

```
GigabitEthernet1/1 192.168.121.181 YES NVRAM up up
GigabitEthernet1/2 192.168.110.2 YES NVRAM up up
GigabitEthernet2/1 10.254.254.1 YES NVRAM up up
GigabitEthernet2/2 10.254.254.5 YES NVRAM up up '''
>>> ip_address = re.search(r'\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}', test)
>>> print (ip_address.group())
192.168.121.181
```

我们用正则表达式 `\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}` 作为模式来匹配任意 IPv4 地址,注意我们在分割每段 IP 地址的“.”前面都加了转义符号\,如果不加\,写成 `\d{1,3}.\d{1,3}.\d{1,3}.\d{1,3}`,则将会匹配到 GigabitEthernet1/1 中的“1/1”,原因请读者自行思考。

`print (a.group())`后可以看到只匹配到了 192.168.121.181 这一个 IPv4 地址,这就是 `re.search()`的短板。如果想匹配其他所有 IPv4 地址 (192.168.110.2, 10.254.254.1, 10.254.254.5),则必须用下面要讲的 `re.findall()`。

3. re.findall()

如果字符串中有多个能被模式匹配到的关键词,并且我们希望把它们全部匹配出来,则使用 `re.findall()`。同 `re.match()`和 `re.search()`不一样,**`re.findall()`**的返回值为列表,每个被模式匹配到的字符串内容分别是该列表中的元素之一。

`re.findall()`函数的语法如下。

```
re.findall(pattern, string, flags=0)
```

还是以上面尝试匹配 `show ip int brief` 命令的回显内容中所有 IPv4 地址的例子为例。

```
>>> test = '''Router#show ip int b
Interface IP-Address OK? Method Status Protocol
GigabitEthernet1/1 192.168.121.181 YES NVRAM up up
GigabitEthernet1/2 192.168.110.2 YES NVRAM up up
GigabitEthernet2/1 10.254.254.1 YES NVRAM up up
GigabitEthernet2/2 10.254.254.5 YES NVRAM up up '''
>>> ip_address = re.findall(r'\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}', test)
```