1. route的定义：

客户端发送请求给web服务器，web服务器再将请求发送给flask程序实例

程序实例需要知道每个url请求所对应的运行代码是谁。所以程序中必须要创建一个url请求地址到python运行函数的一个映射。处理url和函数之间的关系的程序就是“路由”

2.路由的体现

在flask中，路由是通过@app.route装饰器来表示的

1.路由的基本表示

```
#http://localhost:5000/admin/login
 @app.route('/admin/login')
 def admin_login():
      return 'xxx'
```

2.指定参数类型的路由以及传参

```
#http://localhost:5000/show/zengsf/23
 @app.route('/show/<name>/<int:age>')
 def show(name, age):
       //: name: 字符串
       //：age: 整数
```

Flask中所支持的类型转换器

类型转换器	作用
缺省	字符串，不能有斜杠（‘/’）
int	整型
float	浮点型
path	字符串，可以有斜杠（‘/’）

3.多url的路由匹配

为多个访问地址匹配同一个视图处理函数

```
@app.route('/地址1')
@app.route('/地址2')
...  ...
 def index():
  return ""
```

4.路由中设置http请求方法

flask路由也允许设置对应的请求方法（post/get），只有将匹配上请求方法的路径才能交给对应的视图处理函数处理。所有的路由，默认只接受get请求

```
@app.route('/xxx/xxx', methods=['POST'])
    def xxx:
        #该函数只能接收post请求
    pass
@app.route('/xxx/xxx', methods=['GET, POST'])
    def xxx:
        #该函数既能接受post请求也能接收get请求
    pass
```

5.URL的反向解析

正向解析：程序自动解析，根据@app.route()中的访问路径，来匹配处理函数

反向解析：通过视图处理函数的名称自动生成对应的访问路径

在Flask中要实现反向解析的话需要使用 ：

url_for(funName,args)

funName:要生成地址的函数名

args:该地址中需要的参数

函数：s = url_for(funName,arg1=value1,arg2=value2)