一、css定位

```
"""
css定位
基于find_element_by_css_selector()实现
"""
import time

from selenium import webdriver
from selenium.webdriver.common.by import By

 driver=webdriver.Chrome()
 driver.get("https://www.baidu.com")


 # 通过绝对路径进行定位 一般不用
 el1=driver.find_element_by_css_selector("html body div div div div div form span input ")
 el2=driver.find_element_by_css_selector("html>body> div> div> div> div > div> form> span> input ")


 # 2通过id定位 #id属性值
 # 3通过class属性定位 .class属性值 s_ipt
 el3=driver.find_element_by_css_selector("#kw")

 el4=driver.find_element_by_css_selector('.s_ipt')
 # 4通过其他属性定位，并且多个属性定位
 el5=driver.find_element_by_css_selector("[autocomplete='off']")
 el6=driver.find_element_by_css_selector("[autocomplete='off'][class='s_ipt']")
 # 5通过标签定位 标签名+属性/id/class进行定位 组合定位
 el7=driver.find_element_by_css_selector("input#kw")
 el8=driver.find_element_by_css_selector("input.s_ipt")
 el9=driver.find_element_by_css_selector("input[autocomplete='off']")

 #6通过层级定位 层级之间通过>或者空格隔开 相对路径
 el10=driver.find_element_by_css_selector("form#form>span>input#kw")

 # 通过兄弟节点定位
 # 场景：同一个元素下面多一个相同的元素 多胞兄弟
 # 第一个元素 标签：first-child
```

```
# 第二个元素 标签：nth-child(n)
# 最后元素 标签:last-of-type

el11=driver.find_element_by_css_selector("div#s-top-left>a:first-child")
el12=driver.find_element_by_css_selector("div#s-top-left>a:nth-child(3)")

el13=driver.find_element_by_css_selector("div#s-top-left>a:last-of-type")
# el13.click()
# el10.send_keys("chromedriver")
# time.sleep(2)
# driver.close()

"""
定位多个元素
"""
ellist=driver.find_elements_by_css_selector("#kw")
print(ellist)
# 返回WebElement
el14=ellist[0]


"""
元素定位是否通过一个方法，支持所有的定位方式定位到元素
find_element()
find_elements() 基于多个定位方式找到一组元素
"""

el15=driver.find_element(By.CSS_SELECTOR,"#kw")

el16=driver.find_element(By.ID,"kw")

el14.send_keys("chromedriver")
time.sleep(2)
driver.close()

"""
webdriver底层关于元素定位 8+8+2=18

"""
```

二、元素的常用操作

四大操作:

1、输入

2、点击

3、获取文本

4、获取属性

```
1
2  """
3  元素四大操作
4  """
5  import time
6  from selenium import webdriver
7  from selenium.webdriver.common.by import By
8
9  driver=webdriver.Chrome()
10  driver.get("http://www.baidu.com")
11  time.sleep(2)
12  el1=driver.find_element(By.ID,"kw")
13  # 输入
14  # el1.send_keys()
15  # 点击
16  # el1.click()
17  # 获取元素文本内容
18  el2=driver.find_element(By.LINK_TEXT,"新闻")
19  print("打印该元素的文本内容:",el2.text)
20  # 获取元素的属性
21  print("获取autocomplete属性值:",el1.get_attribute("autocomplete"))
22
23
```

三、三大切换

1、切换窗口:当定位的元素不在当前窗口，则需要切换窗口

```
1
2  import time
3  from selenium import webdriver
4  from selenium.webdriver.common.by import By
5  from selenium.webdriver.support.wait import WebDriverWait
6  from selenium.webdriver.support import expected_conditions as EC
```

```
7
8  """
9  元素等待：
10  1、强制等待 time.sleep(秒数) 停留
11  2、智能等待（隐士等待） driver.implicitly_wait(秒数) 给页面上所有的元素设置全局等待时间
12  只要在设置的时间范围内找到了元素，就会执行下一个代码，最多等待设置的时间
13  3、显示等待 显示等待：等待当前需要操作的元素 基于多种条件+等待元素
14   多种条件:等待元素可见？等待url跳转为xxx？等待新窗口出现？很多场景条件
15   from selenium.webdriver.support.wait import WebDriverWait
16   from selenium.webdriver.support import expected_conditions
17  自动化测试框架脚本：
18   以显示等待为主
19   以强制等待为辅
20  """
21
22
23  driver=webdriver.Chrome()
24  # 智能等待
25  # driver.implicitly_wait(10)
26  driver.get("http://www.baidu.com")
27  # 输入搜索内容
28  el1=driver.find_element(By.ID,"kw")
29  el1.send_keys("chromedriver")
30  #点击百度一下
31  el2=driver.find_element(By.ID,'su')
32  el2.click()
33  # 显示等待
34  loc=(By.LINK_TEXT,"ChromeDriver Mirror")
35  # 等待元素存在
36  WebDriverWait(driver,15,0.5).until(EC.presence_of_element_located(loc))
37  # 点击搜索的内容
38  el3=driver.find_element(*loc)
39  el3.click()
40  # 新打开的窗口里面定位元素 需要切换窗口
41  # 获取浏览器窗口列表 最早打开的窗口放到list的最前面
42  wins=driver.window_handles
43  print(wins)
44  # 切换最后打开的窗口
45  driver.switch_to.window(wins[-1])
46
```

```
el4=driver.find_element(By.LINK_TEXT,"2.0/")
el4.click()
# 为了看到效果
time.sleep(3)
driver.close()


```

2、切换iframe：当定位的元素在frame/iframe，则需要切换

```
"""
定位的元素包含在iframe/frame标签里面
切换到iframe/frame
"""
import time

from selenium.webdriver.common.by import By
from selenium import webdriver

driver=webdriver.Chrome()
driver.get("https://ke.qq.com/?tuin=80805fad")

driver.find_element(By.ID,"js_login").click()
time.sleep(2)
# 切换iframe 方式一：id
# driver.switch_to.frame("id")
# 切换iframe 方式二：索引 索引值从0开始
driver.switch_to.frame(1)
# 切换iframe 方式三：找到iframe元素
# driver.switch_to.frame(driver.find_elements(By.XPATH,"//iframe")[2])
# 再定位元素
driver.find_element(By.LINK_TEXT,"帐号密码登录").click()
time.sleep(2)
driver.close()
```

3、切换弹出窗口

下节课讲解

四、元素三大等待

```

import time
from selenium import webdriver
```

```
from selenium.webdriver.common.by import By
from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC

"""
元素等待：
 1、强制等待 time.sleep(秒数) 停留
 2、智能等待（隐士等待） driver.implicitly_wait(秒数) 给页面上所有的元素设置全局等待时间
 只要在设置的时间范围内找到了元素，就会执行下一个代码，最多等待设置的时间
 3、显示等待 显示等待：等待当前需要操作的元素 基于多种条件+等待元素
  多种条件:等待元素可见？等待url跳转为xxx？等待新窗口出现？很多场景条件
  from selenium.webdriver.support.wait import WebDriverWait
  from selenium.webdriver.support import expected_conditions
 自动化测试框架脚本：
  以显示等待为主
  以强制等待为辅
 """


driver=webdriver.Chrome()
# 智能等待
# driver.implicitly_wait(10)
driver.get("http://www.baidu.com")
# 输入搜索内容
el1=driver.find_element(By.ID,"kw")
el1.send_keys("chromedriver")
#点击百度一下
el2=driver.find_element(By.ID,'su')
el2.click()
# 显示等待
loc=(By.LINK_TEXT,"ChromeDriver Mirror")
# 等待元素存在
WebDriverWait(driver,15,0.5).until(EC.presence_of_element_located(loc))
# 点击搜索的内容
el3=driver.find_element(*loc)
el3.click()
# 新打开的窗口里面定位元素 需要切换窗口
# 获取浏览器窗口列表 最早打开的窗口放到list的最前面
wins=driver.window_handles
```

```
print(wins)
# 切换最后打开的窗口
driver.switch_to.window(wins[-1])

el4=driver.find_element(By.LINK_TEXT,"2.0/")
el4.click()
# 为了看到效果
time.sleep(3)
driver.close()

```

作业：实现的业务流程，进入腾讯课堂首页，完成登录的操作