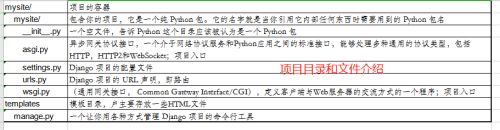


三、项目目录和文件介绍



四、app的创建

在创建的django文件下运行：python manage.py startapp 要创建的app名称

**五、django一些命令，避免忘记**

**六、项目初始化设置**

1.时区

每创建一个项目要先检查当前时区是否为项目要求的时区。

常用时区

上海

LANGUAGE\_CODE = **'zh-hans'**TIME\_ZONE = **'asia/shanghai'**

2.路由（基础）

把项目和app进行绑定

**App/Urls.py基础配置**

*#coding:utf-8***from** django.urls **import** path  
**from** .views **import** TestOne  
  
urlpatterns=[  
 path(**'TestOne'**,TestOne.as\_view(),name=**'testone'**)  
]

**Views.py基础配置**

*#coding:utf-8*

**from** django.views.generic **import** View  
  
**class** TestOne(View):  
 **def** get(self,request):  
 **return ''**

其中TestOne是类名（标红）其他照搬

如果项目中有static，则需要增加以下下代码：ps：这是个元组！最后的，一定要加上

#配置静态文件  
STATICFILES\_DIRS = (os.path.join(BASE\_DIR,'static'),)

TEMPLATES = [  
 {  
 'BACKEND': 'django.template.backends.jinja2.Jinja2',  
 'DIRS': [os.path.join(BASE\_DIR,'templates')],

}]

**七、TEMPLATE**

**2.template的内置标签**

settings.py基础配置

ALLOWED\_HOSTS = [\*]

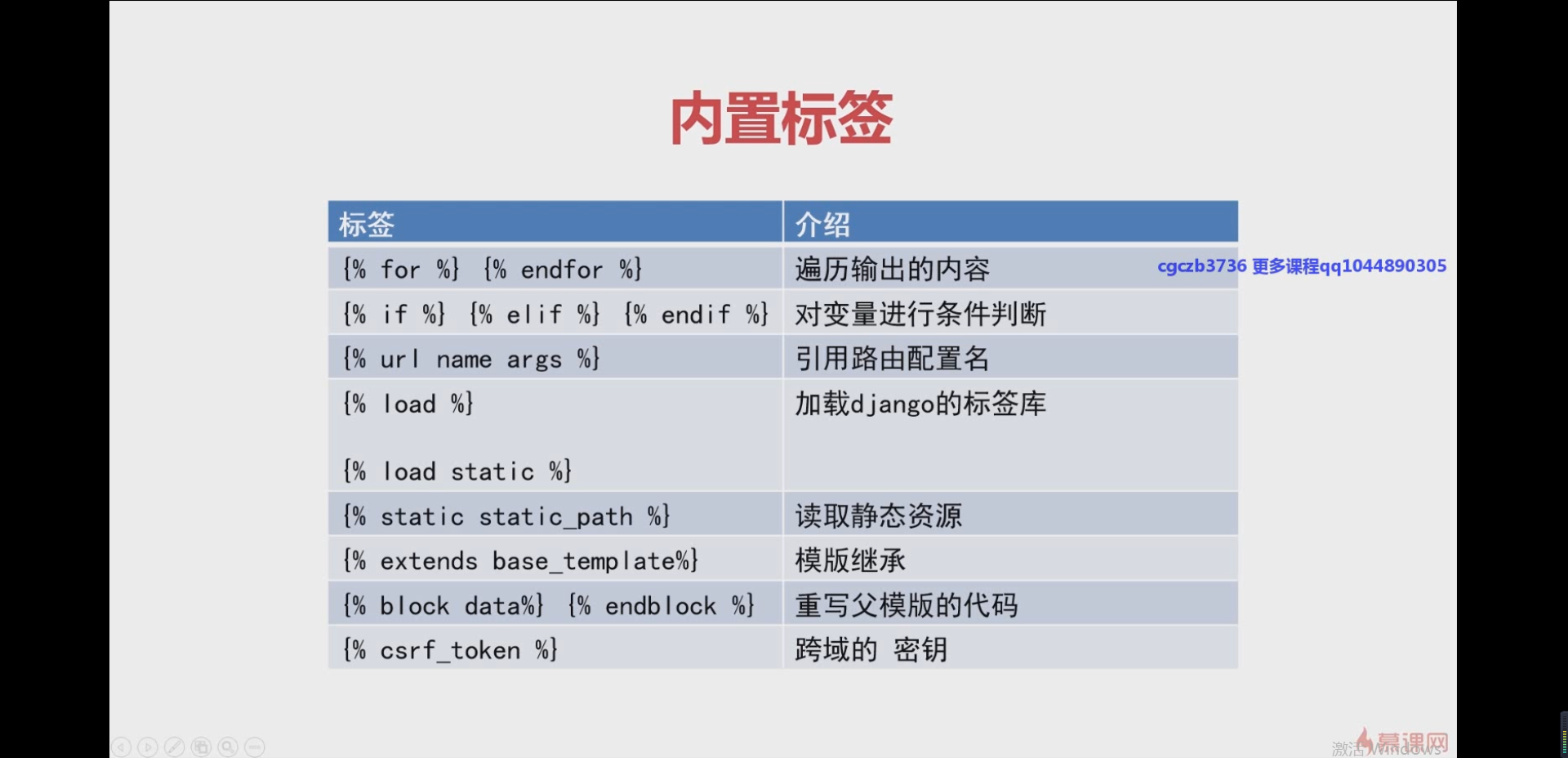
INSTALLED\_APPS = [  
 **'django.contrib.admin'**,  
 **'django.contrib.auth'**,  
 **'django.contrib.contenttypes'**,  
 **'django.contrib.sessions'**,  
 **'django.contrib.messages'**,  
 **'django.contrib.staticfiles'**,  
 **'app'**]

如果项目有template，那么应该在settings.py下将template文件夹注册进去：

urls.py基础配置

**from** django.urls **import** path,include  
**from** django.contrib **import** admin  
**from app** **import** urls **as app**\_urls  
  
urlpatterns = [  
 path(**'admin/'**, admin.site.urls),  
 path(**''**,include(**app**\_urls))  
]

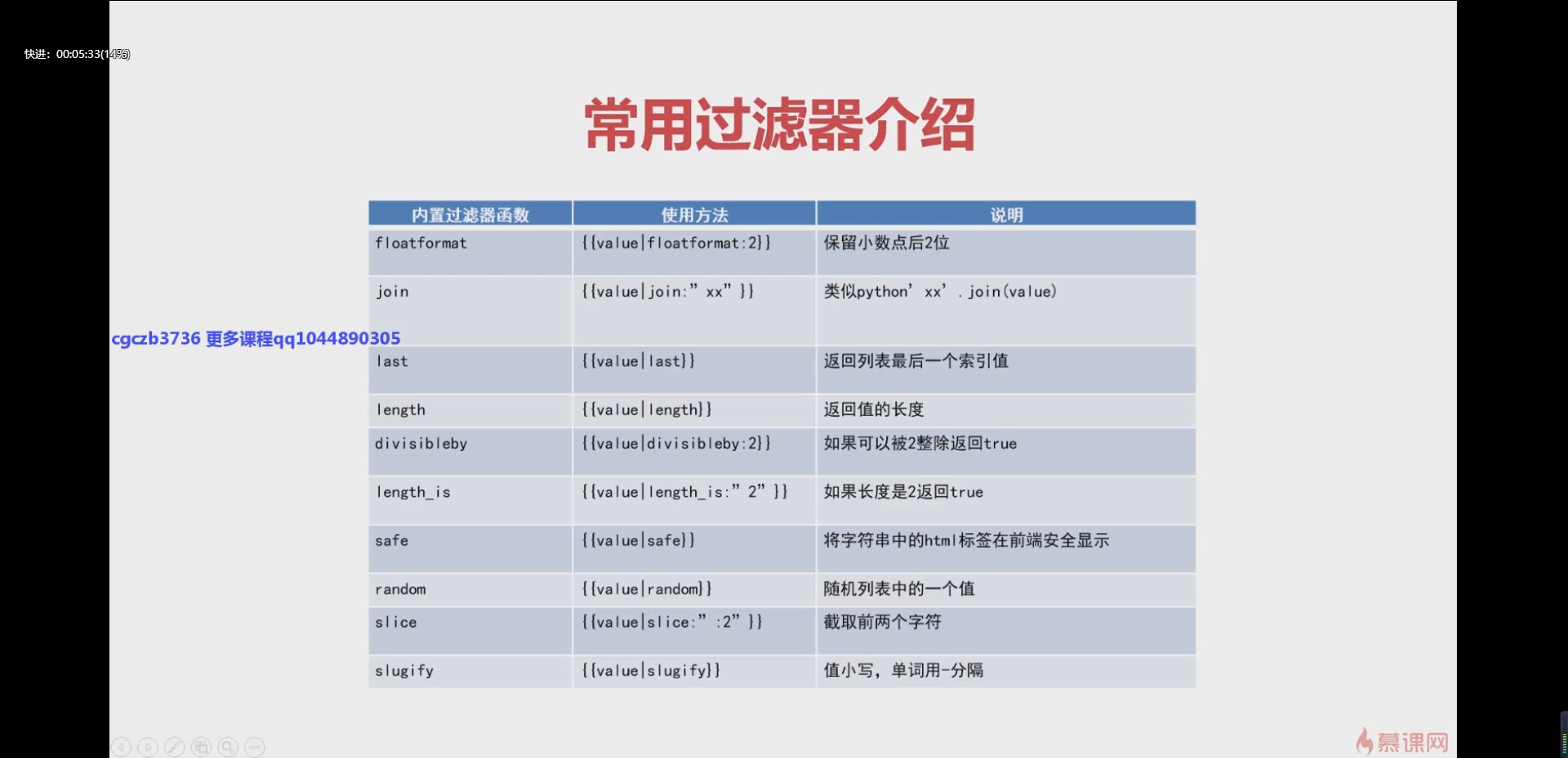
其中app是应用名，其他照搬即可



**3.template的过滤器**

过滤器里空格要使用得当否则会报错







**4.自定义过滤器**

在应用下创建templatetags文件夹，在文件夹下创建myfilter.py（自定义的过滤器需要应用已在settings.py下注册才能被使用，）

下面为示例：

#coding:utf-8  
from django import template#\*  
  
register = template.Library()#\*  
  
@register.filter#\*  
def test(value,args):#\*  
 return value\*args

这是一个简单的乘法过滤器，带‘\*‘的注释是必要的语句，其中红色为可自行更改的部分。

使用时要先导入：{% load myfilter %}

**4.其他的模板引擎**

**Jinja2：**

Jinja2是一套模仿Django模版的模版引擎，由Flask开发者开发，它的使用场景和Django的模版非常相似。它速度快，被广泛使用。

Jinja2提倡让Html设计者和后端Python开发工作分离

jinja2官网:http://jinja.pocoo.org/docs/2.10/



**Jinja2的配置详情见**[**jinja2学习笔记.docx**](jinja2学习笔记.docx)

**MAKO:**

**mako模板算是Python里面比较出色的一个模板了，它宣称有kJinia2更快的解析速度已经更多的语法支持**

**它可以允许你在Html中随意书写python代码**

**MAKO的配置详情见**[**mako学习笔记.docx**](mako学习笔记.docx)