**Template Filters**

Django Template Engine provides filters which are used to transform the values of variables and tag arguments. 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.

{{ variable | filter-name }}

{{ variable | filter-name | filter-name }} 🡪 filter chaining

|  |  |  |
| --- | --- | --- |
| [add](https://www.geeksforgeeks.org/django-template-filters/#add) | [addslashes](https://www.geeksforgeeks.org/django-template-filters/#addslashes) | [capfirst](https://www.geeksforgeeks.org/django-template-filters/#capfirst) |
| [center](https://www.geeksforgeeks.org/django-template-filters/#center) | [cut](https://www.geeksforgeeks.org/django-template-filters/#cut) | [date](https://www.geeksforgeeks.org/django-template-filters/#date) |
| [default](https://www.geeksforgeeks.org/django-template-filters/#default) | [dictsort](https://www.geeksforgeeks.org/django-template-filters/#dictsort) | [divisibleby](https://www.geeksforgeeks.org/django-template-filters/#divisibleby) |
| [escape](https://www.geeksforgeeks.org/django-template-filters/#escape) | [filesizeformat](https://www.geeksforgeeks.org/django-template-filters/#filesizeformat) | [first](https://www.geeksforgeeks.org/django-template-filters/#first) |
| [join](https://www.geeksforgeeks.org/django-template-filters/#join) | [last](https://www.geeksforgeeks.org/django-template-filters/#last) | [length](https://www.geeksforgeeks.org/django-template-filters/#length) |
| [linenumbers](https://www.geeksforgeeks.org/django-template-filters/#linenumbers) | [lower](https://www.geeksforgeeks.org/django-template-filters/#lower) | [make\_list](https://www.geeksforgeeks.org/django-template-filters/#make_list) |
| [random](https://www.geeksforgeeks.org/django-template-filters/#random) | [slice](https://www.geeksforgeeks.org/django-template-filters/#slice) | [slugify](https://www.geeksforgeeks.org/django-template-filters/#slugify) |
| [time](https://www.geeksforgeeks.org/django-template-filters/#time) | [timesince](https://www.geeksforgeeks.org/django-template-filters/#timesince) | [title](https://www.geeksforgeeks.org/django-template-filters/#title) |
| [unordered\_list](https://www.geeksforgeeks.org/django-template-filters/#unordered_list) | [upper](https://www.geeksforgeeks.org/django-template-filters/#upper) | [wordcount](https://www.geeksforgeeks.org/django-template-filters/#wordcount) |

**add**

Adds the argument to the value.

{{ value|add:"2" }}

**addslashes**

Adds slashes before quotes. Useful for escaping strings in CSV, for example.

For example:

{{ value|addslashes }}

If **value** is **"I'm using Django"**, the output will be **"I\'m using Django"**.

**capfirst**

Capitalizes the first character of the value. If the first character is not a letter, this filter has no effect.

**center**

Centers the value in a field of a given width.

For example:

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

If **value** is **"Django"**, the output will be **"     Django    "**.

**cut**

Removes all values of arg from the given string.

For example:

{{ value|cut:" " }}

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

**date**

Formats a date according to the given format.

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

**default[¶](https://docs.djangoproject.com/en/4.1/ref/templates/builtins/" \l "default" \o "Permalink to this headline)**

If value evaluates to **False**, uses the given default. Otherwise, uses the value.

For example:

{{ value|default:"nothing" }}

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

**default\_if\_none**

If (and only if) value is **None**, uses the given default. Otherwise, uses the value.

Note that if an empty string is given, the default value will not be used. Use the [**default**](https://docs.djangoproject.com/en/4.1/ref/templates/builtins/#std-templatefilter-default) filter if you want to fallback for empty strings.

For example:

{{ value|default\_if\_none:"nothing" }}

If **value** is **None**, the output will be **nothing**.

**dictsort**

Takes a list of dictionaries and returns that list sorted by the key given in the argument.

For example:

{{ value|dictsort:"name" }}

**dictsortreversed**

Takes a list of dictionaries and returns that list sorted in reverse order by the key given in the argument. This works exactly the same as the above filter, but the returned value will be in reverse order.

Views.py

from django.shortcuts import render

# Create your views here.

import datetime

def filtertest(request):

    d=datetime.datetime.today()

    dict1={"num1":10,

    "list1":[10,20,30,40,50],

    "list2":[1,2,3,4,5],

    "names":['naresh','suresh','ramesh'],

    "s1":"python programming language",

    "d":d,

    "s2":"",

    "x":None,

    "person":[

    {'name': 'zed', 'age': 19},

    {'name': 'amy', 'age': 22},

    {'name': 'joe', 'age': 31},

]}

    return render(request,"filter.html",context=dict1)

filter.html

html>

    <h2>

        {{num1}} <br>

        {{list1}} <br>

        {{list2}} <br>

        {{num1|add:"5"}} <br>

        {{list1|add:list2}}<br>

        {%for value in list1%}

        {{value|add:"1"}} <br>

        {%endfor%}

        {%for name in names%}

        {{name|capfirst}}<br>

        {%endfor%}

        {%for name in names%}

        {{name|center:"15"}}<br>

        {%endfor%}

        {{s1}}<br>

        {{s1|cut:" "}}<br>

        {{d|date:"d/m/Y"}}<br>

        {{d|date:"l d b Y"}}<br>

        {{d|date:"h:i:s"}}<br>

        {{s2|default:"None"}}<br>

        {{x|default\_if\_none:"empty"}}<br>

        {{person}}<br>

        {{person|dictsort:"name"}}<br>

        {{person|dictsort:"age"}}

    </h2>

</html>

Urls.py

from django.contrib import admin

from django.urls import path

from app import views

urlpatterns = [

    path('admin/', admin.site.urls),

    path("test/",views.filtertest)

]