

Templates

Django Template Language

The Django Template Language, DTL, interpolates Python into HTML.

```
<html>
  <head>
    {% block title %}
    {% endblock %}
  </head>
  <body>
    {% block content %}
    {% endblock %}
  </body>
</html>
```

Templates

Templates are what users see on a web page.

```
<h1>Page title</h1>
```

Template Name Space

Templates are stored in .

Template Tags

Template tags are used to inject Python content.

```
{{% block content %}}
{{% endblock %}}
```

Template Blocks

Content tags can be used to extend templates to reduce repeated code.

```
<!DOCTYPE html>

<html>

  <head>

    {% block headBlock %}

    {% endblock %}

  </head>

  <body>

    <h1>Hello World!</h1>

    {% block content %}

    {% endblock %}

  </body>

</html>
```

Base Templates

A base template can be used as the basis for other templates using the `extends` tag.

```
{% extends "base.html" %}
```

Static Files

Static files, such as CSS, images, and JavaScript files, are placed in a separate location in the project— **/static/** by default.

```
myProject/
├── myProject/
├── myapp/
└── static/
```

Loading Static Files

Templates can load static files using the `{% load static %}` tag in the header of the HTML file. Specific files can be loaded by specifying the exact file path as an argument.

```
{% load static "custom.css" %}
```

Template Variables

Variables can be displayed using variable tags.

```
{{ variable_name }}
```

Conditional Tags

Templates can include logic using `if` statements.

```
{% if conditional_variable %}

{% endif %}
```

for-in Loops

Template variables can be iterated over using `for - in` loops inside template tags.

```
<ul>
  {% for item in dictionary %}
    <li>{{ item.value }}</li>
  {% endfor %}
</ul>
```

URL Tag

URLs can be linked using the `url` tag.

```
{% url "some-other-page" %}
```

Passing Variables in URL Tags

URL tags can include arguments to pass between webpages.

```
{% url "blog-view" blog.id %}
```

Template Filters

Filters can be applied with the `|` symbol.

```
{{ variable_name | filter_name }}
```

Filter Arguments

Arguments can be passed to filters using the `:` symbol after the filter name.

```
{{ variable_name |
filter_name:"argument" }}
```

Sorting Dictionaries

Dictionaries can be sorted using the `dictsort` filter.

```
{{ dictionary_name | dictsort:"key_name" }}
```

Datatypes in Filters

Some filters require variables of a certain data type.

```
{{ date_time | time"H": "i" }}
```