

Django Template Language: The Need

---

# Django Template Language: The Need

Faisal Memon

## What are we using currently?

---

```
HttpResponse("<h1>Hello world</h1>")
```

## Important points to keep in mind

---

- Django's approach to building web applications was to keep design separate from the logic. Logic here is the actual business logic that is written in python.
- Web designers don't need to learn a new language to be able to work on Django related projects.

## Behind the scenes and best practices

---

```
\authentication  
  \templates  
    base.html
```

```
\store  
  \templates  
    base.html
```

## Recommended way

---

/templates/<app-name> folder

## Behind the scenes and best practices

---

```
\authentication
  \templates
    \authentication
      base.html
\store
  \templates
    \store
      base.html
```

## Namespacing

render() function

---

# render() function

Faisal Memon

# Syntax

---

`render(<request-object>, <template>, <context>)`



## What are tags?

---

# What are tags?

Faisal Memon

## What are tags?

---

- A tag is a piece of code in django template language.

Syntax → {% %}

# Logic that can be defined using template tags

---

- **Conditional logic:** if...else statements
- **Loops:** for loop
- **Specifying imports:** import tag
- **Performing inheritance:** extends tag
- **Declaring blocks:** block tag

## If...else statements

---

# If...else statements

Faisal Memon

## Syntax:

---

```
{% if condition %}  
# render if body  
{% endif %}
```

# For loops

---

# For loops

Faisal Memon

# Syntax

---

```
{% for odd in odds %}  
# logic here  
{% endfor %}
```

## Writing comments in templates

---

# Comments in templates

Faisal Memon



# Syntax

---

```
{% comment %}
```

Comment here

```
{% endcomment %}
```

# Behind the scenes of rendering templates

```
TEMPLATES = [  
    {  
        'BACKEND': 'django.template.backends.django.DjangoTemplates',  
        'DIRS': [],  
        'APP_DIRS': True,  
        'OPTIONS': {  
            'context_processors': [  
                'django.template.context_processors.debug',  
                'django.template.context_processors.request',  
                'django.contrib.auth.context_processors.auth',  
                'django.contrib.messages.context_processors.messages',  
            ],  
        },  
    },  
]
```

# Namespacing

```
✓ jobapp
  ✓ app
    > __pycache__
    > migrations
    ✓ templates / app
      <> hello.html
      <> index.html
      <> job_detail.html
```

- It's not recommended to store your templates directly in /templates folder
- But instead its recommended to store them in /templates/<app-name> folder

# Variables

---

```
jobapp > app > templates > app > <> job_detail.html
```

```
1  <a href="{% url 'jobs_home' %}">  
2      <h4>Home</h4>  
3  </a>  
4  
5  <h1>{{job_title}}</h1>  
6  <h3>{{job_description}}</h3>
```

## render() function

---

```
def hello(request):  
    list = ["alpha", "beta"]  
    temp = TempClass()  
    is_authenticated = False  
    context={"name":"Django", "age":30, "first_list":list, "temp_object":temp,  
            "is_authenticated":is_authenticated}  
    return render(request, "app/hello.html", context)
```

# Tags

```
{% if is_authenticated %}  
    <h1>Hello {{name}}</h1>  
{% else %}  
    <h1>Hello User</h1>  
{% endif %}  
<h1>Age: {{age}}</h1>  
<h2>{{ first_list.0 }}</h2>  
<h2>{{temp_object.x}}</h2>  
  
{% for x in first_list %}  
    <p>First: {{x}}</p>  
{% endfor %}
```

- Conditional logic: if...else statements
- Loops: for loop

# Filters

---

```
<h1>Filters</h1>
{{first_list|length}}
{{first_list|length_is:"3"}}
```

# Comments

---

```
{% comment "first_test_comment" %}  
This is learning material  
{% endcomment %}
```



## URL tag

---

```
<ul>
    {% for job in job_title_list %}
    <li>
        <a href="{% url 'jobs_detail' forloop.counter0 %}"><h1>{{job}}</h1></a>
    </li>
    {% endfor %}
</ul>
```