



[Django](#) [Views](#) [Model](#) [Template](#) [Forms](#) [Jinja](#) [Python SQLite](#) [Flask](#) [Json](#) [Postman](#) [Interview Ques](#)

Django Template Tags

Last Updated : 17 May, 2024

Django Web Framework ships with dozens of tags used to implement arbitrary logics right in the template. Tags look like this:

```
{% tag %}
```

. Tags are more complex than variables: Some create text in the output, some control flow by performing loops or logic, and some load external information into the template to be used by later variables. Tags provide arbitrary logic in the rendering process. For example, a tag can output content, serve as a control structure e.g. an “if” statement or a “for” loop, grab content from a database, or even enable access to other template tags.

Syntax

```
{% tag_name %}
```

Example

Tags are surrounded by {% and %} like this:

```
{% csrf_token %}
```

Most tags accept arguments, for example :

```
{% cycle 'odd' 'even' %}
```

Commonly used Tags in Django Templates

1. Comment

Template ignores everything between {% comment %} and {% endcomment %}. An optional note may be inserted in the first tag. For example, this is useful when commenting out code for documenting why the code was disabled.

Example

```
{% comment "Optional note" %}
```

```
Commented out text with {{ create_date|date:"c" }}
```

```
{% endcomment %}
```

To check more about comment tag, visit – [comment – Django template tags](#)

2. cycle

It produces one of its arguments each time this tag is encountered. The first argument is produced on the first encounter, the second argument on the second encounter, and so forth. Once all arguments are exhausted, the tag cycles to the first argument and produces it again.

Example This tag is particularly useful in a loop:

html

```
1 {% for o in some_list %}
2     <tr class="{% cycle 'row1' 'row2' %}">
3         ...
4     </tr>
5 {% endfor %}
```

The first iteration produces HTML that refers to class row1, the second to row2, the third to row1 again, and so on for each iteration of the loop. To check more about cycle tag, visit – [cycle – Django Template Tags](#)

3. extends

extends tag is used for inheritance of templates in django. One needs to repeat the same code again and again. Using extends we can inherit templates as well as variables.

Example Assume the following directory structure:

```
dir1/
    template.html
    base2.html
    my/
        base3.html
    base1.html
```

In template.html, the following paths would be valid:

```
{% extends "../base2.html" %}
{% extends "../base1.html" %}
{% extends "../my/base3.html" %}
```

To check more about extends tag, visit – [extends – Django Template Tags](#)

4. if

The {% if %} tag evaluates a variable, and if that variable is “true” (i.e. exists, is not empty, and is not a false boolean value) the contents of the block are output.

Example

html

```
1 {% if athlete_list %}
2     Number of athletes: {{ athlete_list|length }}
3 {% elif athlete_in_locker_room_list %}
4     Athletes should be out of the locker room soon!
5 {% else %}
6     No athletes.
7 {% endif %}
```

In the above, if athlete_list is not empty, the number of athletes will be displayed by the {{ athlete_list|length }} variable. As one can see, the if tag may take one or several {% elif %} clauses, as well as an {% else %} clause that will be displayed if all previous conditions fail. These clauses are optional. To check more about if tag, visit – [if – Django Template Tags](#)

5. for loop

for tag loops over each item in an array, making the item available in a context variable. **Example** For example, to display a list of athletes provided in athlete_list:

html

```
1 <ul>
2 {% for athlete in athlete_list %}
3     <li>{{ athlete.name }}</li>
4 {% endfor %}
```

```
5 </ul>
```

To check more about for loop tag, visit – [for loop – Django Template Tags](#)

6. for ... empty loop

for tag loops over each item in an array, making the item available in a context variable. The for tag can take an optional `{% empty %}` clause whose text is displayed if the given array is empty or could not be found. This is basically used as a condition to be followed to check if queryset is empty and what action to be performed in the same scenario.

Example

html

```
1 <ul>
2 {% if athlete_list %}
3     {% for athlete in athlete_list %}
4         <li>{{ athlete.name }}</li>
5     {% endfor %}
6 {% else %}
7     <li>Sorry, no athletes in this list.</li>
8 {% endif %}
9 </ul>
```

To check more about for ... empty loop tag, visit – [for ... empty loop – Django Template Tags](#)

7. Boolean Operators

The `{% if %}` tag evaluates a variable, and if that variable is “true” (i.e. exists, is not empty, and is not a false boolean value) the contents of the block are output. One can use various boolean operators with Django If Template tag.

Example

html

```
1 <ul>
2 {% if variable boolean_operator value %}
3     // statements
4 {% endif %}
```

```
5 </ul>
```

To check more about Boolean Operators, visit – [Boolean Operators – Django Template Tags](#)

8. firstof

firstof tag Outputs the first argument variable that is not “false” (i.e. exists, is not empty, is not a false boolean value, and is not a zero numeric value).

Outputs nothing if all the passed variables are “false”.

Example

```
{% firstof var1 var2 var3 %}
```

This is equivalent to:

```
{% if var1 %}
    {{ var1 }}
{% elif var2 %}
    {{ var2 }}
{% elif var3 %}
    {{ var3 }}
{% endif %}
```

One can also use a literal string as a fallback value in case all passed variables are False:

```
{% firstof var1 var2 var3 "fallback value" %}
```

To check more about firstof tag, visit – [firstof – Django Template Tags](#)

9. include

include tag loads a template and renders it with the current context. This is a way of “including” other templates within a template. The template name can either be a variable or a hard-coded (quoted) string, in either single or double quotes.

Example

```
{% include "foo/bar.html" %}
```

Normally the template name is relative to the template loader’s root directory. A string argument may also be a relative path starting with ./ or ../

as described in the [extends](#) tag. To check more about include tag, visit – [include – Django Template Tags](#)

10. lorem

lorem tag displays random “lorem ipsum” Latin text. This is useful for providing sample data in templates.

Example

- `{% lorem %}` will output the common “lorem ipsum” paragraph.
- `{% lorem 3 p %}` will output the common “lorem ipsum” paragraph and two random paragraphs each wrapped in HTML tags.
- `{% lorem 2 w random %}` will output two random Latin words.

To check more about lorem tag, visit – [lorem – Django Template Tags](#)

11. now

now tag displays the current date and/or time, using a format according to the given string. Such string can contain format specifiers characters as described in the [date filter](#) section.

Example

```
It is {% now "D d M Y" %}
```

Above tag will display, Tue 04 Feb 2020 To check more about now tag, visit – [now – Django Template Tags](#)

12. url

url tag Returns an absolute path reference (a URL without the domain name) matching a given view and optional parameters. This is a way to output links without violating the DRY principle by having to hard-code URLs in your templates.

Example

```
{% url 'some-url-name' v1 v2 %}
```

The first argument is a URL pattern name. It can be a quoted literal or any other context variable. Additional arguments are optional and should be space-separated values that will be used as arguments in the URL. To check more about url tag, visit – [url – Django Template Tags](#)

Are you ready to elevate your web development skills from foundational knowledge to advanced expertise? Explore our [Mastering Django Framework - Beginner to Advanced Course](#) on GeeksforGeeks, designed for aspiring developers and experienced programmers. This comprehensive course covers everything you need to know about Django, from the basics to advanced features. Gain practical experience through **hands-on projects** and real-world applications, mastering essential Django principles and techniques. Whether you're just starting or looking to refine your skills, this course will empower you to build sophisticated web applications efficiently. Ready to enhance your web development journey? Enroll now and unlock your potential with Django!

N Nave...



12

Previous Article

[variables - Django Templates](#)

Next Article

[extends - Django Template Tags](#)

Similar Reads

now - Django Template Tags

A Django template is a text document or a Python string marked-up using the Django template language. Django being a powerful Batteries included...

2 min read

comment - Django template tags

A Django template is a text document or a Python string marked-up using the Django template language. Django being a powerful Batteries included...

2 min read

cycle - Django Template Tags

A Django template is a text document or a Python string marked-up using the Django template language. Django being a powerful Batteries included...

3 min read

extends - Django Template Tags

A Django template is a text document or a Python string marked-up using the Django template language. Django being a powerful Batteries included...

2 min read

if - Django Template Tags

A Django template is a text document or a Python string marked-up using the Django template language. Django being a powerful Batteries included...

3 min read

for ... empty loop - Django Template Tags

A Django template is a text document or a Python string marked-up using the Django template language. Django being a powerful Batteries included...

3 min read

for loop - Django Template Tags

A Django template is a text document or a Python string marked-up using the Django template language. Django being a powerful Batteries included...

3 min read

Boolean Operators - Django Template Tags

A Django template is a text document or a Python string marked-up using the Django template language. Django being a powerful Batteries included...

5 min read

firstof - Django Template Tags

A Django template is a text document or a Python string marked-up using the Django template language. Django being a powerful Batteries included...

3 min read

include - Django Template Tags

A Django template is a text document or a Python string marked-up using the Django template language. Django being a powerful Batteries included...

2 min read

Article Tags :

PythonDjango-templatesPython Django

Practice Tags :

python



Corporate & Communications Address:-
A-143, 9th Floor, Sovereign Corporate
Tower, Sector- 136, Noida, Uttar Pradesh
(201305) | Registered Address:- K 061,
Tower K, Gulshan Vivante Apartment,
Sector 137, Noida, Gautam Buddh
Nagar, Uttar Pradesh, 201305



Company

- About Us
- Legal
- In Media
- Contact Us
- Advertise with us
- GFG Corporate Solution
- Placement Training Program
- GeeksforGeeks Community

DSA

- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- Top 100 DSA Interview Problems
- DSA Roadmap by Sandeep Jain
- All Cheat Sheets

Languages

- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial
- Tutorials Archive

Data Science & ML

- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Web Technologies

HTML
CSS
JavaScript
TypeScript
ReactJS
NextJS
Bootstrap
Web Design

Computer Science

Operating Systems
Computer Network
Database Management System
Software Engineering
Digital Logic Design
Engineering Maths
Software Development
Software Testing

System Design

High Level Design
Low Level Design
UML Diagrams
Interview Guide
Design Patterns
OOAD
System Design Bootcamp
Interview Questions

School Subjects

Mathematics
Physics
Chemistry
Biology
Social Science
English Grammar
Commerce
World GK

Python Tutorial

Python Programming Examples
Python Projects
Python Tkinter
Web Scraping
OpenCV Tutorial
Python Interview Question
Django

DevOps

Git
Linux
AWS
Docker
Kubernetes
Azure
GCP
DevOps Roadmap

Interview Preparation

Competitive Programming
Top DS or Algo for CP
Company-Wise Recruitment Process
Company-Wise Preparation
Aptitude Preparation
Puzzles

GeeksforGeeks Videos

DSA
Python
Java
C++
Web Development
Data Science
CS Subjects

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved