**Template Language**

A **template** is a text document, or a normal Python string, that is marked-up using the Django template language. A template can contain **block tags** or **variables**.

**Block tags** are surrounded by **"{%"** and **"%}"**.

Example template with block tags:

{% **if** is\_logged\_in %}Thanks for logging in!{% **else** %}Please log in.{% **endif** %}

**Variables**

Variables look like this: {{ variable }}.

When the template engine encounters a variable, it evaluates that variable and replaces it with the result.

Variable names consist of any combination of alphanumeric characters and the underscore ("\_") but may not start with an underscore.

**Templates**

A template is simply a text file. It can generate any text-based format (HTML, XML, CSV, etc.).

A template contains **variables**, which get replaced with values when the template is evaluated, and **tags**, which control the logic of the template.

**Comment**

Ignores everything between {% comment %} and

{% endcomment %}.

**csrf\_token**

This tag is used for CSRF protection, csrf stands for **Cross Site Request Forgeries.**

**Boolean operators in if tag's**

If tags may use **and, or** or **not**to test a number of variables or to negate a given variable:

**Example 1**

{% if athlete\_list and coach\_list %}

Both athletes and coaches are available.

{% endif %}

**Example 2**

{% if not athlete\_list %}

There are no athletes.

{% endif %}

**Example 3**

{% if athlete\_list or coach\_list %}

There are some athletes or some coaches.

{% endif %}

**Example 4**

{% if not athlete\_list or coach\_list %}

There are no athletes or there are some coaches.

{% endif %}

**Example 5**

{% if athlete\_list and not coach\_list %}

There are some athletes and absolutely no coaches.

{% endif %}

Use of both **and** and **or**clauses within the same tag is allowed, with and having higher precedence than or .

**Example 6**

{% if athlete\_list and coach\_list or cheerleader\_list %}

will be interpreted like:

if (athlete\_list and coach\_list) or cheerleader\_list

**operators in if tag's**  
If tags may also use the operators like ==, !=, <, >, <=, >=, in, not in, is, and is not .

**== operator Example:**

{% if somevar == "x" %}

This appears if variable somevar equals the string "x"

{% endif %}

**!= operator Example:**

{% if somevar != "x" %}

This appears if variable somevar does not equal the string "x",

or if somevar is not found in the context

{% endif %}

**< operator Less than. Example:**

{% if somevar < 100 %}

This appears if variable somevar is less than 100.

{% endif %}

**> operator Greater than. Example:**

{% if somevar > 0 %}

This appears if variable somevar is greater than 0.

{% endif %}

**<= operator Less than or equal to. Example:**

{% if somevar <= 100 %}

This appears if variable somevar is less than 100 or equal to 100.

{% endif %}

**>= operator Greater than or equal to. Example:**

{% if somevar >= 1 %}

This appears if variable somevar is greater than 1 or equal to 1.

{% endif %}

**in operator**

Contained within. This operator is supported by many Python containers to test whether the given value is in the container.

**Example :** {% if "bc" in "abcdef" %}

This appears since "bc" is a substring of "abcdef"

{% endif %}

**Example :**

{% if "hello" in greetings %}

If greetings is a list or set, one element of which is the string

"hello", this will appear.

{% endif %}

**Example :**

{% if user in users %}

If users is a QuerySet, this will appear if user is an

instance that belongs to the QuerySet.

{% endif %}

**not in operator**

Not contained within. This is the negation of the in operator.

**is operator**

Object identity. Tests if two values are the same object.

**Example:**

{% if somevar is True %}

This appears if and only if somevar is True.

{% endif %}

**Example :**

{% if somevar is None %}

This appears if somevar is None, or if somevar is not found in the context. {% endif %}

**is not operator**

Negated object identity. Tests if two values are not the same object. This is the negation of the is operator.

**Example:**

{% if somevar is not True %}

This appears if somevar is not True, or if somevar is not found in the

context.

{% endif %}

**Example :**

{% if somevar is not None %}

This appears if and only if somevar is not None.

{% endif %}

**load**

Loads a custom template tag set.

{% load somelibrary package.otherlibrary %}

You can also selectively load individual filters or tags from a library, using the from argument.

{% load foo bar from somelibrary %}

**include**

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.

**This example** includes the contents of the template "foo/bar.html":

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

This example includes the contents of the template whose name is contained in the variable template\_name:

**{% include template\_name %}**

**now**

Displays the current date and/or time, using a format according to the given string

**Example:**

{% now "jS F Y H:i" %}

{% **now "SHORT\_DATETIME\_FORMAT"** %}<**br**>  
{% **now "SHORT\_DATE\_FORMAT"** %}<**br**>  
{% **now "DATETIME\_FORMAT"** %}<**br**>  
{% **now "DATE\_FORMAT"** %}

**Filters**

You can modify variables for display by using **filters**.

Filters look like this: **{{ name|lower }}**.

This displays the value of the **{{ name }}** variable after being filtered through the **lower** filter, which converts text to lowercase. Use a pipe (**|**) to apply a filter.

Some filters take arguments. A filter argument looks like this: **{{bio|truncatewords:30 }}**. This will display the first 30 words of the **bio** variable.

Filter arguments that contain spaces must be quoted; for example, to join a list with commas and spaced you’d use **{{ list|join:", " }}**.

**Built-in filter reference**

**1) add :** Adds the argument to the value.

**For example:** {{ value|add:"2" }}

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

**2) addslashes :** Adds slashes before quotes. Useful for escaping strings in CSV, for example.

**For example:** {{ value|addslashes }}

If **value** is **"This is Naveen's",** the output will be **"This is Naveen\'s**.

**3) capfirst :** Capitalizes the first character of the value. If the first character is not a letter, this filter has no effect.

**For example:** {{ value|capfirst }}

If **value** is **"naveen"**, the output will be **"Naveen"**.

**4) center :** Center the value in a field of a given width.

**For example:** "{{ value|center:"15" }}"

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

**5) 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"**.

**6) 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**.

**7) 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 the string **"nothing"**.

**8) dictsort:** Takes a list of dictionaries and returns that list sorted by the key given in the argument.

**For example:** {{ value|dictsort:"name" }}

If **value** is : [

{'name': 'zed', 'age': 19},

{'name': 'amy', 'age': 22},

{'name': 'joe', 'age': 31},

]

then the output would be:

[

{'name': 'amy', 'age': 22},

{'name': 'joe', 'age': 31},

{'name': 'zed', 'age': 19},]

**9) 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.

**10) 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**.

**11) filesizeformat :** Formats the value like a ‘human-readable’ file size (i.e. **'13 KB'**, **'4.1MB'**, **'102 bytes'**, etc).

**For example:** {{ value|filesizeformat }}

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

**12) first :** Returns the first item in a list.

**For example:** {{ value|first }}

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

**13) floatformat :** When used without an argument, rounds a floating-point number to one decimal place – but only if there’s a decimal part to be displayed. For example:

| **value** | **Template** | **Output** |
| --- | --- | --- |
| **34.23234** | **{{ value|floatformat }}** | **34.2** |
| **34.00000** | **{{ value|floatformat }}** | **34** |
| **34.26000** | **{{ value|floatformat }}** | **34.3** |

If used with a numeric integer argument, **floatformat** rounds a number to that many decimal places.

**For example:**

| **value** | **Template** | **Output** |
| --- | --- | --- |
| **34.23234** | **{{ value|floatformat:3}}** | **34.232** |
| **34.00000** | **{{ value|floatformat:3}}** | **34.000** |
| **34.26000** | **{{ value|floatformat:3}}** | **34.260** |

Particularly useful is passing 0 (zero) as the argument which will round the float to the nearest integer.

| **value** | **Template** | **Output** |
| --- | --- | --- |
| **34.23234** | **{{ value|floatformat:"0" }}** | **34** |
| **34.00000** | **{{ value|floatformat:"0" }}** | **34** |
| **39.56000** | **{{ value|floatformat:"0" }}** | **40** |

If the argument passed to **floatformat** is negative, it will round a number to that many decimal places – but only if there’s a decimal part to be displayed. For example:

| **value** | **Template** | **Output** |
| --- | --- | --- |
| **34.23234** | **{{ value|floatformat:"-3" }}** | **34.232** |
| **34.00000** | **{{ value|floatformat:"-3" }}** | **34** |
| **34.26000** | **{{ value|floatformat:"-3" }}** | **34.260** |

Using **floatformat** with no argument is equivalent to using **floatformat** with an argument of **-1**.

**14) 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"**.

**15) get\_digit :** Given a whole number, returns the requested digit, where 1 is the right-most digit, 2 is the second-right-most digit, etc. Returns the original value for invalid input (if input or argument is not an integer, or if argument is less than 1). Otherwise, output is always an integer.

**For example:** {{ value|get\_digit:"2" }}

If **value** is **123456789**, the output will be **8**.

**16) 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"**.

**17) 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']**, the output will be **4**.

**18) length\_is :** Returns **True** if the value’s length is the argument, or **False** otherwise.

**For example:** {{ value|length\_is:"4" }}

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

**19) linenumbers :** Displays text with line numbers.

**For example:** {{ value|linenumbers }}

If **value** is :

one

two

three

**the output will be:**

1. one

2. two

3. three

**20) lower:** Converts a string into all lowercase.

**For example:** {{ value|lower }}

If **value** is **This is Naveen**, the output will be **this is naveen**.

**21) make\_list :** Returns the value turned into a list. For a string, it’s a list of characters. For an integer, the argument is cast into an unicode string before creating a list.

**For example:** {{ value|make\_list }}

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']**.

**22) 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"**.

**23) slice :** Returns a slice of the list.

**Example:** {{ some\_list|slice:":2" }}

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

**25) time :** Formats a time according to the given format.

**For example:** {{ value|time:"H:i" }}

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

**26) title :** Converts a string into title case 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"**.

**28) 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 **"This is Naveen From "**, the output will be **"This is..."**.

Newlines within the string will be removed.

**29) upper :** Converts a string into all uppercase.

**For example:** {{ value|upper }}

If **value** is **"This is Naveen"**, the output will be **"THIS IS NAVEEN"**.

**30) wordcount :** Returns the number of words.

**For example:** {{ value|wordcount }}

If **value** is **"This is Naveen Kumar"**, the output will be **4**.

**31) wordwrap:** Wraps words at specified line length.

**Argument:** number of characters at which to wrap the text

**For example:** {{ value|wordwrap:5 }}