



Django Template Filters

Last Updated : 12 Jan, 2023

Django Template Engine provides filters which are used to transform the values of variables and tag arguments. We have already discussed major [Django Template Tags](#). Tags can't modify value of a variable whereas filters can be used for incrementing value of a variable or modifying it to one's own need.

Syntax

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

Filters can be “chained.” The output of one filter is applied to the next. `{{ text|escape|linebreaks }}` is a common idiom for escaping text contents, then converting line breaks to `<p>` tags.

Example

```
{{ value | length }}
```

If value is `['a', 'b', 'c', 'd']`, the output will be `4`.

Major Template Filters in Django

This article revolves around various Django Template Filters one can use during a project. Filters transform the values of variables and tag arguments. Let's check some major filters.

add	addslashes	capfirst
center	cut	date
default	dictsort	divisibleby

escape	filesizeformat	first
join	last	length
linenumbers	lower	make_list
random	slice	slugify
time	timesince	title
unordered_list	upper	wordcount

add

1. It is used to add an argument to the value. **Example**

```
{{ value | add:"2" }}
```

1. If value is 4, then the output will be 6. This filter is used to **increment a variable in django Templates.**

addslashes

1. It is used to add slashes before quotes. Useful for escaping strings in CSV.

Example

```
{{ value | addslashes }}
```

1. If value is “I’m Jai”, the output will be “I\’m Jai”.

capfirst

1. It is used to capitalize the first character of the value. If the first character is not a letter, this filter has no effect. **Example**

```
{{ value | capfirst }}
```

1. If value is “jai”, the output will be “Jai”.

center

1. It is used to center the value in a field of a given width. **Example**

```
"{{ value | center:"15" }}"
```

1. If value is "Jai", the output will be " Jai ".

cut

1. It is used to remove all values of arg from the given string. **Example**

```
{{ value | cut:" " }}
```

1. If value is "String with spaces", the output will be "Stringwithspaces".

date

1. It is used to format a date according to the given format. **Example**

```
{{ value | date:"D d M Y" }}
```

1. If value is a datetime object (e.g., the result of `datetime.datetime.now()`), the output will be the string 'Thu 06 Feb 2020'. For More information and patterns [visit here](#).

default

1. It is used to give a default value to a variable. If variable evaluates to False, it will use the default argument given else the variable value itself. **Example**

```
{{ value | default:"nothing" }}
```

1. If value is "" (the empty string), the output will be nothing.

dictsort

1. It takes a list of dictionaries and returns that list sorted by the key given in the argument. **Example**

```
{{ value | dictsort:"name" }}
```

1. If value is:

```
[
    {'name': 'zed', 'age': 19},
    {'name': 'amy', 'age': 22},
    {'name': 'joe', 'age': 31},
]
```

1. then the output would be:

```
[
    {'name': 'amy', 'age': 22},
    {'name': 'joe', 'age': 31},
    {'name': 'zed', 'age': 19},
]
```

divisibleby

1. It returns True if the value is divisible by the argument. **Example**

```
{{ value | divisibleby:"3" }}
```

1. If value is 21, the output would be True.

escape

1. It is used to escape a string's HTML. Specifically, it makes these replacements:

html

```
< is converted to &lt;
> is converted to &gt;
' (single quote) is converted to &#x27;
" (double quote) is converted to &quot;
& is converted to &amp;
```

1. **Example**

```
{{ title | escape }}
```

filesizeformat

1. It is used to format the value like a 'human-readable' file size (i.e. '13 KB', '4.1 MB', '102 bytes', etc.). **Example**

```
{{ value | filesizeformat }}
```

1. If value is 123456789, the output would be 117.7 MB.

first

1. It is used to return the first item in a list. **Example**

```
{{ value | first }}
```

1. If value is the list ['a', 'b', 'c'], the output will be 'a'.

join

1. It is used to join a list with a string, like Python's str.join(list) **Example**

```
{{ value | join:" // " }}
```

1. If value is the list ['a', 'b', 'c'], the output will be the string "a // b // c".

last

1. It is used to return the last item in a list. **Example**

```
{{ value | last }}
```

1. If value is the list ['a', 'b', 'c', 'd'], the output will be the string "d".

length

1. It is used to return the length of the value. This works for both strings and lists. **Example**

```
{{ value | length }}
```

1. If value is ['a', 'b', 'c', 'd'] or "abcd", the output will be 4.

linenumbers

1. It is used to display text with line numbers. **Example**

```
{{ value | linenumbers }}
```

1. If value is:

```
one  
two  
three
```

1. the output will be:

```
1. one  
2. two  
3. three
```

lower

1. It is used to convert a string into all lowercase. **Example**

```
{{ value | lower }}
```

1. If value is **My Name is Jai**, the output will be **my name is jai**.

make_list

1. It is used to return the value turned into a list. For a string, it's a list of characters. For an integer, the argument is cast to a string before creating a list. **Example**

```
{{ value | make_list }}
```

1. If value is the string "Naveen", the output would be the list ['N', 'a', 'v', 'e', 'e', 'n']. If value is 123, the output will be the list ['1', '2', '3'].

random

1. It is used to return a random item from the given list. **Example**

```
{{ value | random }}
```

1. If value is the list ['a', 'b', 'c', 'd'], the output could be "b".

slice

1. It is used to return a slice of the list. **Example**

```
{{ some_list | slice":2" }}
```

1. If some_list is ['a', 'b', 'c'], the output will be ['a', 'b'].

slugify

1. It is used to convert to ASCII. It converts spaces to hyphens and removes characters that aren't alphanumerics, underscores, or hyphens. Converts to lowercase. Also strips leading and trailing whitespace. **Example**

```
{{ value | slugify }}
```

1. If value is "Jai is a slug", the output will be "jai-is-a-slug".

time

1. It is used to format a time according to the given format. **Example**

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

1. If value is equivalent to **datetime.datetime.now()**, the output will be the string "01:23".

timesince

1. It is used to format a date as the time since that date (e.g., "4 days, 6 hours").
Example

```
{{ blog_date | timesince:comment_date }}
```

1. if blog_date is a date instance representing midnight on 1 June 2006, and comment_date is a date instance for 08:00 on 1 June 2006, then the following would return "8 hours"

title

1. It is used to convert a string into titlecase by making words start with an uppercase character and the remaining characters lowercase. This filter makes no effort to keep “trivial words” in lowercase. **Example**

```
{{ value | title }}
```

1. If value is “my FIRST post”, the output will be “My First Post”.

unordered_list

1. It is used to recursively take a self-nested list and returns an HTML unordered list – WITHOUT opening and closing tags. **Example**

```
{{ var | unordered_list }}
```

1. if var contains ['States', ['Kansas', ['Lawrence', 'Topeka'], 'Illinois']], then {{ var|unordered_list }} would return:

html

```
<li>States
<ul>
  <li>Kansas
  <ul>
    <li>Lawrence</li>
    <li>Topeka</li>
  </ul>
</li>
<li>Illinois</li>
</ul>
</li>
</li>
```

upper

1. It is used to convert a string into all uppercase. **Example**

```
{{ value | upper }}
```

1. If value is “Jai is a slug”, the output will be “JAI IS A SLUG”.

wordcount

1. It is used to return the number of words. **Example**

```
{{ value | wordcount }}
```

1. If value is “jai is a slug”, the output will be 4.

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...



14

Previous Article

cycle - Django Template Tags

Next Article

for ... empty loop - Django Template Tags

Similar Reads

Custom Template Filters in Django

Django is a Python-based web framework that allows you to quickly create efficient web applications. It is also called batteries included framework because...

2 min read

How to Add Multiple Arguments to Custom Template Filter in a Django...

In Django, template filters are useful for modifying data before displaying it in templates. While built-in filters cover common cases, custom logic is sometimes...

4 min read

Customizing Filters in Django REST Framework

Prerequisite: Adding Filtering in APIs – Django REST Framework [link needed article on published yet] Django filters facilitate filtering the queryset to retrieve...

4 min read

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

Render a HTML Template as Response - Django Views

A view function, or view for short, is simply a Python function that takes a Web request and returns a Web response. This article revolves around how to render...

3 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

Article Tags : [Python](#) [Django-templates](#) [Python 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](#)

Languages

- [Python](#)
- [Java](#)
- [C++](#)
- [PHP](#)
- [GoLang](#)
- [SQL](#)
- [R Language](#)
- [Android Tutorial](#)
- [Tutorials Archive](#)

Data Science & ML

- [Data Science With Python](#)

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

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

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

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