

## **:-DTL Tags:-**

### **# Without closing tag**

1. {% csrf\_token %}
2. {% extends %}
3. {% include %}
4. {% load %}
5. {% url %}
6. {% cycle %}

### **# With closing tag:**

7. {% with %}  
{% endwith %}
8. {% if (condition) %}  
{% elif (condition) %}  
{% else %}  
{% endif %}
9. {% for x in condition %}  
{% endfor %}
10. {% block content %}  
{% endblock %}
11. {% comment %}  
{% endcomment %}

## 12. for ... empty

The for tag can take an optional `{% empty %}` clause whose text is displayed if the given array is empty or could not be found:

```
<ul>
  {% for athlete in athlete_list %}
    <li>{{ athlete.name }}</li>
  {% empty %}
    <li>Sorry, no athletes in this list.</li>
  {% endfor %}
</ul>
```

The above is equivalent to – but shorter, cleaner, and possibly faster than – the following:

```
<ul>
  {% if athlete_list %}
    {% for athlete in athlete_list %}
      <li>{{ athlete.name }}</li>
    {% endfor %}
  {% else %}
    <li>Sorry, no athletes in this list.</li>
  {% endif %}
</ul>
```

## DTL filter

**{{ key }}**

### **Add:---**

Adds the argument to the value.

For example:

**{{ key |add:"2" }}**

If **value** is **4**, then the output will be **6**.

This filter will first try to coerce both values to integers. If this fails, it'll attempt to add the values together anyway. This will work on some data types (strings, list, etc.) and fail on others. If it fails, the result will be an empty string.

For example, if we have:

**{{ first|add:second }}**

and **first** is **[1, 2, 3]** and **second** is **[4, 5, 6]**, then the output will be **[1, 2, 3, 4, 5, 6]**.

## title

Converts a string into titlecase by making words start with an uppercase character and the remaining characters lowercase. This tag makes no effort to keep "trivial words" in lowercase.

For example:

**{{ value|title }}**

If **value** is **"my FIRST post"**, the output will be **"My First Post"**.

## truncatechars

Truncates a string if it is longer than the specified number of characters. Truncated strings will end with a translatable ellipsis character ("...").

**Argument:** Number of characters to truncate to

For example:

**{{ value|truncatechars:7 }}**

If **value** is "**Joel is a slug**", the output will be "**Joel i...**".

### **truncatewords**

Truncates a string after a certain number of words.

**Argument:** Number of words to truncate after

For example:

```
{{ value|truncatewords:2 }}
```

If **value** is "**Joel is a slug**", the output will be "**Joel is ...**".

Newlines within the string will be removed.

### **wordcount**

Returns the number of words.

For example:

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

If **value** is "**Joel is a slug**", the output will be **4**.

### **lower**

Converts a string into all lowercase.

For example:

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

If **value** is **Totally LOVING this Album!**, the output will be **totally loving this album!**.

### **upper**

Converts a string into all uppercase.

For example:

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

If **value** is "**Joel is a slug**", the output will be "**JOEL IS A SLUG**".

## **Capfirst:----**

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

For example:

```
{{ key |capfirst }}
```

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

## **Center**

Centers the value in a field of a given width.

For example:

```
<pre> {{ key |center:"15" }} </pre>
```

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

## **ljust**

Left-aligns the value in a field of a given width.

**Argument:** field size

For example:

```
<pre> {{ key |ljust:"10" }} </pre>
```

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

## **rjust**

Right-aligns the value in a field of a given width.

**Argument:** field size

For example:

```
<pre> {{ value|rjust:"10" }} </pre>
```

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"**.

## join

Joins a list with a string, like Python's **str.join(list)**

For example:

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

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

## default

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** 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**.

## divisibleby

Returns **True** if the value is divisible by the argument.

For example:

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

If value is 21, the output would be True.

## first

Returns the first item in a list.

For example:

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

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

## **last**

Returns the last item in a list.

For example:

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

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

## **length**

Returns the length of the value. This works for both strings and lists.

For example:

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

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

The filter returns 0 for an undefined variable.

## **random**

Returns a random item from the given list.

For example:

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

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

## **slice**

Returns a slice of the list.

Uses the same syntax as Python's list slicing. See the [Python documentation](#) for an introduction.

Example:

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

If **some\_list** is ['a', 'b', 'c'], the output will be ['a', 'b'].

## **slugify**

Converts to ASCII. Converts spaces to hyphens. Removes characters that aren't alphanumerics, underscores, or hyphens. Converts to lowercase. Also strips leading and trailing whitespace.

For example:

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

If **value** is **"Joel is a slug"**, the output will be **"joel-is-a-slug"**.

The for loop sets a number of variables available within the loop:

<b>Variable</b>	<b>Description</b>
<b>forloop.counter</b>	The current iteration of the loop (1-indexed)
<b>forloop.counter0</b>	The current iteration of the loop (0-indexed)
<b>forloop.revcounter</b>	The number of iterations from the end of the loop (1-indexed)
<b>forloop.revcounter0</b>	The number of iterations from the end of the loop (0-indexed)
<b>forloop.first</b>	True if this is the first time through the loop
<b>forloop.last</b>	True if this is the last time through the loop
<b>forloop.parentloop</b>	For nested loops, this is the loop surrounding the current one