

什么是过滤器，语法是什么：

1. 有时候我们想要在模版中对一些变量进行处理，那么就必须要类似于Python中的函数一样，可以将这个值传到函数中，然后做一些操作。

在模版中，过滤器相当于是一个函数，把当前的变量传入到过滤器中，然后过滤器根据自己的功能，再返回相应的值，之后再将结果渲染到页面中。

2. 基本语法：{{ variable|过滤器名字 }}。使用管道符号|进行组合。

常用过滤器：

default过滤器：

使用方式{{ value|default('默认值') }}。如果value这个key不存在，那么就会使用default过滤器提供的默认值。如果你想使用类似于python中判断一个值是否为False（例如：

None、空字符串、空列表、空字典等），那么就必须要传递另外一个参数{{ value|default('默认值',boolean=True) }}。

可以使用or来替代default('默认值',boolean=True)。例如：{{ signature or '此人很懒，没有留下任何说明' }}

自动转义过滤器：

1. safe过滤器：可以关闭一个字符串的自动转义。

2. escape过滤器：对某一个字符串进行转义。

3. autoescape标签，可以对他里面的代码块关闭或开启自动转义。

 jinja

 {% autoescape off/on %}

 ...代码块

 {% endautoescape %}

1. first(value)：返回一个序列的第一个元素。names|first。

2. last(value)：返回一个序列的最后一个元素。示例：names|last。

3. length(value)：返回一个序列或者字典的长度。示例：names|length。

4. join(value,d=u'')：将一个序列用d这个参数的值拼接成字符串。

5. safe(value)：如果开启了全局转义，那么safe过滤器会将变量关掉转义。示例：content_html|safe。

6. int(value)：将值转换为int类型。

7. float(value)：将值转换为float类型。

8. lower(value)：将字符串转换为小写。

9. upper(value)：将字符串转换为小写。

- 10. `replace(value,old,new)`: 替换将old替换为new的字符串。
- 11. `truncate(value,length=255,killwords=False)`: 截取length长度的字符串。
- 12. `striptags(value)`: 删除字符串中所有的HTML标签, 如果出现多个空格, 将替换成一个空格。
- 13. `trim`: 截取字符串前面和后面的空白字符。
- 14. `string(value)`: 将变量转换成字符串。
- 15. `wordcount(s)`: 计算一个长字符串中单词的个数。
- 16. `format(value,*arags,**kwargs)`: 格式化字符串。例如以下代码:
`{{ "%s" - "%s"|format('Hello?',"Foo!") }}`
将输出: Helloo? - Foo!

自定义模版过滤器:

过滤器本质上就是一个函数。如果在模版中调用这个过滤器, 那么就会将这个变量的值作为第一个参数传给过滤器这个函数, 然后函数的返回值会作为这个过滤器的返回值。需要使用到一个装饰器: `@app.template_filter('cut')`

```
@app.template_filter('cut')
def cut(value):
    value = value.replace("hello","")
    return value
```

html

```
<p>{{ article|cut }}</p>
```