
Pop Up

What is pop-up?

- A pop-up is a graphical user interface (GUI) display area, usually a small window, which appears in the foreground of the visual interface. Pop-ups can be initiated by a single or double mouse click or rollover (sometimes called a mouse-over).
- A window that appears (pops up) when you select an option with a mouse or press a special function key/link/button.
- A pop-up window must be smaller than the background window or interface; otherwise, it's a replacement interface.

Pop ups are categorized into following types

➔ **HTML Pop up**

1. Invisible pop ups
 - Hidden division pop up.
 - Calendar pop up.
2. Child-browser pop up or HTML pop up

➔ **Non-HTML pop up**

1. Alert/Confirmation/Java Script pop up
2. Notification pop up
3. File upload pop up
4. File download pop up
5. Window pop up

HTML pop up

- Pop up which is built using the HTML is called as HTML pop ups.

1. Hidden division pop up

- The characteristics Hidden division pop-up are
 1. We can inspect it.
 2. We cannot move it.
 3. It will be colorful.

How to handle the hidden division pop-up?

- In order to handle any pop-up, first we need to understand the nature and characteristics of the pop-ups.

- Hidden division pop-ups are created using div (division) html tag, which contains a property called display with the initial value 'none [hidden]' whenever it is required developer, will change it to 'block [Display]'.
- Since it is part of web page itself, we never use 'switch to' in order to handle hidden division pop-up we use 'find element ()' method only.

Write a script to handle the hidden division pop up of ActiTIME application?

```
import string
import random
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
from webdriver_manager.chrome import ChromeDriverManager

task_name = "AAA - Test_task"
customer_name = ''.join(random.choices(string.ascii_uppercase + string.digits, k=6))
project_name = "py-spiders"

driver = webdriver.Chrome(executable_path=ChromeDriverManager().install())
wait = WebDriverWait(driver, 15)
driver.implicitly_wait(30)
driver.maximize_window()
driver.get("https://demo.actitime.com/login.do")

driver.find_element_by_id("username").send_keys("admin")
driver.find_element_by_name("pwd").send_keys("manager")
driver.find_element_by_id("loginButton").click()
driver.find_element_by_xpath("//div[text()='Tasks']/..").click()

add_new_button = wait.until(ec.element_to_be_clickable((By.XPATH, "//div[text()='Add New']")))
driver.execute_script("arguments[0].click()", add_new_button)
new_task_button = driver.find_element(By.XPATH, "//div[contains(text(),'New Tasks')]")
new_task_button.click()

customer_combo_box = wait.until(ec.element_to_be_clickable((By.CSS_SELECTOR,
    "div[class^='customerSelector ']' div[class='comboboxButton']>div.selectedItem"))))
driver.execute_script("arguments[0].click()", customer_combo_box)
new_customer_button = driver.find_element(By.XPATH,
    "//div[text()='~ New Customer ~'] and starts-with(@class, 'itemRow')")
driver.execute_script("arguments[0].click()", new_customer_button)
customer_name_text_box = driver.find_element(By.CSS_SELECTOR,
    "tr.selectCustomerRow input[placeholder='Enter Customer Name']")
customer_name_text_box.send_keys(customer_name)
project_name_text_box = driver.find_element(By.CSS_SELECTOR,
    "tr.selectProjectRow input[placeholder='Enter Project Name']")
project_name_text_box.send_keys(project_name)

task_name_text_box = driver.find_element(By.XPATH,
    "//table[@class='createTasksTable']/tbody/tr[1]//input[@placeholder='Enter task name']")
task_name_text_box.send_keys(task_name)
create_task_button = driver.find_element(By.XPATH, "//div[normalize-space(text())='Create Tasks']")
create_task_button.click()
```

```

success_message_after_creating_task = driver.find_element(By.CSS_SELECTOR,
                                                         "div.toastsContainer span.innerHTML")

if success_message_after_creating_task.is_displayed():
    print("Created the task: ", task_name)
else:
    assert False, "Unable to create the task: " + task_name
pass

driver.quit()

```

Note: If you run the above script, it will create the “new task” in actiTIME application, but if you run script again, script will fail. Because in ActiTIME, it doesn’t allow you to create the duplicate task name.

So, write a script which checks the task name first, if task name is already exists, delete the task and create the new task. If task name is not present, create the new task.

```

import string
import random
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
from webdriver_manager.chrome import ChromeDriverManager

task_name = "AAA - Test_task"
customer_name = ''.join(random.choices(string.ascii_uppercase + string.digits, k=6))
project_name = "py-spiders"

driver = webdriver.Chrome(executable_path=ChromeDriverManager().install())
wait = WebDriverWait(driver, 15)
driver.implicitly_wait(30)
driver.maximize_window()
driver.get("https://demo.actitime.com/login.do")

driver.find_element_by_id("username").send_keys("admin")
driver.find_element_by_name("pwd").send_keys("manager")
driver.find_element_by_id("loginButton").click()
driver.find_element_by_xpath("//div[text()='Tasks']/..").click()

```

```

''' Before creating the task, checking if the task is present with the same name.
    If any task already present, than deleting that task and creating a new task
    # Below code is to check for the task'''

all_tasks = driver.find_elements(By.CSS_SELECTOR, "table.taskRowsTable div.title")

for task in all_tasks:
    driver.execute_script("arguments[0].scrollIntoView();", task)
    if task.text == task_name:
        task_checkbox_xpath = "//*[div[@title='" + task.text + "']/ancestor::tr//div[@class='checkbox inactive']"
        task_checkbox = driver.find_element(By.XPATH, task_checkbox_xpath)
        task_checkbox.click()
        '''Since the delete button present on the hidden division pop up, using explicit wait to
        wait for web element to become visible and active'''
        delete_button = wait.until(ec.element_to_be_clickable((By.CSS_SELECTOR,
                                                                "div.bulkOperationsPanel>div.delete.button")))
        delete_button.click()
        delete_permanently_button = wait.until(ec.element_to_be_clickable((By.XPATH,
                                                                "//*[text()='Delete with users' time entries' or text()='Delete permanently']")))
        delete_permanently_button.click()
        success_message_after_deleting = driver.find_element(By.CSS_SELECTOR, "div.toastsContainer span.innerHTML")

        if success_message_after_deleting.is_displayed():
            print("Deleted the task: ", task_name)
        else:
            assert False, "Unable to delete the task: " + task_name
        pass
        break

''' Below code is for Creating the task'''
add_new_button = wait.until(ec.element_to_be_clickable((By.XPATH, "//*[text()='Add New']")))
driver.execute_script("arguments[0].click()", add_new_button)
new_task_button = driver.find_element(By.XPATH, "//*[contains(text(),'New Tasks')]")
new_task_button.click()

customer_combo_box = wait.until(ec.element_to_be_clickable((By.CSS_SELECTOR,
                                                            "div[class='customerSelector ']' div[class='comboBoxButton']>div.selectedItem")))
driver.execute_script("arguments[0].click()", customer_combo_box)
new_customer_button = driver.find_element(By.XPATH, "//*[text()=' - New Customer -' and starts-with(@class, 'itemRow')]")
driver.execute_script("arguments[0].click()", new_customer_button)
customer_name_text_box = driver.find_element(By.CSS_SELECTOR,
                                              "tr.selectCustomerRow input[placeholder='Enter Customer Name']")
customer_name_text_box.send_keys(customer_name)
project_name_text_box = driver.find_element(By.CSS_SELECTOR,
                                              "tr.selectProjectRow input[placeholder='Enter Project Name']")
project_name_text_box.send_keys(project_name)

task_name_text_box = driver.find_element(By.XPATH,
                                          "//*[table[@class='createTasksTable']/tbody/tr[1]/input[@placeholder='Enter task name']]")
task_name_text_box.send_keys(task_name)
create_task_button = driver.find_element(By.XPATH, "//*[div[normalize-space(text()='Create Tasks')]")
create_task_button.click()

success_message_after_creating_task = driver.find_element(By.CSS_SELECTOR, "div.toastsContainer span.innerHTML")

if success_message_after_creating_task.is_displayed():
    print("Created the task: ", task_name)
else:
    assert False, "Unable to create the task: " + task_name
pass

driver.quit()

```

The above program, first checks for the task name. If it is available, it will delete and creates the new. If it is not available. It will create the new task.

2. Calendar pop up

Characteristics of Calendar pop up

1. We can inspect the elements
2. We can't move this pop up

Practice application for calendar pop up

<https://www.cleartrip.com/>

Write a program to print all the dates of a month

```
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.implicitly_wait(30)
driver.maximize_window()
driver.get("http://www.cleartrip.com/")

depart_on = driver.find_element_by_id("DepartDate")
depart_on.click()
all_dates = driver.find_elements_by_xpath("//span[text()='September']/../tbody/*[contains(@class, 'ui-state-default')]")

for date in all_dates:
    print(date.text)

driver.close()
```

Write a program to select the date

```
from selenium import webdriver

date = input("Enter the Date: ")

driver = webdriver.Chrome()
driver.maximize_window()
driver.implicitly_wait(100)
driver.get("http://www.cleartrip.com/")

depart_on = driver.find_element_by_id("DepartDate")
depart_on.click()
all_dates = driver.find_elements_by_xpath("//span[text()='September']/../tbody/*[contains(@class, 'ui-state-default')]")

count = 0

for count in range(count, len(all_dates)):
    if date == all_dates[count].text:
        all_dates[count].click()
        break

driver.close()
```

Write a program to get the dates of particular week

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

week_name = input("Enter the week name: ")
driver = webdriver.Chrome(executable_path=webdriver.ChromeDriverManager().install())
driver.implicitly_wait(30)
driver.maximize_window()
driver.get("http://www.cleartrip.com/")

depart_on = driver.find_element_by_id("DepartDate")
depart_on.click()
current_month = driver.find_element(By.CSS_SELECTOR, "div.monthBlock.first div.title")
current_month_text = current_month.text
```

```

weeks = driver.find_elements(By.XPATH,
    "//div[@class='monthBlock first']/table[@class='calendar']/thead//th/span")
count = 1
for week in weeks:
    if week.get_attribute("title").casefold() == week_name.casefold():
        break
    count += 1
count = str(count)
xpath = "//div[@class='monthBlock first']/table[@class='calendar']/tbody/tr/td[" + count + "]/*"
dates = driver.find_elements_by_xpath(xpath)
print(week_name + " comes " + str(len(dates)) + " times in " + current_month_text)

for date in dates:
    print(date.text + " " + current_month_text)

driver.quit()

```

Assignment:

Write a program to select the dates of any month in a calendar.