1) # setup webdriver

from selenium import webdriver

from selenium import webdriver  
from selenium.webdriver.chrome.options import Options

chromedriverpath = r'C:\tools\chromedriver\chromedriver.exe'  
chrome\_options = Options()  
chrome\_options.add\_argument('--ignore-certificate-errors')  
chrome\_options.add\_argument("--disable-web-security")  
*# chrome\_options.add\_argument("--incognito")*chrome\_options.add\_argument("--allow-running-insecure-content")  
chrome\_options.add\_argument("--allow-cross-origin-auth-prompt")  
chrome\_options.add\_argument("--disable-cookie-encryption")  
chrome\_options.add\_argument('--disable-dev-shm-usage')  
chrome\_options.add\_argument('--disable-default-apps')  
chrome\_options.add\_argument('--disable-prompt-on-repost')  
chrome\_options.add\_argument("--disable-zero-browsers-open-for-tests")  
chrome\_options.add\_argument("--no-default-browser-check")  
chrome\_options.add\_argument("--test-type")  
prefs = {"profile.default\_content\_setting\_values.notifications" : 2}  
chrome\_options.add\_experimental\_option("prefs", prefs)

driver = webdriver.Chrome(chromedriverpath, options=chrome\_options)

2) Get all children elements from parent element

# you can achieve it by:

# find\_elements\_by\_css\_selector("\*") or find\_elements\_by\_xpath(".//\*").

# parent element / header’s element

driver.get("http://www.stackoverflow.com")

header = driver.find\_element\_by\_id("header")

# start from your target element, here for example, "header"

all\_children\_by\_css = header.find\_elements\_by\_css\_selector("\*")

all\_children\_by\_xpath = header.find\_elements\_by\_xpath(".//\*")

print 'len(all\_children\_by\_css): ' + str(len(all\_children\_by\_css))

print 'len(all\_children\_by\_xpath): ' + str(len(all\_children\_by\_xpath))

3) find element from display text

driver.find\_element\_by\_xpath("//\*[contains(text(),'Currencies')]")

4) send ‘input data’ to element

from selenium.webdriver.common.keys import Keys

# driver.find\_element\_by\_id("username-real").clear() # sometimes you need this

driver.find\_element\_by\_id("username-real").send\_keys(user1 + Keys.ENTER)

# driver.find\_element\_by\_id("pass-real").clear() # sometimes you need this  
driver.find\_element\_by\_id("pass-real").send\_keys(pswd1 + Keys.ENTER)

5) find element by tag\_name vs class\_name

<div class="bar">

<li>one</li>

<li>two</li>

</div>

parentElement = driver.find\_element\_by\_class\_name("bar")

elementList = parentElement.find\_elements\_by\_tag\_name("li")

6) Hoover mouse

basepixitmediaurl = driver.current\_url  
print('URL = ' + str(basepixitmediaurl))  
elements\_pixstor = driver.find\_elements\_by\_xpath("//\*[contains(text(),'PixStor')]")  
element\_pixstor = elements\_pixstor[0] # if more than one elements  
hoover(driver).move\_to\_element(element\_pixstor).perform()  
features\_el = driver.find\_element\_by\_xpath("//\*[contains(text(),'Features')]")  
hoover(driver).move\_to\_element(features\_el).perform()  
powersearch\_el = driver.find\_element\_by\_xpath("//\*[contains(text(),'Powerful Search')]")  
hoover(driver).move\_to\_element(powersearch\_el).perform()  
powersearch\_el.click()

7) Scrol down display (GUI browser display)

def displayheight(self):  
 *# Page scroll down  
 # driver.find\_element\_by\_xpath("//\*[contains(text(),'Contact Us')]").send\_keys(Keys.PAGE\_DOWN)  
 # driver.execute\_script("window.scrollTo(0, document.body.scrollHeight);")  
 # driver.execute\_script("window.scrollTo(0, Y)")*

driver = self.driver  
 try:  
 last\_height = driver.execute\_script("return document.body.scrollHeight")  
 except:  
 last\_height = 1800  
 if last\_height <= 1800:  
 last\_height = 1850  
 half\_height = int(0.5 \* last\_height)  
 oneten\_height = int(0.1 \* last\_height)  
 *# print('HEIGHT = ' + str(last\_height) + ' / 0.5HEIGHT = ' + str(half\_height) + ' / 0.1HEIGHT = ' + str(oneten\_height))* return last\_height, half\_height, oneten\_height

driver.execute\_script("window.scrollTo(0, " + str(self.displayheight()[1]) + ");") # scroll half display

8) pop-up display : to switch to pop-up display, click button, then go back to default display

*# new features by GOOGLE incognito mode - pop-up of Agree button which I need to click*if driver.find\_element\_by\_css\_selector("iframe"):  
 iframe1 = driver.find\_elements\_by\_css\_selector("iframe")  
 driver.switch\_to.frame(iframe1[0])  
 *# driver.find\_element\_by\_xpath("//\*[contains(text(),'I agree')]").click()* driver.find\_element\_by\_id("introAgreeButton").click()  
 driver.switch\_to.default\_content()

9) Get URL from text link (use ‘get\_attribute’ from element)

pdffileurl = driver.find\_element\_by\_xpath("//\*[contains(text(),'DOWNLOAD')]").get\_attribute('href')  
print('PDF FILE URL = ' + str(pdffileurl))

10) Combine 2 xpath

<fieldset class="module aligned ">

<h2>Personal info</h2>

<div class="form-row field-first\_name">

<div>

<label>First name:</label>

<div class="readonly">Iqraq</div>

</div>

</div>

<div class="form-row field-last\_name">

<div>

<label>Last name:</label>

<div class="readonly">Salasa</div>

</div>

</div>

<div class="form-row field-email">

<div>

<label>Email address:</label>

<div class="readonly">akucromox@gmail.com</div>

</div>

</div>

</fieldset>

So, For email XPATH :

email1 = driver.find\_element\_by\_xpath('//\*[@class="form-row field-email"]//\*[@class="readonly"]').text

11) if normal click() not working, we can use JavaScript click

element = driver.find\_element\_by\_xpath("(//a[contains(@href, '')])[20]")

driver.execute\_script("arguments[0].click();", element)

if you has more than 1 elements, e.g. element1 & element2 :

driver.execute\_script("arguments[0].click();arguments[1].click()", element1, element2)

12) find ID which contain some word (you didn’t know the full ID word)

# CSS selector  
# valuetofind = 'input[id\*="uniqName\_"]'  
# list\_ids = driver.find\_elements\_by\_css\_selector(valuetofind)

# XPATH  
valuetofind = '//\*[contains(@id, "uniqName\_")]'  
list\_ids = driver.find\_elements\_by\_xpath(valuetofind)

13) Width & Height browser's display window  
  
lebar = driver.execute\_script("return window.innerWidth")  
print('LEBAR = ', lebar)  
tinggi = driver.execute\_script("return window.innerHeight")  
print('TINGGI = ', tinggi)  
  
14) XPATH - using AND and OR (example only)  
  
<div tabindex="-1" class="chart-container" style="width: 1015px; height: 694px;">  
 <div class="chartLayer" style="width: 1015px; height: 560px;">  
 <div class="clip-path" style="width: 955px; height: 561px; margin-top: 0px; margin-left: 0px;">  
................  
<div tabindex="-1" class="chart-container draggable" style="width: 1015px; height: 694px;">  
 <div class="chartLayer" style="width: 1015px; height: 560px;">  
 <div class="clip-path" style="width: 955px; height: 561px; margin-top: 0px; margin-left: 0px;">  
................  
  
xp\_baca = '//\*[((@class="chart-container") or (@class="chart-container draggable")) and (@tabindex="-1")]'  
driver.find\_elements\_by\_xpath(xp\_baca)  
  
15) list - to multi slicing  
  
# slice the list, then add it together – e.g. to list the first 3 and last 2 elements  
  
test2 = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l']  
print(test2[0:3] + test2[-2:])

16) reverse LIST - use ‘[::-1]’

test3 = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l']  
print(test3) # original list

print(test3[::-1]) # reverse list

17) UI WbDriver/Selenium wait till the element (toolTip element) appear

from selenium.webdriver.common.by import By  
from selenium.webdriver.support.wait import WebDriverWait  
from selenium.webdriver.support import expected\_conditions as EC  
from selenium.webdriver.common.action\_chains import ActionChains as hoover

xpath1 = '//\*[@class="chart-tooltip"]'  
toolTip = WebDriverWait(driver, 10).until(EC.presence\_of\_element\_located((By.XPATH, xpath1)))  
  
hoover(driver).move\_to\_element(toolTip).perform()  
print('ELE LOCATION ', toolTip.location)  
print('TEXT = ', toolTip.text)

18)