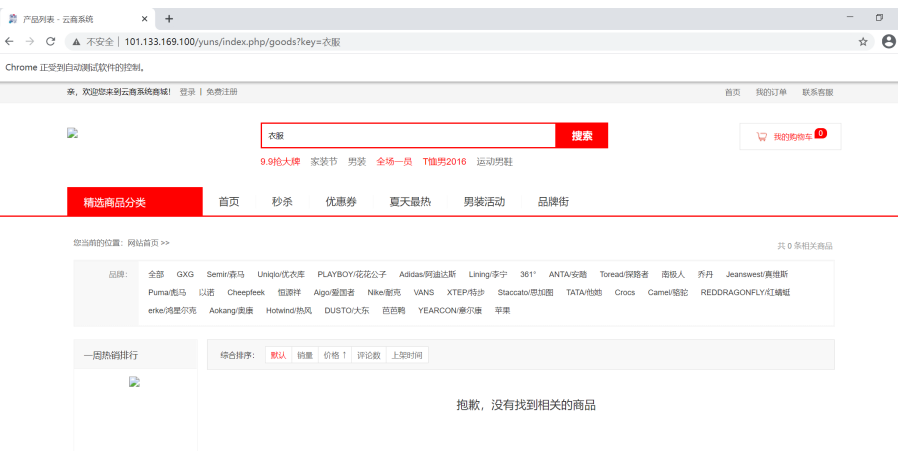
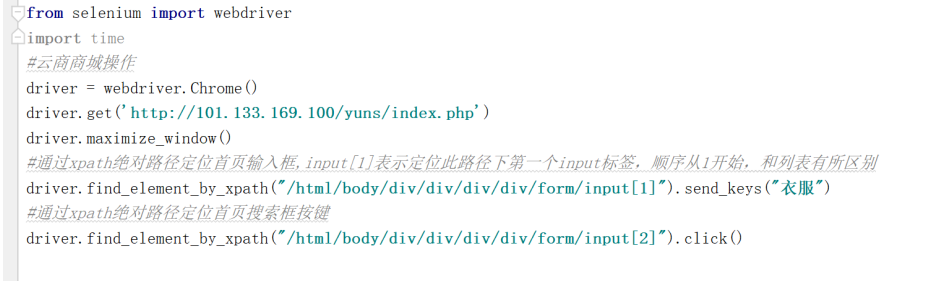
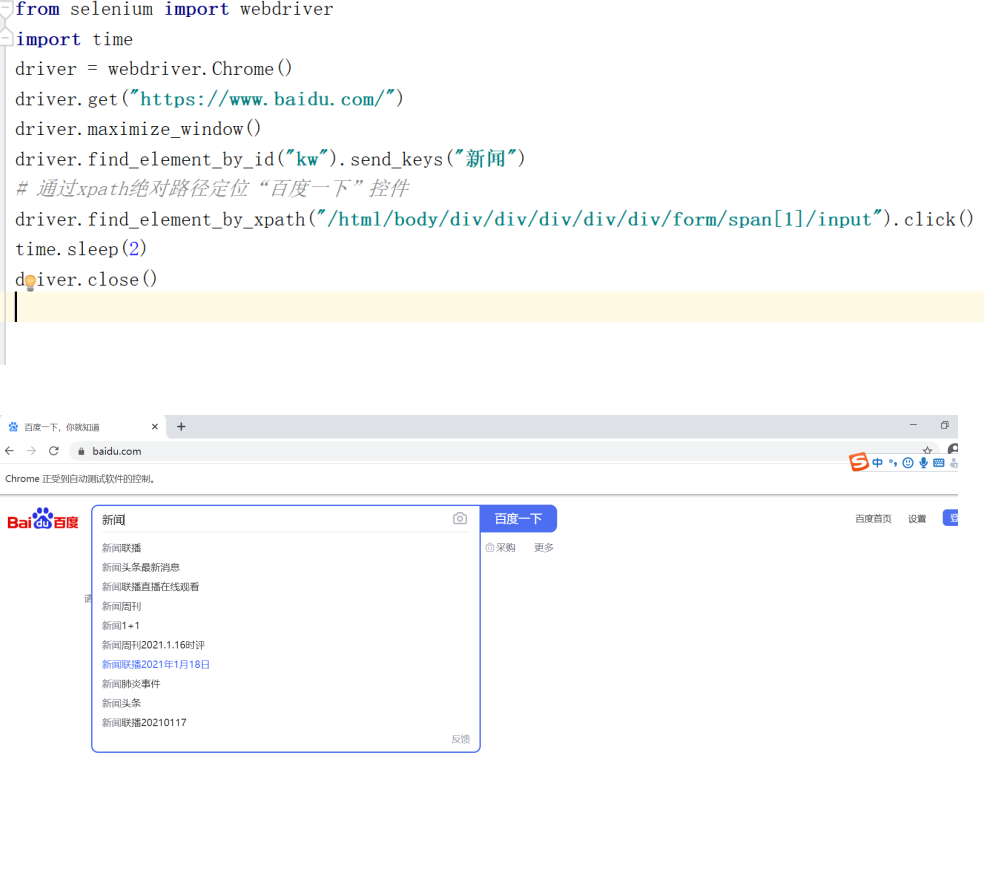
Xpath绝对路径定位方式（**三个案例**）



CSS绝对路径定位（三个案例）



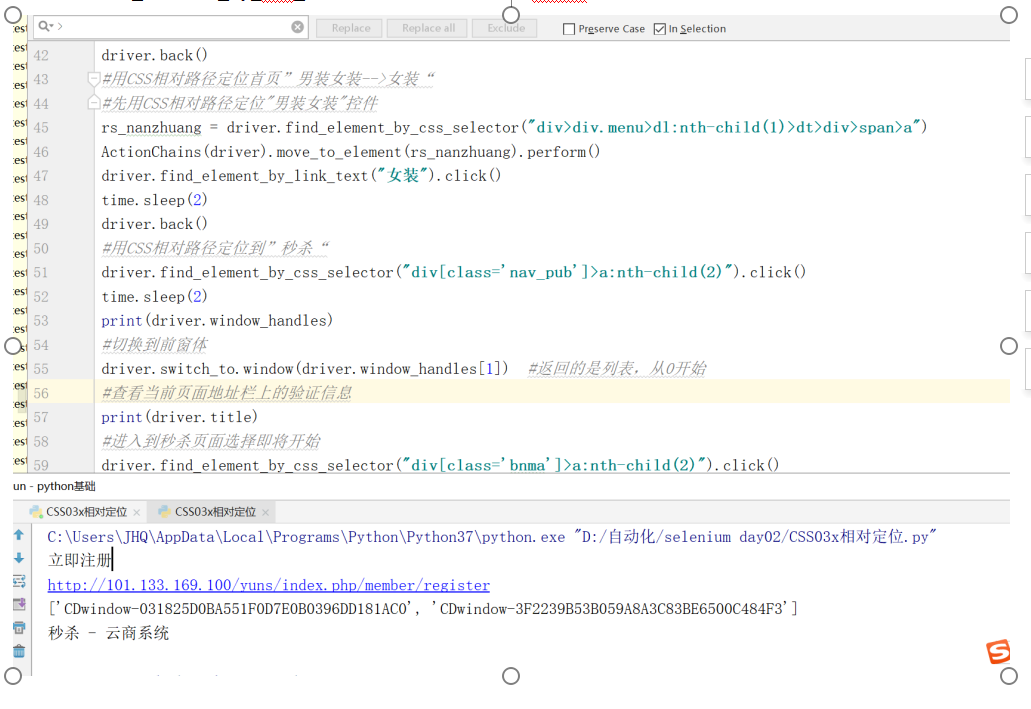
**XPATH的相对定位（15个demo）**

**from**selenium **import**webdriver  
**from**selenium.webdriver.common.action\_chains **import**ActionChains  
**import**time  
#云商商城操作  
driver = webdriver.Chrome()  
driver.get(**'**<http://101.133.169.100/yuns/index.php>**'**)  
driver.maximize\_window()  
#通过xpath相对路径定位首页输入框  
# driver.find\_element\_by\_xpath("//div[@class='schbox']/form/input[1]").send\_keys("dsfds")  
# driver.find\_element\_by\_xpath("//input[@class='but1']").send\_keys("wewew")  
# driver.find\_element\_by\_xpath("//input[@placeholder='请输入你要查找的关键字']").send\_keys("dfd")  
# #通过两个属性确定一个控件  
# driver.find\_element\_by\_xpath("//input[@type='text'and @class='but1']").send\_keys("dsfsda")  
#通过xpath相对路径定位首页搜索框  
# driver.find\_element\_by\_xpath("//input[@class='but2']").click()  
# 通过xpath路径定位到首页登录控件  
**driver.find\_element\_by\_xpath("//div[@class='top']/div[1]/a[1]").click()  
time.sleep(3)**#输入用户名和密码进行登录  
#用xpath相对路径定位到用户名框  
# driver.find\_element\_by\_xpath("//input[@name='username']").send\_keys("931532t45t")  
#删除用户名  
driver.find\_element\_by\_xpath(**"//input[@name='username']"**).send\_keys(**"931532t45t"**)  
  
#用xpath相对路径定位到密码框  
driver.find\_element\_by\_xpath(**"//input[@id='password']"**).send\_keys(**"asfdsate"**)  
#用xpath相对路径定位到登录控件  
driver.find\_element\_by\_xpath(**"//input[@value='登录']"**).click()  
#用xpath相对路径定位忘记密码控件  
driver.find\_element\_by\_xpath(**"//div[@class='login\_reg\_box']/a[1]"**).click()  
time.sleep(3)  
#用xpath相对路径定位立即注册控件  
driver.back()  
time.sleep(2)  
rs=driver.find\_element\_by\_xpath(**"//div[@class='login\_reg\_box']/a[2]"**)  
#查看当前控件属性，文本信息  
print(rs.text)  
print(rs.get\_attribute(**'href'**))  
rs.click()  
time.sleep(2)  
#用xpath相对路径定位QQ登录控件  
driver.back()  
time.sleep(2)  
driver.find\_element\_by\_xpath(**"//div[@class='sub\_con']/a[1]"**).click()  
driver.back()  
driver.back()  
#用xpath相对路径定位首页”男装女装-->女装“  
#先用xpath相对路径定位男装女装  
rs\_nanzhuang = driver.find\_element\_by\_xpath(**"//div[@class='mleft menu']/dl[1]/dt/div/span/a"**)  
ActionChains(driver).move\_to\_element(rs\_nanzhuang).perform()  
driver.find\_element\_by\_link\_text(**"女装"**).click()  
time.sleep(2)  
driver.back()  
#用xpath相对路径定位到”秒杀“  
driver.find\_element\_by\_xpath(**"//div[@class='nav\_pub']/a[2]"**).click()  
time.sleep(2)  
print(driver.window\_handles)  
#切换到前窗体  
# driver.switch\_to.window(driver.current\_window\_handle)  
# print(driver.current\_window\_handle)  
driver.switch\_to.window(driver.window\_handles[1])  #返回的是列表，从0开始  
#查看当前页面地址栏上的验证信息  
print(driver.title)  
#进入到秒杀页面选择即将开始  
driver.find\_element\_by\_xpath(**"//div[@class='bnma']/a[2]"**).click()



**CSS的相对定位15个demo**

**from**selenium **import**webdriver  
**from**selenium.webdriver.common.action\_chains **import**ActionChains  
**import**time  
#云商商城操作  
driver = webdriver.Chrome()  
driver.get(**'**<http://101.133.169.100/yuns/index.php>**'**)  
driver.maximize\_window()  
#通过CSS相对路径定位首页输入框  
# driver.find\_element\_by\_css\_selector("div.schbox>form>[input:first-child](http://inputfirst-child/)").send\_keys("dsfds")  
# driver.find\_element\_by\_css\_selector("input[class='but1']").send\_keys("wewew")  
# driver.find\_element\_by\_css\_selector("input[placeholder='请输入你要查找的关键字']").send\_keys("dfd")  
# driver.find\_element\_by\_css\_selector("input[type='text'][class='but1']").send\_keys("dsfsda")  
#通过CSS相对路径定位首页搜索按键  
# driver.find\_element\_by\_css\_selector("input[class='but2']").click()  
# 通过CSS路径定位到首页"登录"控件  
driver.find\_element\_by\_css\_selector(**".login>a:nth-child(1)"**).click()  
# time.sleep(3)  
# #输入用户名和密码进行登录  
# #用CSS相对路径定位到用户名框  
driver.find\_element\_by\_css\_selector(**"input[name='username'][id='text']"**).send\_keys(**"931532t45t"**)  
# #用CSS相对路径定位到密码框  
driver.find\_element\_by\_css\_selector(**"input#password"**).send\_keys(**"asfdsate"**)  
# #用CSS相对路径定位到登录控件  
driver.find\_element\_by\_css\_selector(**"input[value='登录']"**).click()  
# #用CSS相对路径定位忘记密码控件  
driver.find\_element\_by\_css\_selector(**"div.login\_reg\_box>a:nth-child(1)"**).click()  
time.sleep(3)  
# #用CSS相对路径定位立即注册控件  
driver.back()  
time.sleep(2)  
rs=driver.find\_element\_by\_css\_selector(**"div[class='login\_reg\_box']>a:nth-child(2)"**)  
# #查看当前控件属性，文本信息  
print(rs.text)  
print(rs.get\_attribute(**'href'**))  
rs.click()  
time.sleep(2)  
# #用CSS相对路径定位QQ登录控件  
driver.back()  
time.sleep(2)  
driver.find\_element\_by\_css\_selector(**"div[class='sub\_con']>a:first-child"**).click()  
driver.back()  
driver.back()  
#用CSS相对路径定位首页”男装女装-->女装“  
#先用CSS相对路径定位"男装女装"控件  
rs\_nanzhuang = driver.find\_element\_by\_css\_selector(**"div>div.menu>**[dl:nth-child(1)](http://dlnth-child(1)/)**>dt>div>span>a"**)  
ActionChains(driver).move\_to\_element(rs\_nanzhuang).perform()  
driver.find\_element\_by\_link\_text(**"女装"**).click()  
time.sleep(2)  
driver.back()  
#用CSS相对路径定位到”秒杀“  
driver.find\_element\_by\_css\_selector(**"div[class='nav\_pub']>a:nth-child(2)"**).click()  
time.sleep(2)  
print(driver.window\_handles)  
#切换到前窗体  
driver.switch\_to.window(driver.window\_handles[1])  #返回的是列表，从0开始  
#查看当前页面地址栏上的验证信息  
print(driver.title)  
#进入到秒杀页面选择即将开始  
driver.find\_element\_by\_css\_selector(**"div[class='bnma']>a:nth-child(2)"**).click()



Xpath定位

场景：控件没有class，name,id等属性，前面5种方式没有办法定位到，通过xpath去定位

概念：对于页面来说就是一个dom树，随便一个控件都是dom树->主干和分支

xpath两种定位方式（ctrl+f，建议用chrome）

1.绝对路径：根节点一层一层往下找，一个完整的路径，找到控件

绝对路径较为复杂，一般是用相对路径

2.相对路径：//尽量用 name,id,class这些固定的属性，

通过中间的某一一个特有的属性去定位元素（举例：班级中第四排同学戴了一个红色的帽 子，全班唯一的，就可以通过这个同学找到这个同学的后面的同学，就不用从第一个找起）；通过id，class,pachole等控件的属性（注意：有可能不唯一，回车键看是否会来回跳动到其他地方）

1）.//标签名[@属性=“属性值”]

2）如果一个属性不能准确定位,可通过多个属性进行定位确定控件： //标签名[@属性1=“属性值1”and @属性2=‘属性值2’ ]

3）某一个属性的属性值包含：//input[contains(@placeholder,'关键字')]，

4）返回上一级：//input[contains(@placeholder,'关键字')]/..

5）正则匹配：//\*[@属性=‘属性值’]

CSSselector定位

 1.绝对路径（>分割）

这种定位方式写出来会很长，如果要用的话建议用xpath

2.相对路径

1） 通过id：

标签名#id名（如：i#cart\_num）

直接#id名 找出id为这个名字的所有标签

2）通过class

标签名.class名（也可不加标签，和id同理）

3）没有class，id（和xpath区别是少了@），通用写法（class和id通用）

input[placeholder='请输入你要查找的关键字'][]

4) 通过多个属性定位

driver.find\_element\_by\_css\_selector("input[type='text'][class='but1']")

举例子：div.schbox>form>[input:nth-child(1)](http://inputnth-child(1)/) =first-child nth-chlid(1)和xpath中 input[1]是一样

div[class='schbox']>form>[input:nth-last-child(2)](http://inputnth-last-child(2)/) 从后面开始第二个元素

注意：对于css定位来说：把所有标签类型在一块排序

find\_elements\_by\_tag\_name

找的是标签类型（一组标签），不能精确定位

在自动化过程中常用的一些特殊的API，不可能只用定位就把所有控件都定位到如：鼠标的移动事件，键盘的操作事件，其他一些特殊属性的获取

获取验证信息

应用场景，地址栏以上的区域用是无法定位到的

dirver.title

点击联系客服：产生url地址是否是当前地址

driver.current\_url

控制浏览器

应用场景

比如点到一个新的页面想要返回到上级页面或者前进到下一个页面，或者刷新页面

driver.refresh()

driver.back()

driver.forward()

设置窗体大小

打开一个浏览器，未放大，基于这种情况想要获取该该页面的高和宽

未放大情况下，有些控件被挡住看不到或者无法操作，所以在这种情况下，可能会把窗体设置到某 个固定的尺寸（高和宽），也有可能会把窗体最大化

driver.maximize.window()

driver.set.window.size(1200,800)

动作事件

click:所有控件都可以click(button,输入框，文本都可以click)

send\_keys（只能用于输入框）

情况1：如果输入框先输入了内容，再想继续输入，先把光标放在已有内容后面(即先click再 send\_keys)

情况2：如果输入了内容，此时想清空

.clear()（只能用于输入框）

属性获取的几种方式：

1. 比如获取云商系统的输入框尺寸

1）先要定位到此控件

2）get\_element\_by\_xpath('').szie，返回一个宽高的字典(size是个属性方法，在用的时候直接.size即调用)；如果想取height:print(size['height'])

2.获取控件文本信息：.text（图片上的文本信息是拿不到的）

1）先定位到控件，2）再.text

应用场景：登录云商系统，依次显示：亲欢迎来到云商商城，登录，注册，对比文案信息是不是我想要的，实际结果和预期结果比对，比对控件上的文本信息

3.拿到某一个控件里面的属性值

.get\_attribute('herf')

比如点击9.9,右击检查元素，可以看到有herf属性，class属性，title属性

4.加载框未加载出来不能进行输入，因此要判断输入框是否显示出来

.is\_displayed():返回的是bool值

5.在输入框输入内容，想要拿到回显值(明文)

.get\_attribute('value')

鼠标事件：（from selenium.webdriver.common.action\_chains import ActionChains）

应用场景：鞋子，包包下直接点男鞋

1.怎样去模拟：

1）ele = driver.find\_element\_by\_link\_text('母婴玩具')

2）ActonChains(driver).move\_to\_element(ele).perform() 类名（控制哪个浏览器）.方法名（要挪到的控件）

3）driver.find\_element\_by\_link\_text(‘营养辅食’).click()

2.ActionChains(driver).context\_click(ele).perform（） 右击

3.actionChains(driver).double\_click(ele).click() 双击

4..click 左击

5.某一个控件在一个方向上的偏移

拖拽：比如携程上的验证码条

ActionsChains(driver).drag\_and\_drop\_by\_offset(定位到的元素，100,0)

6.从一个控件的位置到另一个控件的位置：

source = driver.find\_element\_by\_xpath()

target = driver.find\_element\_by\_xpath()

ActionChains(driver).drag\_and\_drop\_by\_offset(source,target)

键盘事件(一般用在input标签，输入框)

使用键盘事件需要导入keys类（from selenium.webdriver.common.keys imprt Keys）

1. 使用场景：比如一个输入框输入内容，想要去删掉一部分内容

回退一格：

定位到的元素.send\_keys（Keys.BACK\_SPACE）

2. 输入空格

.send\_keys(Keys.SPACE)

3.输入内容，ctr+a ctrl+c ctrl+v

.send\_keys(Keys.CONTROL,'a')

三种等待时间（等待时间不够会报错）

什么场景要去用到等待时间：新打开一个页面，页面跳转

为什么要加等待时间：比如打开一个地址，页面上的控件还未加载出来，直接去定位或操作他就会报错，代码是毫秒级别的，会马上去执行

1.强制等待

time.sleep() 强制等待

面试：时间怎么写，写脚本过程中调试，1s，2s，8s，再最慢的时间加上一个冗余时间（最慢8s，加冗余时间可以写9s）

time.sleep和sleep（）有什么区别？前者：import time 后者：from time import sleep

2.隐式等待时间 driver.implicity\_wait()

和强制等待时间的区别：例如参数写为10s，表示等待的最长时间，

注意：是个全局变量，如果在后面写了五行都是定位控件，那么后面的控件每个都是最大等待10s，如果一个定位控件后面写了time.sleep()则会被打断

优点：不需要等待冗余的时间 ，效率比较高；缺点：如果两个页面的xpath路径一样，当还在加载时，没有跳转到第二个页面，就会定位到第一个页面的xpath路径

3.显示等待时间

导入三个类：from selenium .webdriver.support.ui import WebdriverWait

from selenium.webdriver.common.by import By

from selenium.webdriver.support import expected\_conditions as EC

多窗口

具体场景：百度输入内容

driver.window\_hadles：获取所有窗体的基本信息

print(driver.current\_window\_handles):打印当前窗体的信息 返回的是列表

driver.switch\_to.window(driver.window\_handles[1]) 切换到第二个窗体

driver.close() 关闭焦点所在的窗体

driver.quit() 会关闭所有窗体

弹出框alert处理：

什么是alert:鼠标右击弹不出来查看元素的一种弹窗

给alert driver.switch\_to.alert.send\_keys('test')

driver.switch\_to.alert.text 获取alert的文本

driver.switch\_to.alert.accept () 确定

driver.switch\_to.alert.dissmiss() 取消

正向的按钮用accept 逆向的按钮用dismiss

截图方法

driver.get\_screenshot\_as\_file('路径+名称')

alert界面截不了图，如果有alert先dissmiss或者accept后再截图

iframe处理:（163邮箱事例）（类比于拿取柜子里的抽屉里的东西）

背景：在自动化过程中，在有些场景某些控件定位不到，所以需要切换到相应的iframe

面试过程中经常会问的问题：no such element 定位不到控件一般有哪几种原因（1.定位方式不对；2.等待时间不够（控件还没加载出来）3.多窗口（想要的控件是在另外一个窗口）4.iframe原因）

第一种方式：

driverswitch\_to.frame('iframe的id或者name')

第二种方式：

如果id和name属性值是动态变化的，或者没有id或name 那么就要换种方式切换

1）先定位到元素ele

2)driver.switch\_to.frame（ele）

如果有多个iframe，iframe<-iframe<-iframe，此时需要逐层切换到相应的iframe

切进去想切出来：逐层往回切

1. driver.switch\_to.parent\_frame() 从子frame切换到父frame

2.driver.switch\_to.default\_content() 切回主文档，直接切回去

select下拉选择框处理

场景：携程旅行，选择房间数

1.先要导入select类 from selenium.webdriver.support.select import Select

2.定位到下拉选择框 ele

3.Select(ele).select\_by\_visible\_text('2间') 从下拉选择框中选择某一个条件;也可以Select(ele).select\_by\_index(下标) 注意：下标是从0开始的;还可以Select(ele).select\_by\_value('value值')

时间控件处理：

携程旅行

1.先定位到时间控件进行clear处理；再send\_keys输入日期，一定要按照页面定义的日期格式输入

2.设置了readonly的时间控件，是无法输入的

处理方法：

js = "documen.getElementById('noticeEndTime').removeAttribute('readonly')" 定义了js

d.excute\_script(js) 执行js

d.find\_element\_by\_name('').send\_keys

有些控件没有id，有name:

滚动条处理：

场景：比如要去获取一个页面最底部的控件，就需要滚动到页面的底部再定位，也是需要用到js实现

使页面到达最底部：js=documen.documentElement.scrollTop = 10000，一般设置成10000,10000已经足够大，能够到达底部

driver.execute\_script(js)

1)页面滚到底部后又想滚到顶部: scrollTop = 0

2)移动到页面底部：

js=documen.documentElement.scrollTop = 10000

3)滚动页高的50%：

driver.execute\_script("window.scrollTo(0, document.body.scrollHeight\*0.5)");

4)从当前坐标到新坐标：by 相对一个偏移

driver.execute\_script("window.scrollBy(0,-200)");

5）移动到绝对坐标：to 不管起始点在哪，到一个固定点

driver.execute\_script("window.scrollTo(0, 1500)");

文件上传(编测编学-技术交流)

正常的普通的文件控件，type是file类型