```
def render(request, template_name, context=None, content_type=None, status=None, using=None):
                                         Return a HttpResponse whose content is filled with the result of calling
                                         django.template.loader.render\_to\_string() with the passed arguments.
                                         content = loader.render_to_string(template_name, context, request, using=using)
                                         return HttpResponse(content, content_type, status)
                                      参数说明: (request, template_name, context, content_type, status, using )
                     render函数:
                                      返回结果: HttpResponse对象,其内容是调用jango执行的结果来填充HttpResponse的结果
                                     原理: 先执行, loader.render_to_string('index.html', context=None, request=request, using=None), 把HTML页面转换为字符串
                                          然后: HttpResponsse返回上述执行结果
                                               匹配任何非空字符串, 包含了路径分隔符
                                     捕获: 使用尖括号<>从url中捕获值、捕获值中可以包含一个特化器类型 (converter type)
                                                     (捕获的内容),正则的分组中就是正则的捕获
                                                    捕获的内容未命名时:
                                                                         接受时的参数名可以随便起
                                     正则中捕获:
                                                    捕获内容命名: PP<名字>
                                                <类型: 捕获的类型名>
                                                接受时:参数要与捕获时名字一致
                                     极式
                                                                           context = {key:value, key: value}
                     基本规则:
                                                context(上下文环境)(页面渲染):
                                                                           一个字典类型的值
                                                    str,匹配除了路径分隔符 (/) 之外的非空字符串,这是默认的形式
                                                    int,匹配正整数,包含0。
                                    path转化器:
                                                    slug,匹配字母、数字以及横杠、下划线组成的字符串。abc_123
                                                    uuid,匹配格式化的uuid, 如 075194d3-6885-417e-a8a8-6c931e272f00
                                                                                                备注:
                                                               1.X
                                                                                django.urls.path
                                                                                               新增, url的增强版
                                                               django.conf.urls.include
                                                                                django.urls.include
                                                                                               异名同1不会立即废弃
                                     Django1.x₩Django2.x约区别:
                                                               django.conf.urls.url
                                                                                django.urls.re_path
                     页面渲染:
                                     views有数据要往页面上传,这个就叫页面渲染
                                       Django1.x版本: url
                     正则的使用:
Django第二天
                                      Django2.x 成本: re_path
                                                      主路由:根目录下的路由
                                                     子路由: 应用目录下的路由
                                                                    1、主路由中: path('news/',include('news.urls',namespace='news'))
                                                                    2、在news应用中复制或新建一个urls.py文件,在此文件中的添加一个app_name = 应用名
                     路由的分发:
                                       主/子路由:
                                                                    3、在文件中定义子路由:

    path('',news_index,name='news_index')

                                                     配置子路由:
                                                                    4、在应用中的views.py中就需要定义一个news_index函数
                                                                                                       1、如果模板直接在templates目录下,就可以写xxx.html
                                                                    5、在函数中通过render函数返回一个模板:
                                                                                                       2、但是如果模板在 templates 目录的子目录中,就需要: 子目录/xxx.html
                                           认识模板:
                                                        模板的语法: 就是通过服务器进行翻译
                                                              name属性
                                          做路径的反向解析:
                                                              namespace属性
                                                     注意: namespace, name属性都要添加
                                                     1、在主路由中, 如果有子路由, 则需要添加 ('应用下的urls', namespace='值'),
                                                     如: path('sport/',include('sports.urls',namespace='sports')) 沒有子路由的时候則不需要include
                     路由的反向解析:

    path('',index,name='index')

                                                                                在子路由中正常填写路径:
                                          步骤:
                                                     2、有子路由則需要以下部分:

    p: path('message1', sport_message1, name='sport_message1'),
                                                                                  path('message2', sport_message2, name='sport_message2'),
                                                     3、如何使用"模板",怎么使用?
在Template模板中,只需要有链接的部分則需要:
                                                     格式: <a href='{% url '主路由的namespace的值:子路由的name值' 捕获的参数值 %}'>体育消息1</a>
                                                        如: {% url 'sports:sport_message' 参数值 %}
                                                       1、scheme: 协议
                                                       2、path: 路径, 会把http://ip:端口整个看成根目录"/"
                                                       3、method: 请求方式
                     get/post请求:
                                       request的属性:
                                                        4、encoding: 编码
                                                        GET/POST: 返回的是QueryDict, 是个字典, 存放的是请求的参数, 可用获取传递的参数
```