

Templates

Variables

Used to access some values from the context dictionary. If the variable passed in the context dictionary is a callable then the callable will be called with no arguments.

All variables need to be used with the `{{ }}`.

Dictionary lookup, attribute lookup and list-index lookups are implemented with a dot notation:

```
{{ my_dict.key }}  
{{ my_object.attribute }}  
  
{{ my_list.0 }}
```

Tags

Used to access something other than variables for example if else or for loop.

Requires the tag to be wrapped in the `{% %}`

```
{% for k, v in defaultdict.items %}  
    Do something with k and v here...  
{% endfor %}
```

Filters

Django provides us built in filters so that any text that is passed can be modified according to the filter.

```
{{ django | title }}
```

TemplateInheritance

The most powerful – and thus the most complex – part of Django’s template engine is template inheritance. Template inheritance allows you to build a base “skeleton” template that contains all the common elements of your site and defines blocks that child templates can override.

Whatever we write inside the block can be overridden by the child. Everything else is fixed.

```
<title>{% block title %}My amazing site{% endblock %}</title>
```

```
{% extends "base.html" %}
{% block title %}My amazing blog{% endblock %}
```

For Loops

```
<ul>
{% for athlete in athlete_list %}

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

{% endfor %}

</ul>
```

If, elif and else

```
{% 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 %}
```

```
{% if athlete_list|length > 1 %}  
    Team: {% for athlete in athlete_list %} ... {% endfor %}  
  
{% else %}  
    Athlete: {{ athlete_list.0.name }}  
  
{% endif %}
```

We can even use this for checking length however this is the only exception where filter returns an integer. All other returns string.

Comments

```
{# Anything Written here is a comment #}
```

In Django, if we use the default template engine then all outputs are escaped.