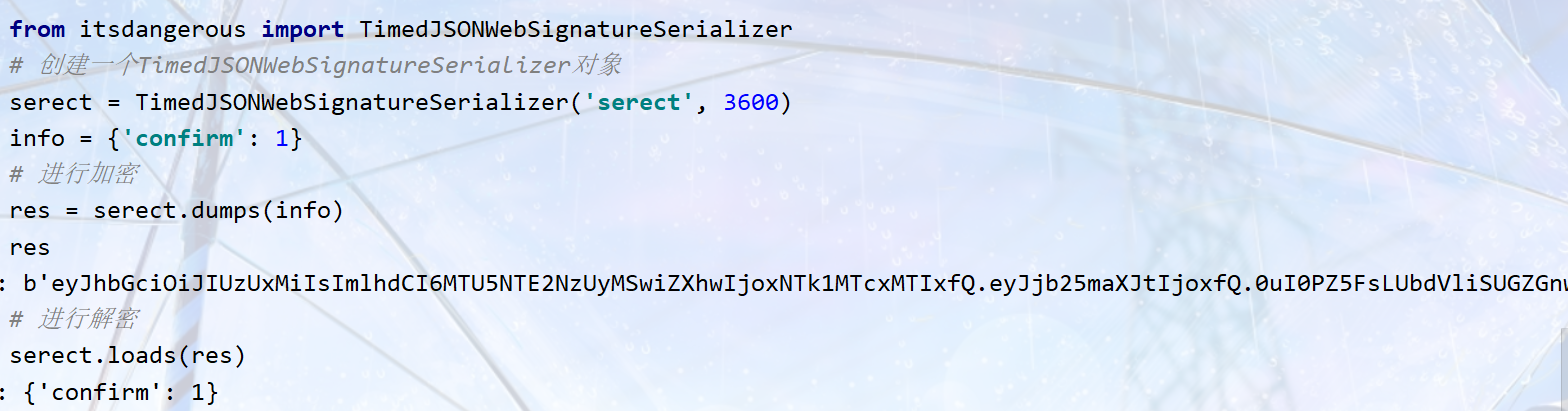
在python中有一个all方法，可以传入一个可迭代对象，如果这个对象里面的内容都存在的话，那么才返回一个true

在设计一个url的时候，以register为例，register页面用于显示注册页面，register\_handle用于注册信息处理，可以把这两个url合并在一起，那么如何将两个功能区分？

在显示页面的时候，采用的是GET方法，信息处理的时候，采用的是POST方法，因此可以通过判断这两种方法来进行判断，然后再处理即可

在发送邮件的时候，邮件中需要有用户信息，但是要对用户信息进行加密，这时候就可以用到itsdangerous，这个里面有一个TimedJSONWebSignatureSerializer模块，这个模块用于加密以及设置过期时间，使用方法如下

1. 先创建一个TimedJSONWebSignatureSerializer对象。第一个参数是加密的密钥（注意：在django的setting文件中有一个SECRET\_KEY，这个密钥本来是django创建项目的时候自己内部使用的，我们在设置密钥的时候可以借助这个密钥，也可以自己设置一个），可以自己设置，第二个参数是过期时间，单位是秒，然后找个对象接收
2. 创建好以上的对象有一个dumps方法，传入的参数就是要加密的对象，得到的返回值就是加密处理后的结果
3. 如果要进行解密的话，最开始创建的对象还有个loads方法，传入加密后的数据，得到的就是解密后的内容



使用django自带的发送邮件

1. 在邮箱中开启smtp/pop3服务
2. 在settings文件中添加以下内容

EMAIL\_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'

EMAIL\_HOST = 'smtp.163.com'

EMAIL\_PORT = 25

#发送邮件的邮箱

EMAIL\_HOST\_USER = 'itcast88@163.com'

#在邮箱中设置的客户端授权密码

EMAIL\_HOST\_PASSWORD = 'python808'

#收件人看到的发件人

EMAIL\_FROM = 'python<itcast88@163.com>'

3、导入from django.core.mail import send\_mail

4、使用send\_mail发送邮件，第一个参数为subject，也就是邮件的主题(标题)，第二个参数为message，也就是邮件的正文，第三个参数为from\_email，也就是发件人是谁，由于上面在settings文件中已经设置了EMAIL\_FROM，因此可以直接使用这个EMAIL\_FROM，

第四个参数为recipient\_list，也就是收件人的列表，如果发送的正文里面有html内容的话，就不可以通过message参数来传，这个函数里面还有个html\_message参数，这个时候，通过关键字参数来传即可，注意，如果第二个参数不传的话，就传入一个空字符即可