

Operators

| | |
|----------|--|
| [not] in | Containment operator |
| is [not] | Test operator |
| .. | Creates a sequence |
| | Applies a filter |
| ~ | Concatenation |
| . or [] | Gets an attribute of an object |
| ?: | Ternary operator |
| {{}} | Used to print the result of an expression evaluation |
| {% %} | Used to execute statements |

Variables

Print

```
{{ a_variable }}
{{ foo.bar }}
{{ foo['bar'] }}
```

Set

```
{% set value = 'foobar' %}
{% set foo = 'foo' %}
{% set foo = [ 1, 2 ] %}
{% set foo = {'foo': 'bar'} %}
{% set foo = 'foo' ~ 'bar' %}
{% set foo, bar = 'foo', 'bar' %}
```

```
{% set value %}
<div id="foobar">
    ...
</div>
{% endset %}
```

Filters

```
{{ foo|striptags }}
{% filter upper %}This text becomes uppercase{% endfilter %}
```

Comment

```
{# single line comment #}
```

Whitespace Control

```
{% spaceless %}
<div>
  <strong>title</strong>
</div>
{% endspaceless %}
{# Output will be <div><strong>title</strong></div> #}
```

```
{% set value = 'Lorem Ipsum' %}
{{- value -}}
```

```
<h1> {{- value }} </h1>
{# Will output <h1>Lorem Ipsum </h1> #}
```

Macro

```
{% macro link(url,label) %}
  <a href="{{ url }}"> {{ label }}</a>
{% endmacro %}
```

```
{{ _self.link('index.html','Home') }}
```

```
{% import "forms.html" as forms %}
{% from 'forms.html' import input as input_field %}

{{ form.input('foobar') }}
{{ input_field('foobar') }}
```

Escaping

```
{{ '({{ }})' }}
```

```
{% raw %}
  {% set value = 'Lorem Ipsum' %}
  <h1>{{ value }}</h1>
{% endraw %}
```

Control Structures

FOR

```
<ul>
  {% for item in collection %} <li>{{ item }}</li> {% endfor %}
</ul>
```

```
{% for letter in 'a'..'z' %}
  {{ letter }}
{% endfor %}
```

```
{% for i in 0..9 %}
  {{ i }}
{% endfor %}
```

```
{% for item in collection %}
  {{ item }}
{% else %}
  No item.
{% endfor %}
```

```
{% for i in range(0,10,2) %}
  {{ i }}
{% endfor %}
```

```
{% for item in collection %}
  {{ loop.index }}
{% endfor %}
```

```
{% for key in items|keys %}
  {{ key }}
{% endfor %}
```

```
{% for key, item in items %}
  {{ key }} : {{ item }}
{% endfor %}
```

| | |
|-----------------------|---|
| loop.index | The current iteration of the loop (1 indexed) |
| loop.index0 | The current iteration of the loop (0 indexed) |
| loop.revindex | The number of iterations from the end of the loop (1 indexed) |
| loop.revindex0 | The number of iterations from the end of the loop (0 indexed) |
| loop.first | True if first iteration |
| loop.last | True if last iteration |
| loop.length | The number of items in the sequence |
| loop.parent | The parent context |

IF

```
{% if foo %}
  ...
{% elseif bar %}
  ...
{% else %}
  ...
{% endif %}
```

```
{% if items %}
  <ul>
    {% for item in items %}
      <li>{{ item }}</li>
    {% endfor %}
  </ul>
{% endif %}
```

Templates and Blocks

Define a block (redefine is existing)

```
{% block sidebar %} ... {% endblock %}
```

Parent block

```
{% block sidebar %} {{ parent() }} {% endblock %}
```

Extends a template

```
{% extends "base.html.twig" %}
```

Include a template

```
{% include "header.html.twig" %}
```

Include with access options

```
{% include "header.html.twig" with { 'foo' : 'bar' } only %}
```

Import blocks from a template

```
{% use "blocks.html.twig" %}
```

Import blocks with alias

```
{% use "blocks.html.twig" with sidebar as base_sidebar %}
```

Call a block

```
{{ block('base_sidebar') }}
```

Built-in Filters

| | |
|--------------------|--|
| date | <pre>{{ post.created_at date("d/m/Y") }} {{ "now" date("d/m/Y") }}</pre> |
| format | <pre>{{ "I like %s and %s" format(foo, 'bar') }}</pre> |
| replace | <pre>{{ "Hello %name!" replace({'%name%': 'world'}) }}</pre> |
| url_encode | URL encodes a given string. |
| json_encode | Return the JSON representation of a given string. |
| title | Return a titlecased version of the value. |
| capitalize | Return a capitalized version of the value. |
| upper | Convert a value to uppercase |
| lower | Convert a value to lowercase |
| striptags | Strips SGML/XML tags and replace adjacent whitespaces by one space. |
| join | <pre>{{ [a, b, c] join(', ') }} {# return "a,b,c" #} {{ [a, b, c] join }} {# return abc #}</pre> |
| reverse | Reverses an array or an object implementing the iterator interface |
| length | Return the number of items of a sequence or array, or the length of a string |
| sort | Sort an array |
| default | <pre>{{ var default('var is not defined') }}</pre> |
| keys | <pre>{% for key in array keys %} ... {% endfor %}</pre> |
| escape, e | Convert a string to HTML-safe sequences. |
| raw | Value will not be escaped |
| merge | Merge an array with the value. |

Built-in Tests

| | |
|--------------------|---|
| divisibleby | <pre>{% if myNumber is divisibleby(2) %} ... {% endif %}</pre> |
| none | <pre>{% if var is none %} ... {% endif %}</pre> |
| even | <pre>{% if var is even %} ... {% endif %}</pre> |
| odd | <pre>{% if var is odd %} ... {% endif %}</pre> |
| sameas | Check if a variable points to the same memory address than another. <pre>{% if var1 is sameas(var2) %} ... {% endif %}</pre> |
| constant | <pre>{% if foo.var is constant('FOO:BAR') %} ... {% endif %}</pre> |
| defined | <pre>{% if var is defined %} ... {% endif %}</pre> |
| empty | <pre>{% if var is empty %} ... {% endif %}</pre> |

Global Functions

| | |
|-----------------|---|
| range | <pre>{% for i in range(0, 3) %} {{ i }} {% endfor %} {# returns 0, 1, 2, 3 #} {% for i in range(0, 6, 2) %} {{ i }} {% endfor %}</pre> |
| cycle | <pre>{% for i in 0..10 %} {{ cycle(['odd', 'even'], i) }} {% endfor %}</pre> |
| constant | Returns the constant value for a given string. <pre>{{ some_date date(constant('DATE_W3C')) }}</pre> |

Automatic Escaping

| |
|--|
| <pre>{% autoescape %} Everything will be automatically escaped in this block using the HTML strategy {% endautoescape %}</pre> |
| <pre>{% autoescape 'html' %} Everything will be automatically escaped in this block using the HTML strategy {% endautoescape %}</pre> |
| <pre>{% autoescape 'js' %} Everything will be automatically escaped in this block using the js escaping strategy {% endautoescape %}</pre> |
| <pre>{% autoescape false %} Everything will be outputted as is in this block {% endautoescape %}</pre> |

TWIG Extension

```
Class
class Project_Twig_Extension extends Twig_Extension
{
    // returns the unique name of the extension
    public function getName()
    {
        return 'project';
    }

    // returns an array of the implemented filters
    public function getFilters()
    {
        return array(
            'myFilter' => new Twig_Filter_Method($this, 'myFilter'),
        );
    }

    // exemple of a self-made filter
    public function myFilter($string)
    {
        return '...' . $string . '...';
    }

    // returns an array of global variables and functions
    public function getGlobals()
    {
        return array(
            'text' => new Text(),
            'lipsum' => new Twig_Function(new Text(), 'getLipsum'),
        );
    }

    // returns an array of implemented tests
    public function getTests()
    {
        return array(
            'even' => new Twig_Test_Function('twig_test_even'),
        );
    }

    // returns an array of tags
    public function getTokenParsers()
    {
        return array(new Project_Set_TokenParser());
    }

    // ...
}
```

Registering extension

```
$twig = new Twig_Environment($loader);
$twig->addExtension(new Project_Twig_Extension());
```

TWIG Extensions

```
Text :
  truncate, wordwrap and nl2br filters
I18N:
  trans, transchoice filter
Debug:
  debug tag

# Load extension
$this->addExtension(new Twig_Extension_Text());

# add <br /> instead line break
{{ content|nl2br }}

# truncate at 20 words
{{ content|wordwrap(20) }}

# truncat at 20 characters
{{ content|truncate(20) }}

# translate
{{ content|trans }}

# output when the debug environment option is set to true
{{ debug items }}
{{ debug foo.bar }}
```

Symfony2

Generate URL

```
{{ path('route_name', {'param': 'value'}) }}
```

Generate absolute URL

```
{{ url('route_name') }}
```

Link to Assets

```
<link rel="stylesheet" href="{{ asset('css/style.css') }}" type="text/css" />

```

Registering extension

```
<services>
  <service id="acme.twig.acme_extension" class="Project_Twig_Extension">
    <tag name="twig.extension" />
  </service>
</services>
```