Délimiteurs

{{ Variable }}	
{{ Var.attrib }}	{{ Var['attrib'] }}
{{ Variable | filtre | ... }}	
{% Controle %}	
{# Commentaires #}	

Controles

for else endfor	loop.xxx
if elif else endif	
macro endmacro	call
filter endfilter	
set	set block endset
include	ignore missing
import	from import as

Opérateurs

+	-	/	//
%	*	**	
==	!=	>	>=
<	<=		
and	or	not	(expr)
in	is	|	~
./[]	foo.bar	foor['bar']	

Espaces

{%- -%}	supprime espaces avant ou après
{%+	

Filtre Chaines

capitalize(s)	center(value,width)
escape(s)	forceescape(value)
format(value,*args,**kwargs))	
indent(s,width=4,indentfirst=False)	
lower(s)	upper(s)
pprint(value,verbose=False)	
replace(s,old,new,count=None)	
string(object)	striptags(value)
title(s)	trim(value)
urlencode(value)	
urlize(value,trim_url_limit,nofollow,target)	
wordcount(s)	xmlattr(d,autospace=True)

Filtre Nombres

abs(number)	random(seq)
int(value,default=0)	float(value,default=0.0)
sum(iterable,attribute=None,start=0)	

Héritage

block name	endblock
extends "templfile"	
self.name	super()
Les endblock peuvent être nommés	

Global

range(start,stop,step)	
libsum(n=5,html=True,min=20,max=100)	
dict(**items)	class cylcler(*items
reset()	next()
current	class joiner(sep=',')

Filtre Divers

attr(obj,name)	rejectattr()
default(value,default,boolean=False) |d	
filesizeformat(value,binary=False)	
safe(value)	
select()	selectattr()

Utilisé dans {{ Variable \| filtre \| ... }} ou dans les commandes

Filtre Liste

dicsort(value,case_sensitive=False,by='key'))	
first(seq)	last(seq)
groupby(value,attr)	map()
join(value,d='u',attribute=None)	
lenght(object)	llist(value)
reverse(value)	
select()	selectattr()
slice(value,slices,fille_with=None)	
sort(value,reverse,case_sensistive,attribute)	
sum(iterable,attribute=None,start=0)	

Tests	
callable(*object*)	divisibleby(*value,num*)
escaped(*value*)	mapping(*value*)
iterable(*value*)	sequence(*value*)
string(*value*)	number(*value*)
lower(*value*)	upper(*value*)
defined(*value*)	undefined(*value*)
none(*value*)	
even(*value*)	odd(*value*)
equalto(*value,other*)	sameas(*value*)