냄
- 기본적으로 변수(객체)값이 있어야 하며 필터를 여러 개 연결할 수도 있음
- 기본 문법은 {{ 변수 | 필터명 }}과 같이 쓰임

#### 자주 사용되는 필터

```
{{ value|default:
"nothing" }}
```

만약 변수가 비어있거나 false이면 “nothing”이라는 string이 표시됨

```
{{ value|length }}
```

변수가 컬렉션이라면 길이를 표시함

```
{{ value|join: “-” }}
```

변수의 원소 사이에 “-” 글자를 넣어 이어 붙임

## 템플릿 상속

### 1 레이아웃

#### 1 레이아웃 구성

대부분의 웹사이트는 비슷한 레이아웃을 가짐

하나의 웹 사이트의 각 페이지는 동일한 레이아웃을 가짐

페이지마다 일정 부분 동일한 코드를 가지며 이는 재사용되어야 함

#### 2 기본 레이아웃

Head

웹 페이지 제목, CSS, 각종 웹 프로그래밍 언어 선언문 위치

Menu

주 메뉴로 모든 페이지마다 항상 나타나야 함

Contents

주 콘텐츠로 페이지마다 다르게 나타남

Footer

저작권, 사이트맵, js 파일 등의 위치

## 템플릿 상속

### 1 레이아웃

#### 3 기본 레이아웃 템플릿 구성

```
<!DOCTYPE html>
<html>
  <head>
    <title>Django Framework</title>
  </head>
  <body>
    <nav class="menu">
      <ul>
        <li><a href="/">Home</a></li>
        <li><a href="/polls">Polls</a></li>
      </ul>
    </nav>
    <div class="contents"></div>
    <footer>
      This is footer
    </footer>
  </body>
</html>
```



## 템플릿 상속

### 2 템플릿 상속 방법

#### 1 부모 템플릿 작성

- 1 Django에서는 템플릿 상속을 통해 코드를 재사용할 수 있음
- 2 재사용에 쓰일 부모 템플릿을 작성해야 하며, 이 부모 템플릿을 상속하여 각 페이지마다 화면을 다르게 표현함
- 3 부모 템플릿은 block 태그를 통해 상속받는 템플릿에서 콘텐츠를 바꿀 부분을 지정해줌
- 4 자식 템플릿에서 block 태그 안의 내용을 채우지 않으면 부모 템플릿에 있는 내용이 나타남

#### 2 부모 템플릿(base.html) 구성 예

mysite/polls/templates/polls/base.html 파일 작성

{% block 블럭이름 %} {% endblock %} 형태로 블럭 지정

## 템플릿 상속

### 2 템플릿 상속 방법

#### 2 부모 템플릿(base.html) 구성 예

```
<!DOCTYPE html>
<html>
  <head>
    <title>{% block title %}Django Framework{% endblock %}</title>
  </head>
  <body>
    <nav class="menu">
      {% block menu %}
      <ul>
        <li><a href="/">Home</a></li>
        <li><a href="/polls">Polls</a></li>
      </ul>
    </nav>
    {% endblock %}

    <div class="contents">{% block contents %}{% endblock %}</div>
    <footer>
      {% block footer %}This is footer{% endblock %}
    </footer>
  </body>
</html>
```

## 템플릿 상속

### 2 템플릿 상속 방법

#### 3 자식 템플릿(child.html) 구성 예

1

부모와 같은 폴더에 child.html을 작성하고 extends 태그를 통해 부모 템플릿 (base.html)을 상속

2

자식 템플릿에서 변경사항이 없는 블록은 구현하지 않아도 됨

3

자식 템플릿에서 변경사항이 있는 블록만 따로 block 태그를 통해 구현

```
{% extends 'polls/base.html' %}

{% block title %} This is child {% endblock %}
{% block contents %}
child contents
{% endblock %}
```

## 정적 파일 처리

### 1 정적 파일이란?

#### 1 정적 파일

서버에서 생성된 HTML을 제외하고,  
웹 어플리케이션은 일반적으로  
전체 웹 페이지를 렌더링하는 데  
필요한 추가 파일  
(이미지, Javascript, CSS 등)

Django Framework에서는  
하나의 프로젝트 내에 여러 개의  
어플리케이션이 존재하고,  
각 어플리케이션마다의 정적 파일을  
관리하기 쉽지 않음

#### 2 정적 파일 위치

1

Django Framework에서는 각 어플리케이션 폴더 밑에 'static'이라는 이름의 폴더를  
생성한 후 정적 파일을 위치시킴

2

기존에 'templates' 폴더를 생성한 것과 동일함

3

예를 들어 'custom.css'을 추가할 경우 경로는  
'mysite/polls/static/polls/custom.css'와 같이 설정함

## 정적 파일 처리

### 2 정적 파일 설정

#### 1 {% load static %} 태그

- 1 html 파일 상단에 {% load static %} 태그를 삽입하면 정적 파일의 절대 주소 생성
- 2 정적 파일의 주소가 'mysite/polls/static/polls/custom.css'와 같을 때, 다음과 같은 태그로 정적 파일을 로드할 수 있음
  - `<link rel="stylesheet" type="text/css" href="{% static 'polls/style.css' %}" />`
- 3 Django 프로젝트 내에서 생성한 파일이 아닌 외부 정적 파일(CDN 경로)은 Static 태그를 이용할 수 없음

#### 2 정적 파일 설정 적용

```
<!DOCTYPE html>
<html>
  <head>
    {% load static %}
    <link rel="stylesheet" href="{% static 'admin/css/base.css' %}" />
    <title>{% block title %}Django Framework{% endblock %}</title>
  </head>
  <body>
    <nav class="menu">
      {% block menu %}
      <ul>
        <li><a href="/">Home</a></li>
        <li><a href="/polls">Polls</a></li>
      </ul>
      {% endblock %}
    </nav>

    <div class="contents">{% block contents %}{% endblock %}</div>
    <footer>
      {% block footer %}This is footer{% endblock %}
    </footer>
  </body>
</html>
```

### 1. 템플릿 기본 문법

- 템플릿 태그
  - Django의 템플릿 태그는 html코드 내에서 동적 화면 구성을 위해 쓰임
  - if, for 등의 프로그래밍 문법이 있고 python과는 다름
  - 템플릿 태그를 먼저 해석하여 문법을 실행하고 최종적으로는 html 파일로 나타냄
- 템플릿 필터
  - 템플릿 필터는 객체나 변수에 추가적인 옵션을 적용하여 결과 값을 추려냄
  - 기본적으로 변수(객체)값이 있어야 하며 필터를 여러 개 연결할 수도 있음
  - 기본 문법은 {{ 변수 | 필터명 }}과 같이 쓰임

### 2. 템플릿 상속

- 레이아웃
  - 대부분의 웹사이트는 동일한 레이아웃을 가짐
  - 하나의 웹 사이트의 각 페이지는 동일한 레이아웃을 가짐
  - 페이지마다 일정 부분 동일한 코드를 가지며 이는 재사용되어야 함
- 부모 템플릿 작성
  - Django에서는 템플릿 상속을 통해 코드를 재사용할 수 있음
  - 그러기 위해서는 재사용에 쓰일 부모 템플릿을 작성해야 하며, 이 부모 템플릿을 상속하여 각 페이지 마다 화면을 다르게 표현함
  - 부모 템플릿은 block 태그를 통해 상속받는 템플릿에서 콘텐츠를 바꿀 부분을 지정해줌
  - 자식 템플릿에서 block 태그 안의 내용을 채우지 않으면 부모 템플릿에 있는 내용이 나타남

### 3. 정적 파일 처리

- 정적 파일이란?
  - 서버에서 생성된 HTML을 제외하고, 웹 어플리케이션은 일반적으로 전체 웹 페이지를 렌더링하는데 필요한 추가 파일(이미지, Javascript, CSS 등)
  - Django Framework에서는 하나의 프로젝트 내에 여러 개의 어플리케이션이 존재하고, 각 어플리케이션마다의 정적 파일을 관리하기 쉽지 않음
- {% load static %} 태그
  - html 파일 상단에 {% load static %} 태그를 삽입하면 정적 파일의 절대 주소를 생성함
  - 정적 파일의 주소가 'mysite/polls/static/polls/custom.css'와 같을 때, 다음과 같은 태그로 정적 파일을 로드할 수 있음  
`<link rel="stylesheet" type="text/css" href="{% static 'polls/style.css' %}" />`
  - Django 프로젝트 내에서 생성한 파일이 아닌 외부 정적 파일(CDN 경로)은 static 태그를 이용할 수 없음