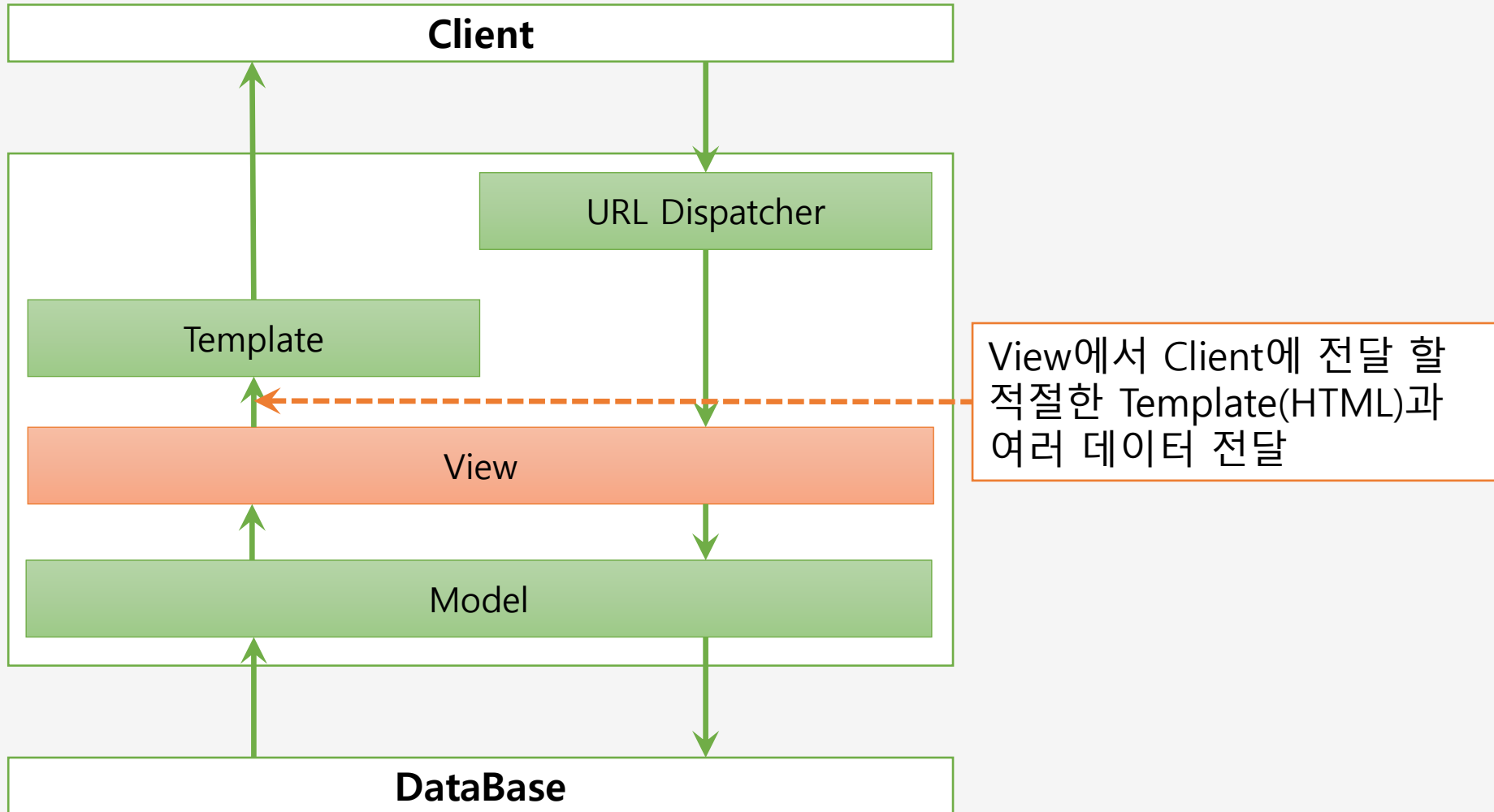


Django Template

- Template 역할
- Template Setting
- Template Directory Structure
- View와 Template 연결
- Template Code
- Template Filter
- HTML Code

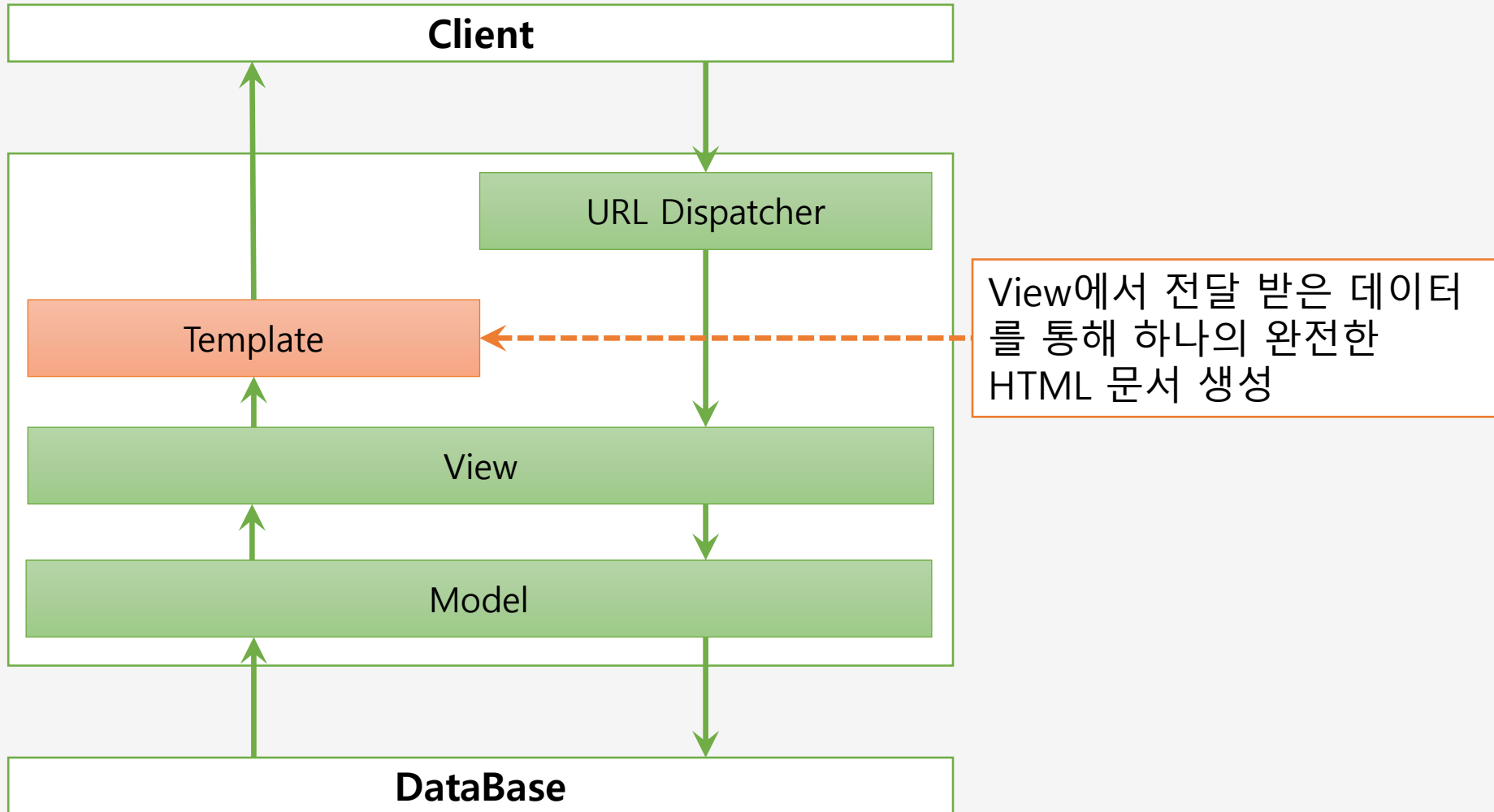
Template

1. Template 역할 (1/2)



Template

1. Template 역할 (2/2)



Template

2. Template Setting

mainApp > settings.py

```
TEMPLATES = [  
    {  
        'BACKEND': 'django.template.backends.django.DjangoTemplates',  
        'DIRS': [os.path.join(BASE_DIR, 'template')],  
        'APP_DIRS': True,  
        .....  
    }  
]
```

'DIRS' : Template 폴더의 위치를 절대 경로로 설정

os.path.join(BASE_DIR, 'template') 코드를 입력 한다.

'APP_DIRS' : Project 안의 template 폴더 부터 검색 후 존재 하지 않는 경우
APP 안의 templates 폴더 검색 할 것인지의 여부 설정

Template

3. Template Directory Structure (1/2)

```
myWeb
├── myWeb
│   └── ...
├── sub_app
│   └── ...
└── template
    ├── myWeb
    │   ├── index.html
    │   └── other.html
    └── sub_app
        ├── index.html
        └── other.html
```

Template

3. Template Directory Structure (2/2)

```
myWeb
├── myWeb
│   └── templates
│       └── myWeb
│           ├── index.html
│           └── other.html
├── sub_app
│   └── templates
│       └── sub_app
│           ├── index.html
│           └── other.html
└──
```

Template

4. View와 Template 연결 (1/2)

```
from django.shortcuts import render
```

```
def mainIndex(request):  
    .....  
    .....  
    return render(request, 'template_path')
```

Template의 경로는 settings.py의 TEMPLATES에 설정 한 폴더가 기본 폴더로 사용되고 있기 때문에 설정된 폴더 부터의 경로만을 'template_path'로 작성하면 된다.

Template

4. View와 Template 연결 (2/2)

```
from django.shortcuts import render

def mainIndex(request):
    context = {
        ...
        ...
    }
    return render(request, 'template_path', context)
```

단순히 HTML 파일만을 사용자에게 넘기고자 하는 경우 전달하고자 하는 Template 파일의 경로만 넣어주면 되지만, Template에 적용해야 하는 Data가 존재하는 경우 사전형 자료로 자료를 구성하여 전달해야 한다.

Template

5. Template Code (1/11)

```
from django.shortcuts import render

def mainIndex(request):
    context = {
        'key1': 'string value',
        'key2': 10000,
        'key3': 100.0,
        'key4': [1, 2, 3, 4],
        'key5': {'a':1, 'b':2, 'c':3},
    }
    return render(request, 'template_path', context)
```

Template

5. Template Code (2/11)

`{{ }}` : 변수를 사용하고자 할 때 사용하는 Template Tag

`{% %}` : if 또는 for문 등 특별한 기능을 하는 코드를 작성 할 때 사용하는 Template Tag

`{# #}` : 주석을 위한 Template Tag, 한 줄에 대한 주석 처리만 가능

- # Django에서 Template Code는 가급적 복잡한 코드가 사용되는 것을 원치 않는다. 즉, if문 또는 for문을 지원하지만 복잡한 로직(Logic)이 구현되는 것은 지양하게 된다.
- # Template은 사용자에게 보여지기 위한 문서의 구조와 디자인을 위한 것이지 복잡한 Logic을 구현하기 위한 페이지가 아니다.
- # Template을 개발하는 개발자는 Python 코드를 모르는 개발자일 가능성이 많기 때문에 코드의 사용이 어려워서는 안 된다.

Template

5. Template Code (3/11)

Template 변수

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML 문서</title>
</head>
<body>
  {{ key1 }} | {{ key2 }} | {{ key3 }}<br>
  {{ key4 }} | {{ key5 }}<br>
</body>
</html>
```

Template

5. Template Code (4/11)

Template 변수

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML 문서</title>
</head>
<body>
  {{ key4.0 }} | {{ key4.1 }} | {{ key4.2 }}<br>
  {{ key5.a }} | {{ key5.c }}<br>
</body>
</html>
```

Template

5. Template Code (5/11)

Template if 문

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML 문서</title>
</head>
<body>
  {% if condition %}
    ...
  {% endif %}
</body>
</html>
```

Template

5. Template Code (6/11)

Template if 문

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML 문서</title>
</head>
<body>
  {% if condition %}
    ...
  {% else %}
    ...
  {% endif %}
</body>
</html>
```

Template

5. Template Code (7/11)

Template if 문

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML 문서</title>
</head>
<body>
  {% if condition %}
    ...
  {% elif condition %}
    ...
  {% else %}
    ...
  {% endif %}
</body>
</html>
```

Template

5. Template Code (8/11)

Template for 문

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML 문서</title>
</head>
<body>
  {% for x in variable %}
    ...
  {% endfor %}
</body>
</html>
```


Template

5. Template Code (9/11)

Template for 문

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML 문서</title>
</head>
<body>
  {% for x in variable %}
    {{ forloop.counter }}      # 1-indexed
    ...
  {% endfor %}
</body>
</html>
```

Template

5. Template Code (10/11)

Template for 문

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML 문서</title>
</head>
<body>
  {% for x in variable %}
    {{ forloop.counter0 }}      # 0-indexed
    ...
  {% endfor %}
</body>
</html>
```

Template

5. Template Code (11/11)

Template for 문

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML 문서</title>
</head>
<body>
  {% for x in variable %}
    {% for y in variable %}
      {{ forloop.parentloop.counter0 }}      # 중첩반복의 부모 반복문
      ...
    {% endfor %}
  {% endfor %}
</body>
</html>
```

Template

6. Template Filter (1/9)

Template Filter

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML 문서</title>
</head>
<body>
  {{ variable|add:'2' }}
</body>
</html>
```

Template

6. Template Filter (2/9)

Template Filter

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML 문서</title>
</head>
<body>
  {{ variable|date:'D d M Y' }}    {# 월요일 16 10월 2017 #}
</body>
</html>
```

Template

6. Template Filter (3/9)

y : 2 자리 년도 표시
Y : 4 자리 년도 표시
m : 01 ~ 12 월 표시
M : 1 ~ 12 월 표시, 월 문자 추가
d : 01 ~ 31 일 표시
j : 1 ~ 31 일 표시
D : 한 주의 요일 표시, 영문자의 경우 3자리 문자 표시
a 또는 A : 오전/오후 표시
g : 1 ~ 12 시간 표시
G : 0 ~ 23 시간 표시
h : 01 ~ 12 시간 표시
H : 00 ~ 23 시간 표시
i : 00 ~ 59 분 표시
s : 00 ~ 59 초 표시

Template

6. Template Filter (4/9)

Template Filter

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML 문서</title>
</head>
<body>
  {{ variable|default:'없음' }}    {# 변수가 False로 평가 되면, 기본 값 사용 #}
</body>
</html>
```

Template

6. Template Filter (5/9)

Template Filter

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML 문서</title>
</head>
<body>
  {{ variable|dictsort:'key' }}    {# 리스트 자료의 사전 요소들에 대한 정렬 #}
</body>
</html>
```


Template

6. Template Filter (6/9)

Template Filter

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML 문서</title>
</head>
<body>
  {{ variable|dictsortreversed:'key' }}    {# 리스트 자료의 사전 요소들에 대한 정렬 #}
</body>
</html>
```

Template

6. Template Filter (7/9)

Template Filter

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML 문서</title>
</head>
<body>
  {{ variable|slice:':2' }}    {# 리스트 자료에 대한 슬라이싱 #}
</body>
</html>
```

Template

6. Template Filter (8/9)

Template Filter

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML 문서</title>
</head>
<body>
  {{ variable|truncatechars:10 }}    {# 10 자리 초과 문자는 ...으로 표시 #}
</body>
</html>
```

Template

6. Template Filter (9/9)

Template Filter

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML 문서</title>
</head>
<body>
  {{ variable|truncatewords:2 }}    {# 2 단어 초과는 ...으로 표시 #}
</body>
</html>
```

Template

7. HTML Code (1/18)

<h1~6></h1~6>

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML 문서</title>
</head>
<body>
  <h1>제목 1</h1>
  <h2>제목 2</h2>
  <h3>제목 3</h3>
  <h4>제목 4</h4>
  <h5>제목 5</h5>
  <h6>제목 6</h6>
</body>
</html>
```

Template

7. HTML Code (2/18)

`<div></div>`

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML 문서</title>
</head>
<body>
  <div>영역 분리1</div>
  <div>영역 분리2</div>
</body>
</html>
```

Template

7. HTML Code (3/18)

<p></p>

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML 문서</title>
</head>
<body>
  <p>문자열 단락</p>
</body>
</html>
```

Template

7. HTML Code (4/18)

`<a>`

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML 문서</title>
</head>
<body>
  <a href="http://127.0.0.1">링크 이동</a>
</body>
</html>
```


Template

7. HTML Code (5/18)

`, `

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML 문서</title>
</head>
<body>
  <ul>
    <li>목록 1</li>
    <li>목록 2</li>
    <li>목록 3</li>
  </ul>
</body>
</html>
```

Template

7. HTML Code (6/18)

`, `

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML 문서</title>
</head>
<body>
  <ol>
    <li>목록 1</li>
    <li>목록 2</li>
    <li>목록 3</li>
  </ol>
</body>
</html>
```

Template

7. HTML Code (7/18)

`<table></table>, <tr></tr>, <td></td>`

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML 문서</title>
</head>
<body>
  <table>
    <tr><td>1행 1셀</td>
      <td>1행 2셀</td></tr>
    <tr><td>2행 1셀</td>
      <td>2행 2셀</td></tr>
  </table>
</body>
</html>
```

Template

7. HTML Code (8/18)

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML 문서</title>
</head>
<body>
  
</body>
</html>
```

Template

7. HTML Code (9/18)

`<input>`

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML 문서</title>
</head>
<body>
  <input type="text" name="var">
</body>
</html>
```

Template

7. HTML Code (10/18)

`<input>`

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML 문서</title>
</head>
<body>
  <input type="password" name="var">
</body>
</html>
```

Template

7. HTML Code (11/18)

`<input>`

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML 문서</title>
</head>
<body>
  <input type="date" name="var">
</body>
</html>
```

Template

7. HTML Code (12/18)

<input>

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML 문서</title>
</head>
<body>
  <input type="checkbox" name="var1" value="val1">
  <input type="checkbox" name="var1" value="val2">
  <input type="checkbox" name="var1" value="val3">
</body>
</html>
```


Template

7. HTML Code (13/18)

<input>

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML 문서</title>
</head>
<body>
  <input type="radio" name="var1" value="val1">
  <input type="radio" name="var1" value="val2">
  <input type="radio" name="var1" value="val3">
</body>
</html>
```

Template

7. HTML Code (14/18)

`<input>`

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML 문서</title>
</head>
<body>
  <input type="button" value="button">
</body>
</html>
```

Template

7. HTML Code (15/18)

`<input>`

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML 문서</title>
</head>
<body>
  <input type="submit" value="submit">
</body>
</html>
```

Template

7. HTML Code (16/18)

<select></select>, <option></option>

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML 문서</title>
</head>
<body>
  <select name="val">
    <option value="opt1">옵션 1</option>
    <option value="opt2">옵션 2</option>
    <option value="opt3">옵션 3</option>
  </select>
</body>
</html>
```

Template

7. HTML Code (17/18)

<textarea></textarea>

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML 문서</title>
</head>
<body>
  <textarea name="val" cols="50" rows="30"></textarea>
</body>
</html>
```

Template

7. HTML Code (18/18)

<form>

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML 문서</title>
</head>
<body>
  <form action="request_url" method="POST">
    {% csrf_token %}
    <!-- 사용자 입력 양식 -->
  </form>
</body>
</html>
```

Template

8. 문제

1. `http://127.0.0.1:8000/html/tag/hn/` 의 주소로 접근하는 경우 `h1 ~ h6` 태그 예제가 출력 될 수 있도록 하시오.
2. `http://127.0.0.1:8000/html/tag/div/` 의 주소로 접근하는 경우 `div` 태그 예제가 출력 될 수 있도록 하시오.
3. `http://127.0.0.1:8000/html/tag/p/` 의 주소로 접근하는 경우 `p` 태그 예제가 출력 될 수 있도록 하시오.
4. `http://127.0.0.1:8000/html/tag/a/` 의 주소로 접근하는 경우 `a` 태그 예제가 출력 될 수 있도록 하시오.
5. 위의 페이지들에서 목록으로 돌아가기 위한 기능을 `a` 태그를 사용하여 추가하고 목록 표시는 `http://127.0.0.1:8000/html/tag/` 에서 표시가 될 수 있도록 하시오.

Template

8. 문제

다음 양식에 맞추어 HTML 문서가 사용자에게 전달 되도록 하시오.

	TV 시청	인터넷 검색	산책	게임	음악 감상	헬스	잡담 통화	낮잠	등산	스포츠 간접 관람
남자	46.5%	13.2%	3.6%	6.8%	2.0%	3.1%	1.7%	1.1%	1.3%	2.0%
여자	56.1%	9.7%	5.3%	1.3%	3.0%	1.8%	3.1%	1.4%	0.8%	0.1%

HTML Table Tag를 사용하시오.

Template Tag를 반드시 사용하시오.