

Python_面试题

1、邮箱正则

```
import re
```

```
r=re.findall("[a-zA-Z0-9_-]+@[a-zA-Z0-9_-]+\.[a-zA-Z0-9_-]+", "<ksf_skfkd kfj@sfjlsf.com>")
```

```
print(x)
```

```
:\\Users\\zhaow\\AppData\\Local\\Programs\\Python\\Python37\\python.exe D:/study/python/atp/lib/t.py
```

```
['ksf_skfkd kfj@sfjlsf.com']
```

2、身份证正则(备注过滤不准确，如需准确的可百度，哈哈哈哈哈)

```
import re
```

```
r=re.findall("\\d{17}[a-zA-Z]", "<ssf41060319890218456xdkjf>")
```

```
print(x)
```

```
/Library/Frameworks/Python.framework/Versions/3.7/bin/python3.7 /Users/dongyf/Documents/python/besttest_study/ryg.py
```

```
['41060319890218456x']
```

3、手机号正则

```
import re
```

```
r=re.findall("1[3,4,5,7,8,9]\\d{9}", "<ssf134567898769xdkjf>")
```

```
print(x)
```

```
/Library/Frameworks/Python.framework/Versions/3.7/bin/python3.7 /Users/dongyf/Documents/python/besttest_study/ryg.py
```

```
['13456789876']
```

4、IP正则

```
import re
```

```
r=re.findall("\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}", "<ssf134.234.21.2332xdkjf")
```

```
print(x)
```

```
/Library/Frameworks/Python.framework/Versions/3.7/bin/python3.7 /Users/dongyf/Documents/python/besttest_study/ryg.py
```

```
['134.234.21.233']
```

5、post和get

GET和POST的区别就是：请求的数据GET是在url中，POST则是存放在请求体里面。

GET:一般向服务器获取数据用get请求，get请求的数据都是放在url中的，实质上上和post请求没有太大的区别，当然也可以用来向服务器发送数据。

POST:一般向服务器发送数据用post请求，post请求的数据放在请求体里。

6、session,cookie

<http://www.nnzhp.cn/archives/426>

cookie是存在本地的一对键值，有有效期，超过有效期，需要重新获取，本地清除cookie后也需要重新获取

session一般存在redis等缓存中，有有效期，请求发来的session，超过有效期，验证不通过，需要重新获取，其也是一对键值，用来加强安全性，防止伪造cookie等

7、token

token就是服务端发的一个令牌，身份的校验，一般也有有效期，定期更新，用户发送请求带token，服务端进行比对

8、写代码

数组转字符串

处理多维数组：

```
def merge2(m):
```

```
    if isinstance(m, list):
```

```
        return ''.join(['%s' % d for i in m for d in merge2(i)])
```

```
    else:
```

```
        return ''.join(['%s' % m])
```

或仅处理1维数组

```
def store(m):
```

```
    if isinstance(m,list):
```

```
        return ''.join('%s'%i for i in m)
```

```
    else:
```

```
        return ''
```

字符串转字典

```
import json

def load(a):

    try:

        return json.loads(a)

    except Exception as e:

        return ""

a='{"zhaozhao":{"price":"4.33","num":"5"}}'

print(load(a))
```

9、

2. 我们程序中用到了一个数组 a，数组的每个元素都是一个字典（map/dict），字典的 key/value 都是字符串，字符串中可能包含任意字符。

示例：

```
a[0]['k1'] = 'v1'
a[0]['k2'] = 'v2'
a[1]['A'] = 'XXX'
...
```

实际使用过程中，我们自定义了一个基于字符串的存储结构，数组元素之间用“换行”分割，字典元素之间使用“分号”分割，key/value 之间用“等号”分割。

上述数据序列化之后，应该得到一个字符串：

```
"k1=v1;k2=v2\nA=XXX"
```

请实现一个“保存”函数、一个“加载”函数。

```
text = store(a); //把数组保存到一个字符串中
a = load(text); //把字符串中的内容读取为字典数组
```

请考虑所有边界情况，不要出现 bug。在满足上述需求的前提下，可自行增加一些定。

#数组转字符串

```
def store(a):  
    s=''  
    for j,i in enumerate(a,1):  
        for index,key in enumerate(i,1):  
            s=s+key+"="+i[key]  
            if index<len(i): #最后一个元素无需加分号  
                s=s+";"  
            if j<len(a): #最后一个元素无需加换行符  
                s=s+r"\n"  
    return s  
  
a=[{"k1":"value1"}, {"k2":"value2"}, {"k3":"value3", "k4":"value4"}]  
  
print(store(a))
```

#字符串解析字典数组

```
def load(a):  
    y=[]  
    for i in a.split("\n"):  
        x = {} #每次字典添加完成清空  
        for j in i.split(";"):  
            k,v=j.split("=") #元素分割后装入字典  
            x[k]=v  
        y.append(x) #字典装入完毕加入数组  
    return y
```

```
s="k1=value1\nk2=value2\nk3=value3;k4=value4"
```

```
print(load(s))
```