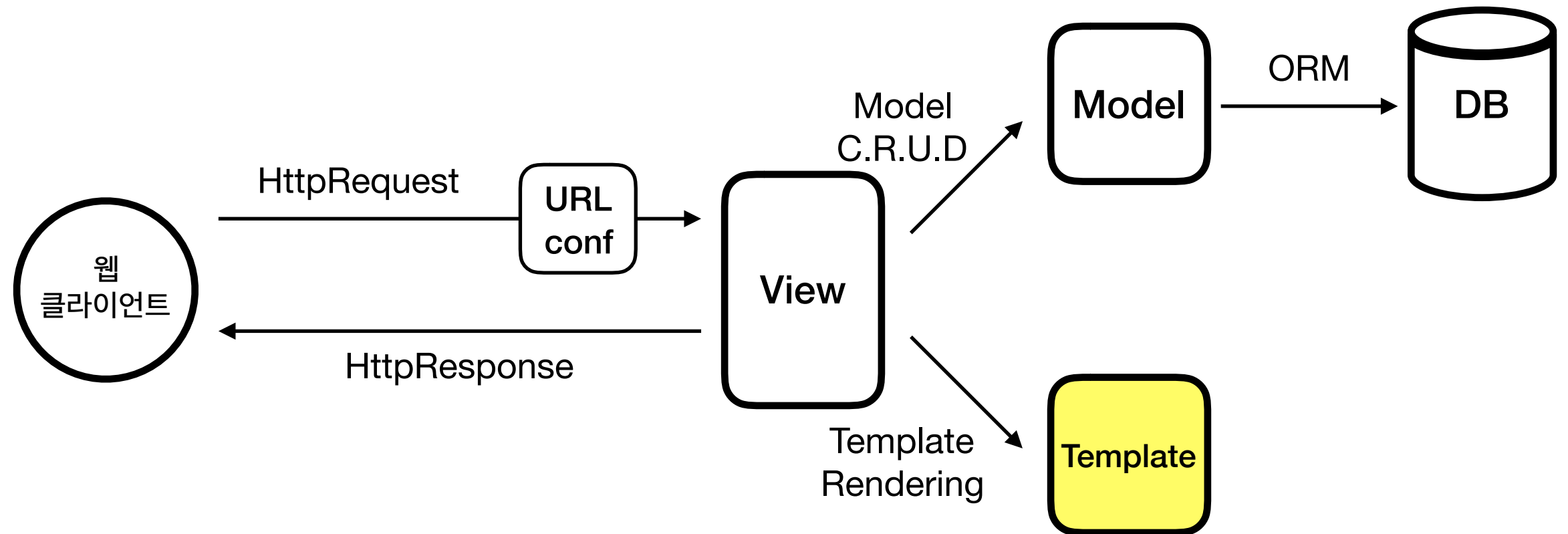


# 웹 프로그래밍 실전

# 6. Template 작성

# MTV 계층



# 템플릿 시스템

- 템플릿에서는 로직을 표현하는 것이 아니라 사용자에게 어떻게 보여줄지에 대한 룩앤필을 표현한다.
- 템플릿 : 개발자가 작성하는 \*.html 파일
- 템플릿 코드를 작성 시에 HTML 코드와 장고의 템플릿 코드가 섞인다.

# 템플릿 변수

- `{{ variable }}`
- 템플릿 시스템은 변수를 평가하여 변수값으로 출력한다.
- 템플릿 문법에서 도트(.)를 만나면 장고는 찾기(lookup)를 시도한다.
  - `{{ foo.bar }}`
    1. `foo`가 사전 타입인지 확인한다. 그렇다면 `foo['bar']`로 해석한다.
    2. `foo`의 속성을 찾는다. `bar`라는 속성이 있으면 `foo.bar`로 해석한다.
    3. `foo`가 리스트인지 확인한다. 그렇다면 `foo[bar]`로 해석한다.

# 템플릿 태그

- {% tag %}
- 어떤 태그는 시작 태그와 끝 태그 둘 다 있어야 한다.
- 텍스트 결과물을 만든다.
- 템플릿 로직을 제어한다.
- 외부 파일을 템플릿 내로 로딩한다.

# {% if %} 태그

- 변수를 평가하여 True이면 바로 아래 문장이 표시된다.

```
{% 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 %}
```

# {% for %} 태그

- 리스트에 담겨 있는 항목들을 순회하며 출력할 수 있다.

<ul>

{% for athlete in athlete\_list %}

<li>{{ athlete.name }}</li>

{% endfor %}

<ul>



# {% for %} 태그에 사용되는 변수들

변수명	설명
forloop.counter	현재까지 루프를 실행한 루프 카운터(1부터 카운트)
forloop.revcounter	루프 끝에서 현재가 몇 번째인지 카운트한 숫자(1부터 카운트)
forloop.first	루프에서 첫 번째 실행이면 True 값을 가짐
forloop.last	루프에서 마지막 실행이면 True 값을 가짐
forloop.parentloop	중첩된 루프에서 현재의 루프 바로 상위의 루프를 의미함

# polls/templates/polls/index.html

```
{% if latest_question_list %}
```

```
<ul>
```

```
    {% for question in latest_question_list %}
```

```
        <li><a href="/polls/{{ question.id }}/">{{ question.question_text }}</a></li>
```

```
    {% endfor %}
```

```
</ul>
```

```
{% else %}
```

```
    <p>No polls are available.</p>
```

```
{% endif %}
```

# 폼(Form)

- 사용자로부터 입력을 받기 위해 사용한다.
- `<form></form>` 사이에 있는 요소(element)들의 집합이다.
- 사용자는 폼을 통해 텍스트를 입력하거나 항목을 선택할 수 있다.
- 폼에 입력된 데이터는 서버로 보내진다.
- action 속성 : 폼 데이터를 어디로 보낼지 지정한다.
- method 속성 : 어떤 HTTP 메서드로 보낼지 지정한다.
  - 폼에서 사용할 수 있는 HTTP 메서드는 GET과 POST뿐이다.
  - 폼 데이터를 전송할 때는 POST 방식을 사용한다.

# {% csrf\_token %} 태그

- POST 방식의 <form>을 사용하는 템플릿 코드에서는 CSRF(Cross Site Request Forgery) 공격을 방지하기 위하여 {% csrf\_token %} 태그를 사용한다.
- <form> 요소의 첫 줄 다음에 넣어준다.
- <form action="." method="post">{% csrf\_token %}
- 이 태그를 사용하면 장고는 내부적으로 CSRF 토큰값의 유효성을 검증하여 실패할 경우 사용자에게 403 에러를 보여준다.

## NOTE

CSRF는 사이트 간 요청 위조 공격이다. 웹 사이트의 취약점을 공격하는 방식 중 하나로 특정 웹 사이트에서 이미 인증을 받은 사용자를 이용하여 공격한다. 인증 받은 사용자가 공격 코드가 삽입된 페이지를 열면 공격 대상이 되는 웹 사이트는 위조된 공격 명령이 믿을 수 있는 사용자로부터 발송된 것으로 판단하고 공격 받는다.

# {% url %} 태그

- URL을 하드코딩하는 것을 방지한다.
- {% url %} 태그를 사용하면 URL이 변경되더라도 URLconf만을 참조하여 원하는 URL을 추출할 수 있다.
- {% url 'namespace:view-name' arg1 arg2 %}
- namespace : urls.py 파일의 include() 함수 또는 app\_name 변수에 정의한 이름 공간(namespace) 이름
- view-name : urls.py 파일에서 정의한 URL 패턴 이름
- argN : 뷰 함수에서 사용하는 인자로 없을 수도 있고 여러 개일 수도 있음

# polls/templates/polls/detail.html

```
<h1>{{ question.question_text }}</h1>

{% if error_message %}<p><strong>{{ error_message }}</strong></p>{% endif %}

<form action="{% url 'polls:vote' question.id %}" method="post">

    {% csrf_token %}

    {% for choice in question.choice_set.all %}

        <input type="radio" name="choice" id="choice{{ forloop.counter }}" value="{{ choice.id }}" />

        <label for="choice{{ forloop.counter }}">{{ choice.choice_text }}</label><br />

    {% endfor %}

    <input type="submit" value="Vote" />

</form>
```

# question.choice\_set.all

- Question 테이블과 Choice 테이블의 관계는 1:N
  - 외래키로 연결되어 있다.
- 1:N 관계 : 1 테이블에 연결된 N 테이블의 항목들
  - xxx\_set 속성을 디폴트로 제공한다.
- 즉 question.choice\_set.all 이란 Question 테이블의 question 객체에 연결된 Choice 테이블의 객체 모두를 뜻한다.
- 템플릿 문법상 메서드 호출을 표시하는 ()는 사용하지 않는다.