[使用Django实现分页器功能](https://www.cnblogs.com/renpingsheng/p/7461193.html)

要使用Django实现分页器,必须从Django中导入Paginator模块

from django.core.paginator import Paginator

假如现在有150条记录要显示,每页显示10条

>>> from django.core.paginator import Paginator#导入Paginator模块

>>> list1=[i for i in range(0,150)]#使用列表生成器生成一个包含150个数字的列表

>>> page1=Paginator(list1,10)#生成一个Paginator对象

>>> print(page1.count)#打印总的记录数,即列表list1的长度

150

>>> print(page1.num\_pages)#打印总的页数,即总记录数除以每页显示的条目数

15

>>> print(page1.page\_range)#页数的列表

range(1, 16)

>>> print(page1.page(1))#打印第一页的page对象

<Page 1 of 15>

>>> page1.page(1).object\_list#打印第一页的所有记录

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

>>> page1.page(2).object\_list#打印第二页的所有记录

[10, 11, 12, 13, 14, 15, 16, 17, 18, 19]

>>> page1.page(2).next\_page\_number()#打印当前的页(第2页)的下一页的页码

3

>>> page1.page(2).has\_next()#第2页是否有下一页

True

>>> page1.page(2).has\_previous()#第2页是否有上一页

True

>>> page1.page(2).has\_other\_pages()#第2是否有其他页

True

>>> page1.page(2).start\_index()#第2页第一条记录的序号

11

>>> page1.page(2).end\_index()#第2页最后一条记录的序号

20

>>> page1.page(0)#第0页是否有记录,会报错

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

...

raise EmptyPage(\_('That page number is less than 1'))

django.core.paginator.EmptyPage: <exception str() failed>

>>> page1.page(15)#打印第15页的对象

<Page 15 of 15>

例子,使用**Django**实现一个分页效果

前端代码:

{% load staticfiles %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<link rel="stylesheet" href="{% static 'bootstrap/css/bootstrap.css' %}">

</head>

<body>

<div class="container">

<h4>分页器</h4>

<ul>

#遍历boot\_list中的所有元素

{% for book in book\_list %}

#打印书籍的名称和价格

<li>{{ book.title }} {{ book.price }}</li>

{% endfor %}

</ul>

<ul class="pagination" id="pager">

{#上一页按钮开始#}

{# 如果当前页有上一页#}

{% if book\_list.has\_previous %}

{# 当前页的上一页按钮正常使用#}

<li class="previous"><a href="/?page={{ book\_list.previous\_page\_number }}">上一页</a></li>

{% else %}

{# 当前页的不存在上一页时,上一页的按钮不可用#}

<li class="previous disabled"><a href="#">上一页</a></li>

{% endif %}

{#上一页按钮结束#}

{# 页码开始#}

{% for num in paginator.page\_range %}

{% if num == currentPage %}

<li class="item active"><a href="/?page={{ num }}">{{ num }}</a></li>

{% else %}

<li class="item"><a href="/?page={{ num }}">{{ num }}</a></li>

{% endif %}

{% endfor %}

{#页码结束#}

{# 下一页按钮开始#}

{% if book\_list.has\_next %}

<li class="next"><a href="/?page={{ book\_list.next\_page\_number }}">下一页</a></li>

{% else %}

<li class="next disabled"><a href="#">下一页</a></li>

{% endif %}

{# 下一页按钮结束#}

</ul>

</div>

</body>

</html>

后端代码:

#导入render和HttpResponse模块

from django.shortcuts import render,HttpResponse

#导入Paginator,EmptyPage和PageNotAnInteger模块

from django.core.paginator import Paginator, EmptyPage, PageNotAnInteger

#从Django项目的应用中导入模块

from app01.models import \*

def index(request):

#获取Book数据表中的所有记录

book\_list=Book.objects.all()

#生成paginator对象,定义每页显示10条记录

paginator = Paginator(book\_list, 10)

#从前端获取当前的页码数,默认为1

page = request.GET.get('page',1)

#把当前的页码数转换成整数类型

currentPage=int(page)

try:

print(page)

book\_list = paginator.page(page)#获取当前页码的记录

except PageNotAnInteger:

book\_list = paginator.page(1)#如果用户输入的页码不是整数时,显示第1页的内容

except EmptyPage:

book\_list = paginator.page(paginator.num\_pages)#如果用户输入的页数不在系统的页码列表中时,显示最后一页的内容

return render(request,"index.html",locals())