

Django for 标签详解

通过上一节的《[Django if 标签详解](#)》的讲解，相信大家对于模板标签已经不再陌生，在本节我们再给大家隆重介绍一位最熟悉的陌生人，那就是 for 标签，学习它的时候，我们也要结合 Python 语言中的 for 循环，这样很便于大家理解与使用它。

1. for 标签的使用

for 标签用于对可迭代对象进行遍历，包括列表、元组等，它与 Python 中的 for 语法是类似的。for 标签使用时也需要和 endfor 标签配合使用，当然它也有不同之处，那就是它多了一个可选的 empty 标签，比如用它来显示当列表不存在或者列表中元素为空的时候要显示的内容，它的使用格式如下：

```
{% for 变量 in 可迭代对象 %}  
    ... 循环语句  
{% empty %}  
    ... 可迭代对象无数据时填充的语句  
{% endfor %}
```

我们通过一个具体的实例来看一下它的用法，在 views.py 中添加如下代码：

```
from django.template import Template, Context  
def test_for(request):  
    #调用template()方法生成模板  
    t1=Template("""  
        {% for item in list %}  
        <li>{{ item }}</li>  
        {% empty %}  
        <h1>如果找不到你想要，可以来C语言中文网(网址：http://c.biancheng.net/)</h1>  
        {% endfor %}  
    """)  
    #调用 Context()方法  
    c1= Context({'list': ['Python', 'Java', 'C', 'Javascript', 'C++']})  
    html=t1.render(c1)  
    return HttpResponse(html)
```

并配置路由映射关系，如下所示：

```
path('test_for/',views.test_for)
```

访问 127.0.0.1:8000/test_for 得到如下页面：



图1：for 标签的使用

当我们将列表改为空的时候，或者不是一个可迭代对象的时候，会得到如下页面：



图2：for 标签的使用

提示：与 Python 中的 for 循环不同的是，for 标签只能一次性地遍历完列表中的元素，不能中断（break），也不能跳过（continue）。

2. for 标签嵌套使用

for 标签同样可以嵌套使用，示例代码如下所示：

```
def test01_for(request):  
    #使用嵌套for标签依次遍历列表取值  
    website=Template("""  
        {% for course in list01 %}  
        <div>  
            {% for coursename in course %}  
            <p><b>{{ coursename }}</b></p>  
            {% endfor %}  
        </div>  
        {% endfor %}  
    """)  
    webname=Context({'list01':[[' Django', ' Flask', ' Tornado'], [' c语言中网', ' Django官网', ' Pytho官网']]])  
    html=website.render(webname)  
    return HttpResponse(html)
```

然后为其配置路由映射关系，最后访问 127.0.0.1/test01_for 获得相应页面如下所示：

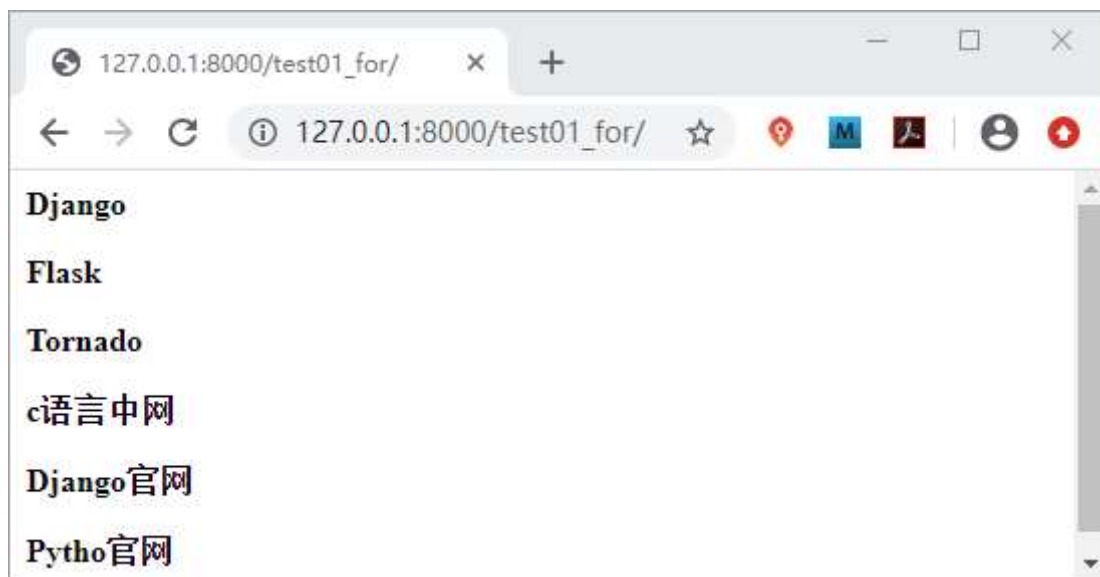


图3：for 循环嵌套使用

3. forloop变量详解

在 for 标签还提供了内置变量 `forloop`，我们可以访问这个变量的属性从而获取 for 循环迭代过程中的一些信息，比如 `forloop.first`，它的返回值是一个布尔值，当 for 循环迭代第一个元素的时候返回 `True`，若有其余元素则返回的是 `False`。`forloop` 的属性主要有以下几个：

forloop变量属性

变量	描述
<code>forloop.counter</code>	用来计数，查看当前迭代第几个元素（从1开始索引）
<code>forloop.counter0</code>	用来计数，查看当前迭代第几个元素（从0开始索引）
<code>forloop.revcounter</code>	表示当前循环中剩余的未被迭代的元素数量（从1开始索引）
<code>forloop.revcounter0</code>	表示当前循环中剩余的未被迭代的元素数量（从0开始索引）
<code>forloop.first</code>	如果当前迭代的是第一个元素，则为 <code>True</code>
<code>forloop.last</code>	如果当前迭代的是最后一个元素，则为 <code>True</code>
<code>forloop.parentloop</code>	在嵌套循环中，用来引用外层循环的 <code>forloop</code>

实例说明如下所示：

```
def test_forloop(request):
    a=Template("""
    {% for item in lists %}
    <div>
    <p><b>{{ forloop.counter }}:{{ item }}</b></p>
    </div>
    {% endfor %}
    """)
    b=Context({'lists': ['c语言中网', 'Django官网', 'Pytho官网']})
    html=a.render(b)
    return HttpResponse(html) #数字与元素以 1:'c语言中文网' 的形式出现
```

最终效果如图所示：

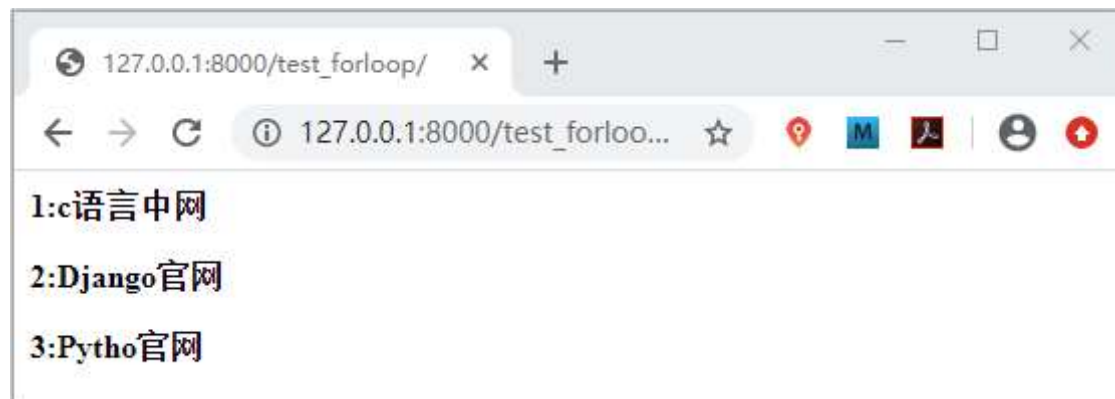


图4：forloop 属性使用方法

提示：forloop 变量只能在 for 与 endfor 之间使用，它的 first 和 last 属性，可以通过 if 标签对当前迭代元素做逻辑判断，而像 counter 属性，常用于调试程序。