Selenium Lab

Preconditions

- Feel free to install **selenium** with Java or Python
- Install Firefox/Chrome (Chrome is recommended) and their **corresponding webdriver**

Simple Example

```
from selenium import webdriver
# Open the web browser and navigate to the login page
driver = webdriver.Chrome()
driver.get("https://example.com/login")
# Enter a valid username and password
username_input = driver.find_element_by_id("username")
password_input = driver.find_element_by_id("password")
username_input.send_keys("example_user")
password_input.send_keys("example_password")
# Click the login button
login_button = driver.find_element_by_id("login-button")
login_button.click()
# Verify that the user is redirected to the home page
assert driver.current_url == "https://example.com/home"
# Verify that the user's username is displayed on the home page
welcome_message = driver.find_element_by_id("welcome-message")
assert welcome_message.text == "Welcome, example_user!"
# Close the web browser
driver.quit()
```

Tasks

Task 1: Verify Google/Baidu search results page

1. Open Chrome browser and navigate to https://www.google.com/ or https://www.baidu.com/
2. Enter "selenium" in the search box and press Enter.
3. Verify that the search results page displays at least 10 search results.
4. Verify that the first search result link is from the Selenium official website (https://www.selenium.dev/).
5. Verify that the search box text contains "selenium".

Task 2: Test a login form

1. Open Chrome browser and navigate to a website with a login form (e.g. https://the-internet.herokuapp.com/login).
2. Enter a valid username and password.
3. Click the login button.
4. Verify that the user is redirected to the home page.

5. Verify that the user's username is displayed on the home page.
6. Logout and verify that the user is redirected to the login page.
7. Enter a invalid username and password and check what's gonna happen

Task 3: Test form validation

1. Open Chrome browser and navigate to a website with a registration form (e.g.https://demoqa.com/automation-practice-form).
2. Enter invalid data into the form fields (e.g. a non-email address in the email field, a phone number with letters in the phone number field).
3. Submit the form and verify that appropriate error messages are displayed for each invalid field.
4. Enter valid data into the form fields.
5. Submit the form and verify that the user is redirected to a success page.

Task 4: Test website navigation

1. Open Chrome browser and navigate to a website with a navigation menu (e.g. https://www.udemy.com/).
2. Click on each link in the navigation menu and verify that the corresponding page loads.
3. Click on the website logo or home link and verify that the user is redirected to the home page.
4. Use the back and forward buttons in the browser to navigate between pages and verify that the correct pages are loaded.

Task 5: Test dynamic content

1. Open Chrome browser and navigate to a website with dynamic content (e.g. https://the-internet.herokuapp.com/dynamic_content).
2. Verify that the page displays three images.
3. Click the "Shuffle" button to change the images.
4. Verify that the images have been shuffled and that there are still three images displayed.

Requirements for the experiment:

1. Finish the tasks above individually.
2. Check in your java code to gitee
3. Please send your experiment report to 智慧树, the following information should be included in your report:

a) The brief description that you install Selenium, Firefox/Chrome and SeleniumIDE.

b) Steps for recording and exporting scripts and your gitee repo link.

c) Steps for testing the website using Selenium

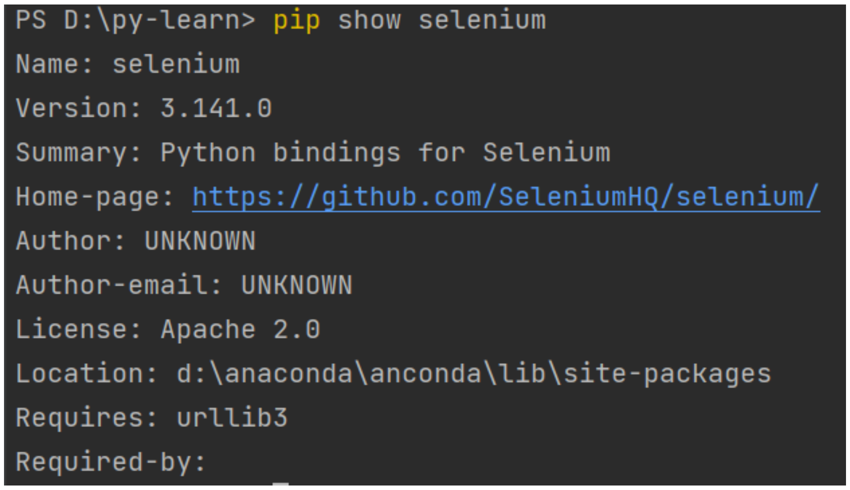
IMPORTANT： Your report should be named **'学号_姓名'.pdf (e.g., '3020123123_张三.pdf')**, otherwise we **could miss** your report.

Submission deadline: 23:59 April 12, 2023.

1.install selenium with Python

```
pip install selenium
```

执行上述命令，成功安装后如下图所示。

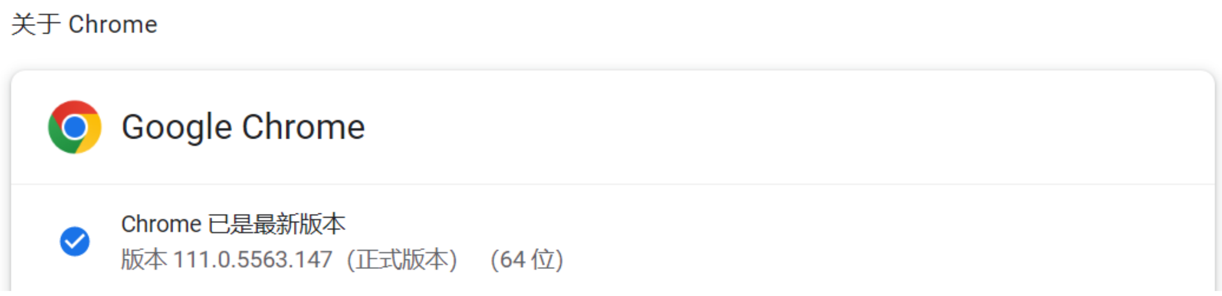


```
PS D:\py-learn> pip show selenium
Name: selenium
Version: 3.141.0
Summary: Python bindings for Selenium
Home-page: https://github.com/SeleniumHQ/selenium/
Author: UNKNOWN
Author-email: UNKNOWN
License: Apache 2.0
Location: d:\anaconda\anconda\lib\site-packages
Requires: urllib3
Required-by:
```

2.Install Firefox/Chrome and their corresponding webdriver

Chrome浏览器在实验前已经安装，此处不再赘述。

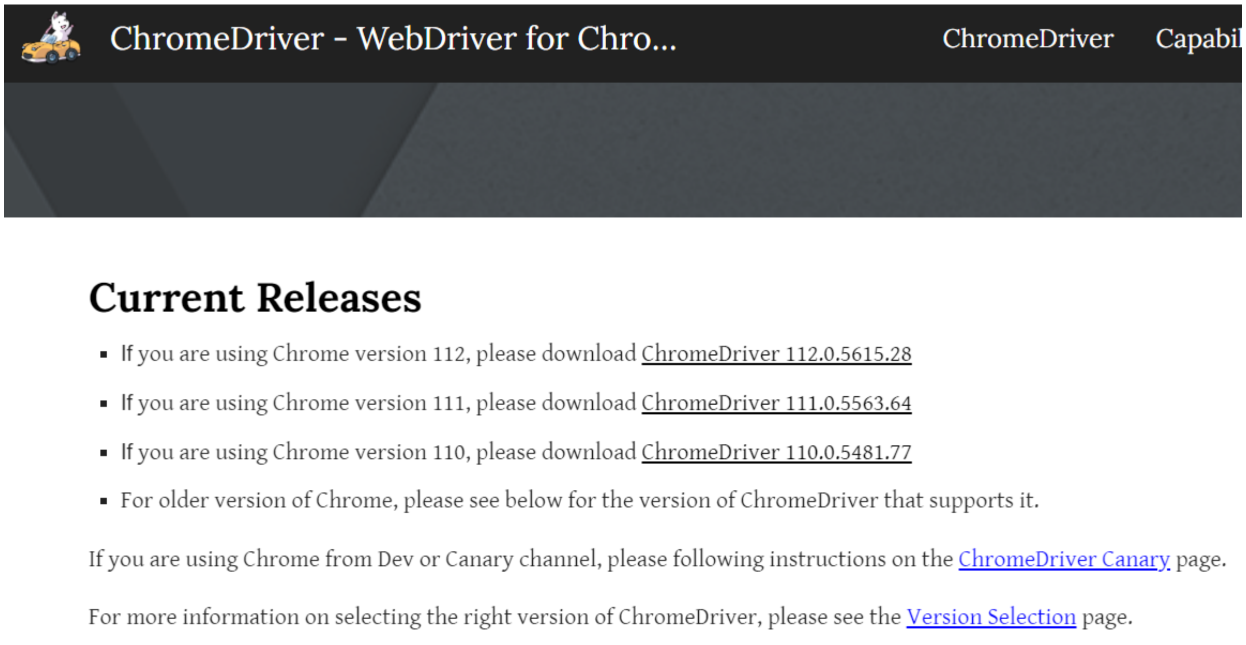
检查Chrome版本，并且前往链接https://chromedriver.chromium.org/downloads下载驱动，这里我们要确保下载的版本与Chrome一致。下载完成后将chromedriver.exe移动到python环境目录下。



关于 Chrome

Google Chrome

Chrome 已是最新版本
版本 111.0.5563.147（正式版本） （64 位）


ChromeDriver - WebDriver for Chro... ChromeDriver Capabil

Current Releases

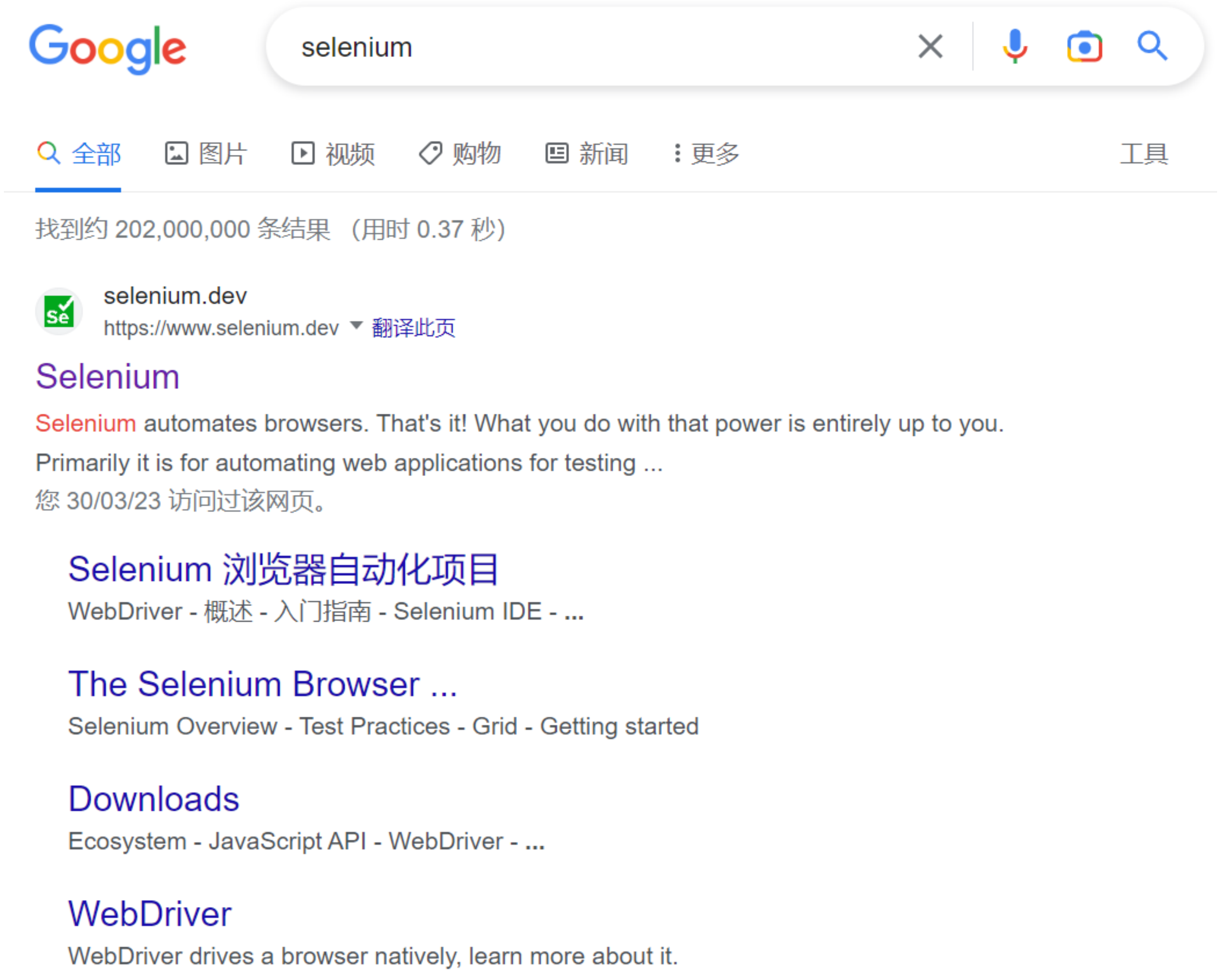
- If you are using Chrome version 112, please download ChromeDriver 112.0.5615.28
- If you are using Chrome version 111, please download ChromeDriver 111.0.5563.64
- If you are using Chrome version 110, please download ChromeDriver 110.0.5481.77
- For older version of Chrome, please see below for the version of ChromeDriver that supports it.

If you are using Chrome from Dev or Canary channel, please following instructions on the ChromeDriver Canary page.

For more information on selecting the right version of ChromeDriver, please see the Version Selection page.

3 Tasks

3.1 Task 1: Verify Google/Baidu search results page



找到约 202,000,000 条结果 （用时 0.37 秒）

selenium.dev
https://www.selenium.dev ▾ 翻译此页

Selenium

Selenium automates browsers. That's it! What you do with that power is entirely up to you. Primarily it is for automating web applications for testing ...
您 30/03/23 访问过该网页。

Selenium 浏览器自动化项目
WebDriver - 概述 - 入门指南 - Selenium IDE - ...

The Selenium Browser ...
Selenium Overview - Test Practices - Grid - Getting started

Downloads
Ecosystem - JavaScript API - WebDriver - ...

WebDriver
WebDriver drives a browser natively, learn more about it.

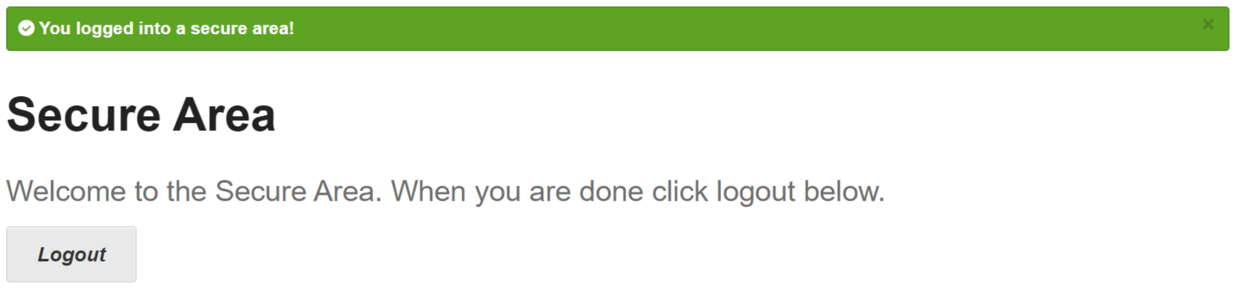
搜索结果页面显示10个以上的搜索结果。

第一个搜索结果链接来自Selenium官方网站。

搜索框文本包含“selenium”。

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
import time

# 1. 打开当地浏览器
driver = webdriver.Chrome()

# 1. Open Chrome browser and navigate to https://www.google.com/
driver.get("https://www.google.com/")

# 2. Enter "selenium" in the search box and press Enter.
search_box = driver.find_element_by_name("q")
search_box.send_keys("selenium")
search_box.send_keys(Keys.RETURN)

# Wait for the search results to load
time.sleep(3)

# 3. Verify that the search results page displays at least 10 search results.
search_results = driver.find_elements_by_css_selector(".g")
assert len(search_results) >= 10, "Less than 10 search results"

# 4. Verify that the first search result link is from the Selenium official website
(https://www.selenium.dev/).
first_result_link = search_results[0].find_element_by_css_selector("a")
assert "https://www.selenium.dev/" in first_result_link.get_attribute(
    "href"), "First result link is not from the Selenium official website"

# 5. Verify that the search box text contains "selenium".
search_box = driver.find_element_by_name("q")
assert search_box.get_attribute("value") == "selenium", "Search box text does not
contain 'selenium'"

# Close the browser
driver.quit()
```

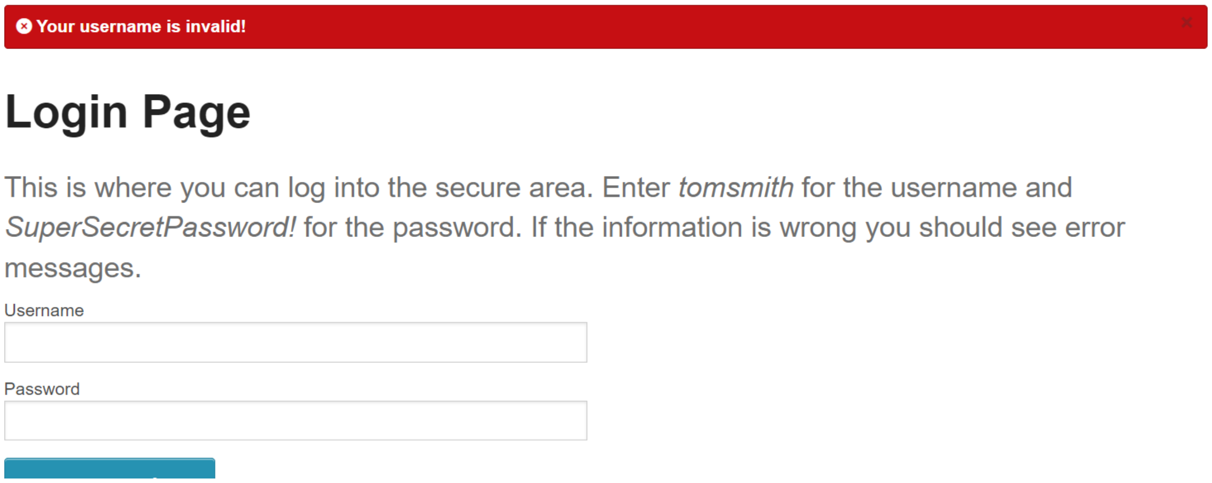
3.2 Task2 Test a login form


You logged into a secure area!

Secure Area

Welcome to the Secure Area. When you are done click logout below.

用户的用户名没有显示在主页上。


Your username is invalid!

Login Page

This is where you can log into the secure area. Enter *tomsmith* for the username and *SuperSecretPassword!* for the password. If the information is wrong you should see error messages.

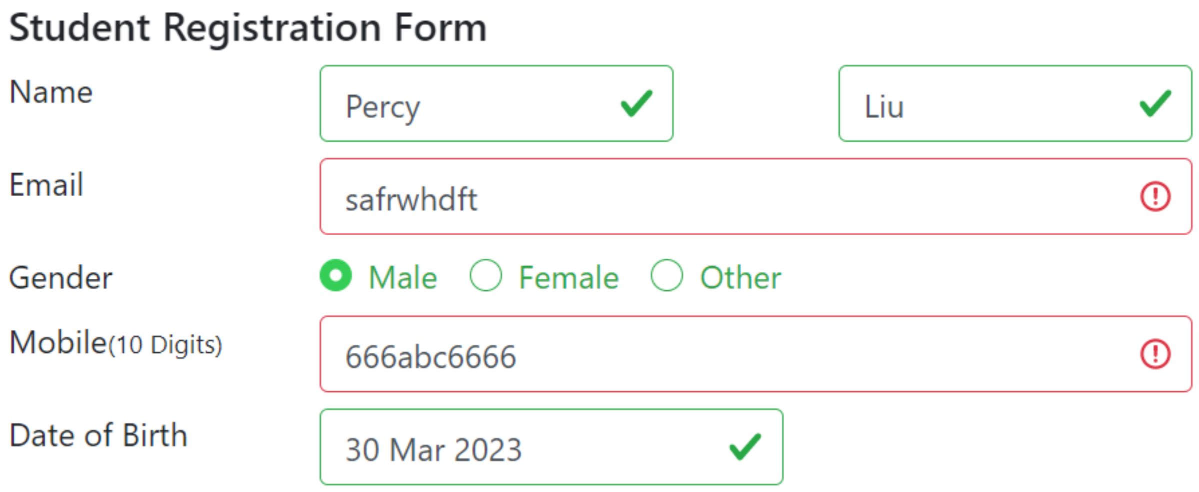
输入无效的用户名和密码后顶部将显示一个警告标签"Your username is invalid"。

```python
import time
from selenium import webdriver

# 1. Open Chrome browser and navigate to a website with a login form
driver = webdriver.Chrome()
driver.get("https://the-internet.herokuapp.com/login")

# 2. Enter a valid username and password
username = driver.find_element_by_id("username")
password = driver.find_element_by_id("password")
username.send_keys("tomsmith")
password.send_keys("SuperSecretPassword!")

# 3. Click the login button
login_button = driver.find_element_by_css_selector(".fa-sign-in")
login_button.click()

# Wait for the home page to load
time.sleep(2)

# 4. Verify that the user is redirected to the home page
assert "https://the-internet.herokuapp.com/secure" in driver.current_url, "Not
redirected to the home page"

# 5. Verify that the user's username is displayed on the home page
username_display = driver.find_element_by_id("flash-messages")
assert "tomsmith" not in username_display.text, "Username displayed on the home
page"

# 6. Logout and verify that the user is redirected to the login page
logout_button = driver.find_element_by_css_selector(".icon-2x.icon-signout")
logout_button.click()

# Wait for the login page to load
time.sleep(2)
```

```
assert "https://the-internet.herokuapp.com/login" in driver.current_url, "Not
redirected to the login page"

# 7. Enter an invalid username and password and check what's gonna happen
username = driver.find_element_by_id("username")
password = driver.find_element_by_id("password")
username.send_keys("invalid_username")
password.send_keys("invalid_password")

# Click the login button
login_button = driver.find_element_by_css_selector(".fa-sign-in")
login_button.click()

# Wait for the error message to load
time.sleep(2)

# Check if an error message is displayed
error_message = driver.find_element_by_id("flash-messages")
assert "Your username is invalid!" in error_message.text, "Error message not
displayed for invalid credentials"

# Close the browser
driver.quit()
```

3.3 Task 3: Test form validation

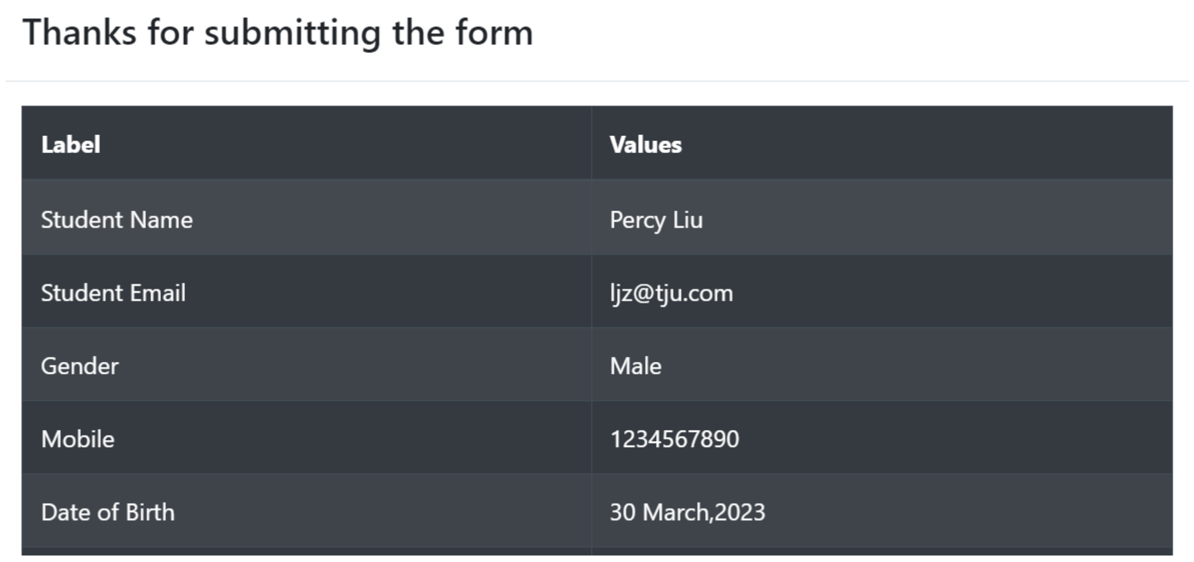
向表单字段中输入无效数据后



Student Registration Form

Name	Percy ✓	Liu ✓
Email	safrwhdft ⓘ	
Gender	● Male ○ Female ○ Other	
Mobile(10 Digits)	666abc6666 ⓘ	
Date of Birth	30 Mar 2023 ✓	

在表单字段中输入有效数据。提交表单后，会有一个完整的表单弹出。



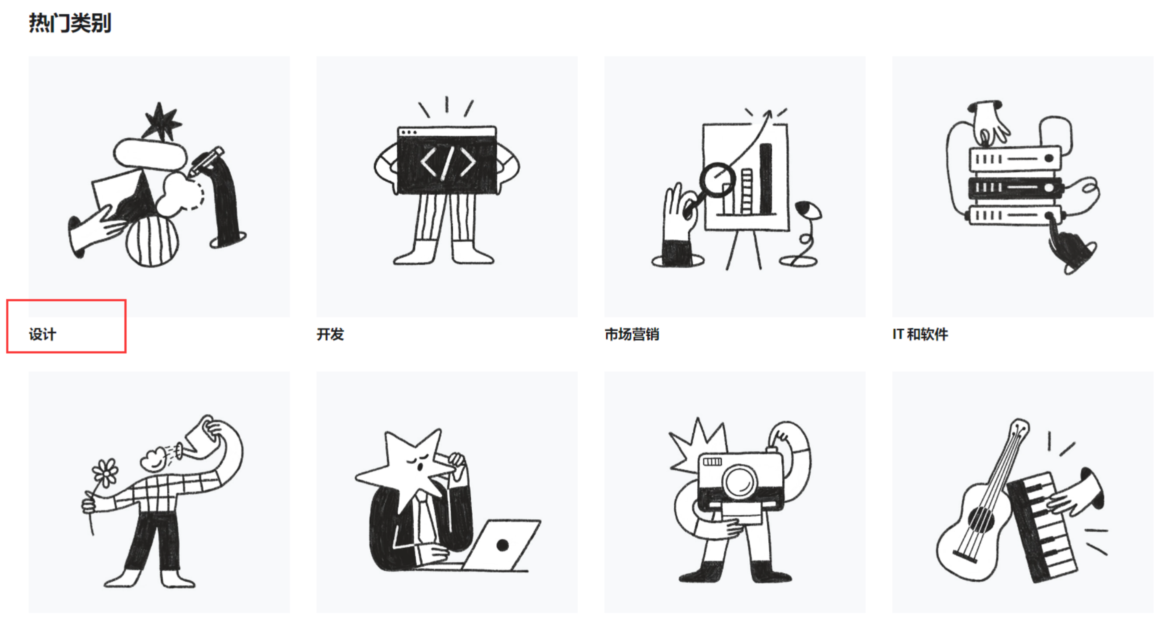
Thanks for submitting the form

Label	Values
Student Name	Percy Liu
Student Email	ljz@tju.com
Gender	Male
Mobile	1234567890
Date of Birth	30 March,2023

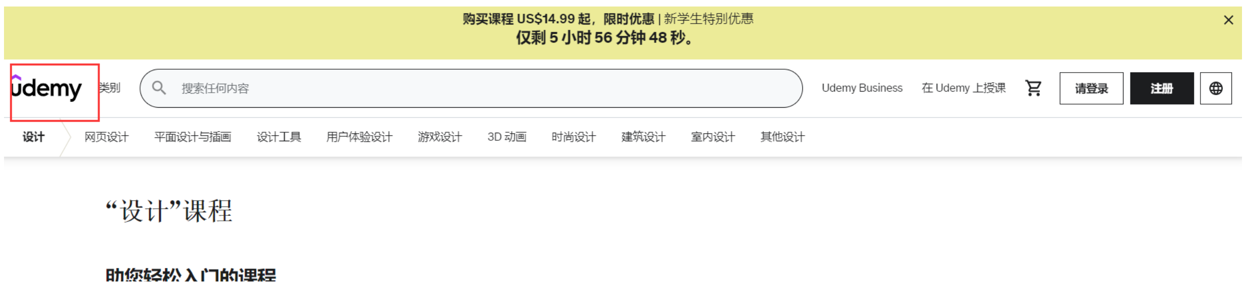
```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
import time
from selenium.webdriver.common.by import By
from selenium.webdriver.support import expected_conditions as EC

# 1. Open Chrome browser and navigate to a website with a registration form
from selenium.webdriver.support.wait import WebDriverWait

driver = webdriver.Chrome()
driver.get("https://demoqa.com/automation-practice-form")

# 2. Enter invalid data into the form fields
first_name = driver.find_element_by_id("firstName")
last_name = driver.find_element_by_id("lastName")
email = driver.find_element_by_id("userEmail")
mobile = driver.find_element_by_id("userNumber")

first_name.send_keys("Percy")
last_name.send_keys("Liu")
email.send_keys("safrwhdft")
mobile.send_keys("666abc6666")
# Select gender
wait = WebDriverWait(driver, 20)
gender_male_label = wait.until(EC.element_to_be_clickable((By.XPATH,
"//label[@for='gender-radio-1']")))
gender_male_label.click()

# 3. Submit the form and verify that appropriate error messages are displayed for
each invalid field
submit_button = driver.find_element_by_xpath("//button[@id='submit']")
submit_button.send_keys(Keys.RETURN)

# Wait for the error messages to load
time.sleep(2)
```

```
# 4. Enter valid data into the form fields
email.clear()
mobile.clear()

email.send_keys("ljz@tju.com")
mobile.send_keys("1234567890")
submit_button.send_keys(Keys.RETURN)

# 5. Verify that the user is redirected to a success page
wait = WebDriverWait(driver, 20)
success_message = wait.until(EC.visibility_of_element_located((By.CSS_SELECTOR,
".modal-title.h4")))
assert "Thanks for submitting the form" in success_message.text, "Success message
not displayed"

# Close the browser
time.sleep(3)
driver.quit()
```

3.4 Task 4: Test website navigation



点击导航中文字进行跳转。



购买课程 US$14.99 起，限时优惠 | 新学生特别优惠
仅剩 5 小时 56 分钟 48 秒。

udemy 类别 搜索任何内容 Udemy Business 在 Udemy 上授课 请登录 注册

设计 网页设计 平面设计与插画 设计工具 用户体验设计 游戏设计 3D 动画 时尚设计 建筑设计 室内设计 其他设计

“设计”课程

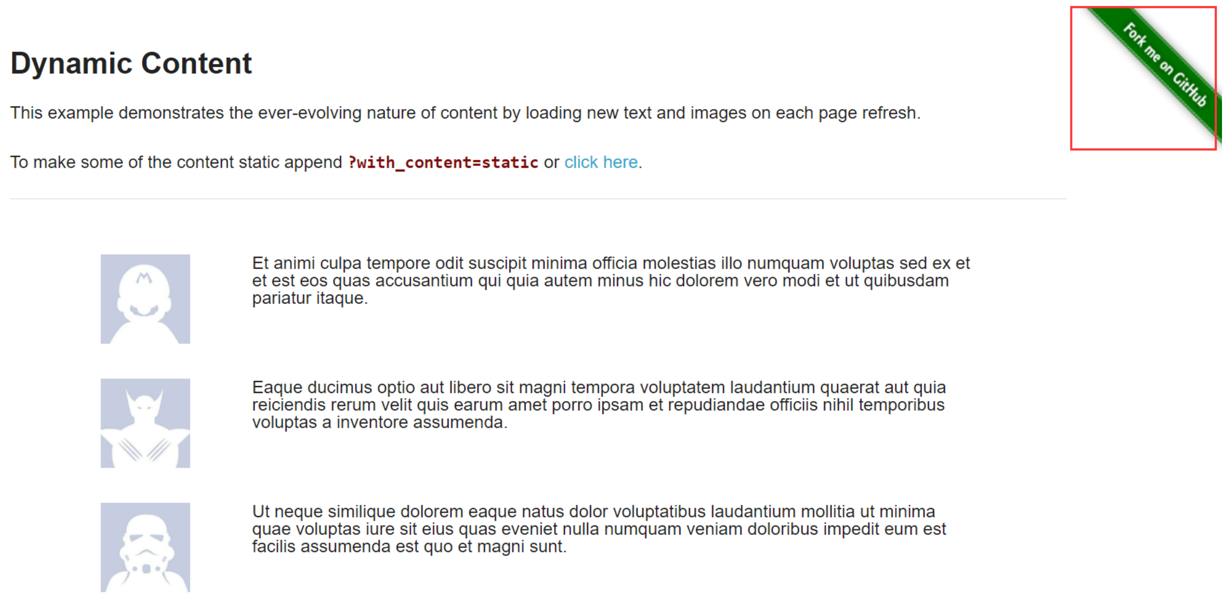
助你轻松入门的课程

点击图标回到首页。

```
from selenium import webdriver
```

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

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

# Wait for the navigation menu to load
wait = WebDriverWait(driver, 20)
link = wait.until(EC.element_to_be_clickable((By.LINK_TEXT, "设计")))
link.click()

# verify the url
assert "https://www.udemy.com/zh-cn/courses/design/" in driver.current_url, "URL is not correct"
link = wait.until(EC.element_to_be_clickable((By.CSS_SELECTOR, "a.ud-btn.ud-btn-large.ud-btn-link.ud-heading-md")))

# Click the link
link.click()

# 返回上一页并检查
driver.back()
assert "https://www.udemy.com/zh-cn/courses/design/" in driver.current_url, "URL is not correct"

# 前进一页并检查
driver.forward()
assert "https://www.udemy.com/zh-cn/courses/design/" in driver.current_url, "URL is not correct"

# Close the browser
driver.quit()
```

3.5 Task 5: Test dynamic content


Dynamic Content

This example demonstrates the ever-evolving nature of content by loading new text and images on each page refresh.

To make some of the content static append `?with_content=static` or click here.

Et animi culpa tempore odit suscipit minima officia molestias illo numquam voluptas sed ex et et est eos quas accusantium qui quia autem minus hic dolorem vero modi et ut quibusdam pariatur itaque.

Eaque ducimus optio aut libero sit magni tempora voluptatem laudantium quaerat aut quia reiciendis rerum velit quis earum amet porro ipsam et repudiandae officiis nihil temporibus voluptas a inventore assumenda.

Ut neque similique dolorem eaque natus dolor voluptatibus laudantium mollitia ut minima quae voluptas iure sit eius quas eveniet nulla numquam veniam doloribus impedit eum est facilis assumenda est quo et magni sunt.

需要注意整个页面其实有4张图片，页面右上角的“Fork me on Github”其实也是一张图片，但根据题意，我们应该关注的是主页面的三张图片。

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

driver = webdriver.Chrome()
driver.get("https://the-internet.herokuapp.com/dynamic_content")

# Step 2: Verify that the page displays three images
wait = WebDriverWait(driver, 10)
images = wait.until(EC.presence_of_all_elements_located((By.CSS_SELECTOR, "#content 
img")))
assert len(images) == 3, "There should be 3 images on the page"
print("There are 3 images on the page")
# Get the src attributes of the images before shuffle
src_before_shuffle = [img.get_attribute("src") for img in images]

# Step 3: Click the "Shuffle" button to change the images
shuffle_button = wait.until(
    EC.element_to_be_clickable((By.CSS_SELECTOR, "a[href='/dynamic_content?
with_content=static']")))
shuffle_button.click()

# Step 4: Verify that the images have been shuffled and that there are still three 
images displayed
images_after_shuffle = 
wait.until(EC.presence_of_all_elements_located((By.CSS_SELECTOR, "#content img")))
assert len(images_after_shuffle) == 3, "There should still be 3 images on the page 
after shuffle"

# Get the src attributes of the images after shuffle
```

```
src_after_shuffle = [img.get_attribute("src") for img in images_after_shuffle]

# Check if the images have been shuffled
assert src_before_shuffle != src_after_shuffle, "The images have not been shuffled"
print(src_before_shuffle)
print(src_after_shuffle)
# Close the browser
driver.quit()
```

控制台输出结果如下，表明图片在点击后内容发生了变化:

```
There are 3 images on the page
['https://the-internet.herokuapp.com/img/avatars/Original-Facebook-Geek-Profile-
Avatar-2.jpg', 'https://the-internet.herokuapp.com/img/avatars/Original-Facebook-
Geek-Profile-Avatar-2.jpg', 'https://the-
internet.herokuapp.com/img/avatars/Original-Facebook-Geek-Profile-Avatar-1.jpg']
['https://the-internet.herokuapp.com/img/avatars/Original-Facebook-Geek-Profile-
Avatar-1.jpg', 'https://the-internet.herokuapp.com/img/avatars/Original-Facebook-
Geek-Profile-Avatar-2.jpg', 'https://the-
internet.herokuapp.com/img/avatars/Original-Facebook-Geek-Profile-Avatar-6.jpg']
```