

Django Templates

Cruz Arredondo Jose David. Tijuana Technological University. November 03, 2023

Summary – Django templates are a fundamental component of the Django web framework for Python, enabling developers to create dynamic, data-driven web applications. These templates serve as a bridge between Python code in the backend and HTML code for frontend presentation.

I. INTRODUCTION

I. Django templates are a powerful and versatile tool provided by the Django web framework for Python developers. They serve as a bridge between the backend logic of a web application and the frontend presentation, allowing developers to create dynamic and interactive web pages. Django templates follow a simple and intuitive syntax, enabling the embedding of Python-like code directly into HTML, making it easier to generate dynamic content.

II. DJANGO TEMPLATES CONCEPT

Django templates are a powerful tool used in Django web development framework to generate dynamic HTML content. They allow you to create HTML dynamically by embedding Python-like expressions inside double curly braces {}, which are then replaced with actual values when the template is rendered.



III. DJANGO TEMPLATES EXAMPLES

:

Variables: {{ variable_name }} - Variables enclosed within double curly braces are replaced with their corresponding values when the template is rendered.

Tags: {% tag %} - Tags are used for control flow and logic within templates. For example, {% for item in items %} ... {% endfor %} is a loop tag.

```
{% if condition %}
    display this
{% else %}
    display that
{% endif %}
```

Filters: {{ variable|filter }} - Filters allow you to format variables or perform other operations on them before they are displayed. For example, {{ name|title }} capitalizes the first letter of each word in the 'name' variable.

Template Inheritance: Django templates support inheritance, allowing you to create a base template with common structure and then extend it in other templates.

TEMPLATE TAGS AND FILTERS: DJANGO PROVIDES A WIDE

```
<!-- base_template.html -->
<html>
<body>
    {% block content %}
    {% endblock %}
</body>
</html>
```

```
{% for item in items %}
    {{ item }}
{% endfor %}
```

RANGE OF BUILT-IN TEMPLATE TAGS AND FILTERS THAT OFFER VARIOUS FUNCTIONALITIES, SUCH AS DATE FORMATTING, STRING MANIPULATION, AND MORE. YOU CAN ALSO CREATE YOUR OWN CUSTOM TAGS AND FILTERS.

ESCAPING: DJANGO TEMPLATES AUTOMATICALLY ESCAPE VARIABLES TO PREVENT COMMON SECURITY VULNERABILITIES LIKE CROSS-SITE SCRIPTING (XSS) ATTACKS. IF YOU WANT TO OUTPUT HTML CODE UNESCAPED, YOU CAN USE THE SAFE FILTER.

CONTEXT: WHEN RENDERING A TEMPLATE, YOU PASS A CONTEXT, WHICH IS A DICTIONARY CONTAINING VARIABLE NAMES AND THEIR CORRESPONDING VALUES. THESE VARIABLES ARE THEN ACCESSIBLE IN THE TEMPLATE.

IV. CONCLUSION

Django templates stand at the core of Django's web development framework, offering a powerful and flexible way to create dynamic, data-driven web applications. Their simplicity, combined with their rich feature set, makes them an essential tool for developers building robust and interactive websites. In this detailed conclusion, let's delve deeper into the various aspects of Django templates and their significance in web development.

REFERENCES

- [1] Django (no date) Django Templates, Django Project. Available at: <https://docs.djangoproject.com/en/4.2/topics/templates/> (Accessed: 03 November 2023).
- [2] GeekForGeeks (ed.) (2021) Django templates, GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/django-templates/> (Accessed: 03 November 2023).
- [3] w3schools (no date) Django Templates, Django templates. Available at: https://www.w3schools.com/django/django_templates.php (Accessed: 03 November 2023).