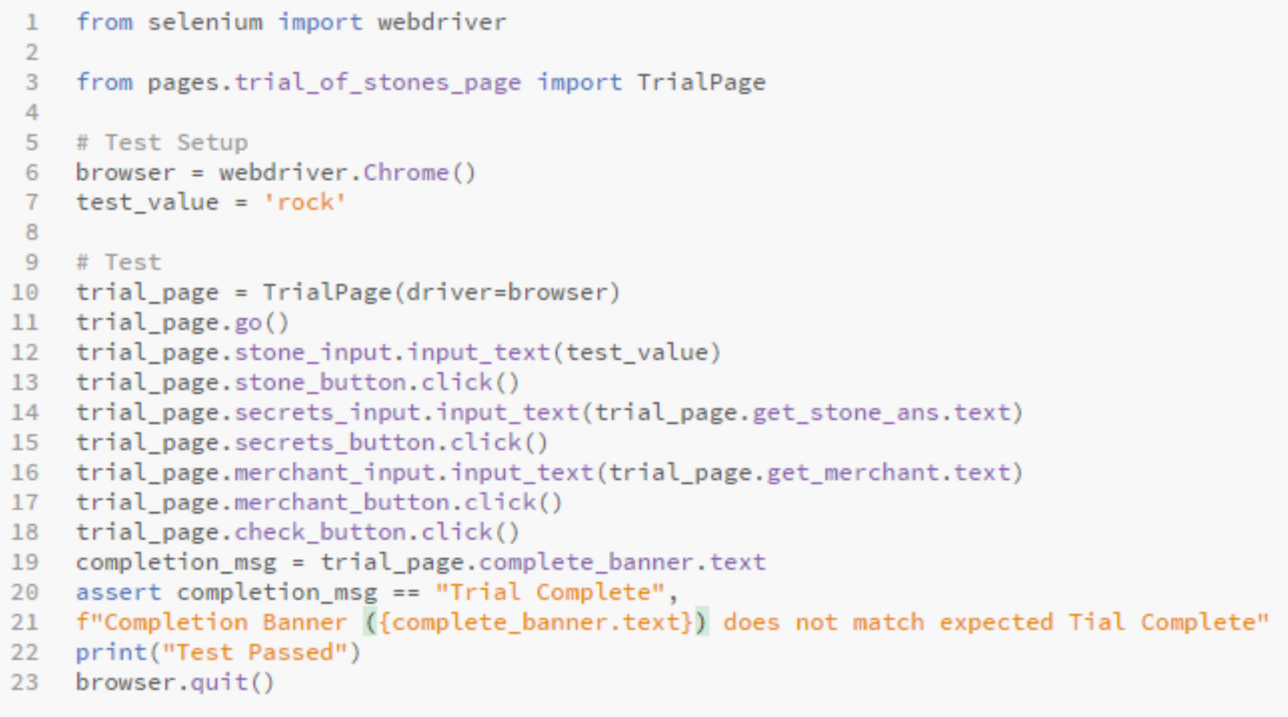
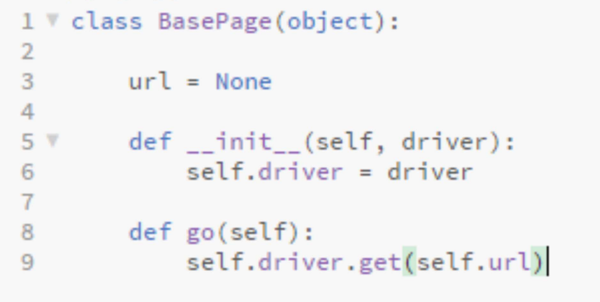
Code for project3_stone_challeneg_pom

pom_test_trial_page.py



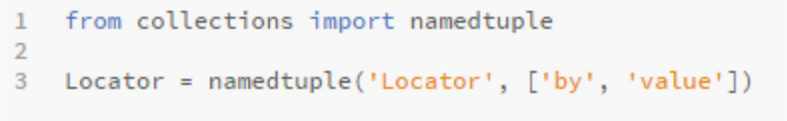
```python
from selenium import webdriver

from pages.trial_of_stones_page import TrialPage

# Test Setup
browser = webdriver.Chrome()
test_value = 'rock'

# Test
trial_page = TrialPage(driver=browser)
trial_page.go()
trial_page.stone_input.input_text(test_value)
trial_page.stone_button.click()
trial_page.secrets_input.input_text(trial_page.get_stone_ans.text)
trial_page.secrets_button.click()
trial_page.merchant_input.input_text(trial_page.get_merchant.text)
trial_page.merchant_button.click()
trial_page.check_button.click()
completion_msg = trial_page.complete_banner.text
assert completion_msg == "Trial Complete",
f"Completion Banner ({complete_banner.text}) does not match expected Tial Complete"
print("Test Passed")
browser.quit()
```

base_page.py



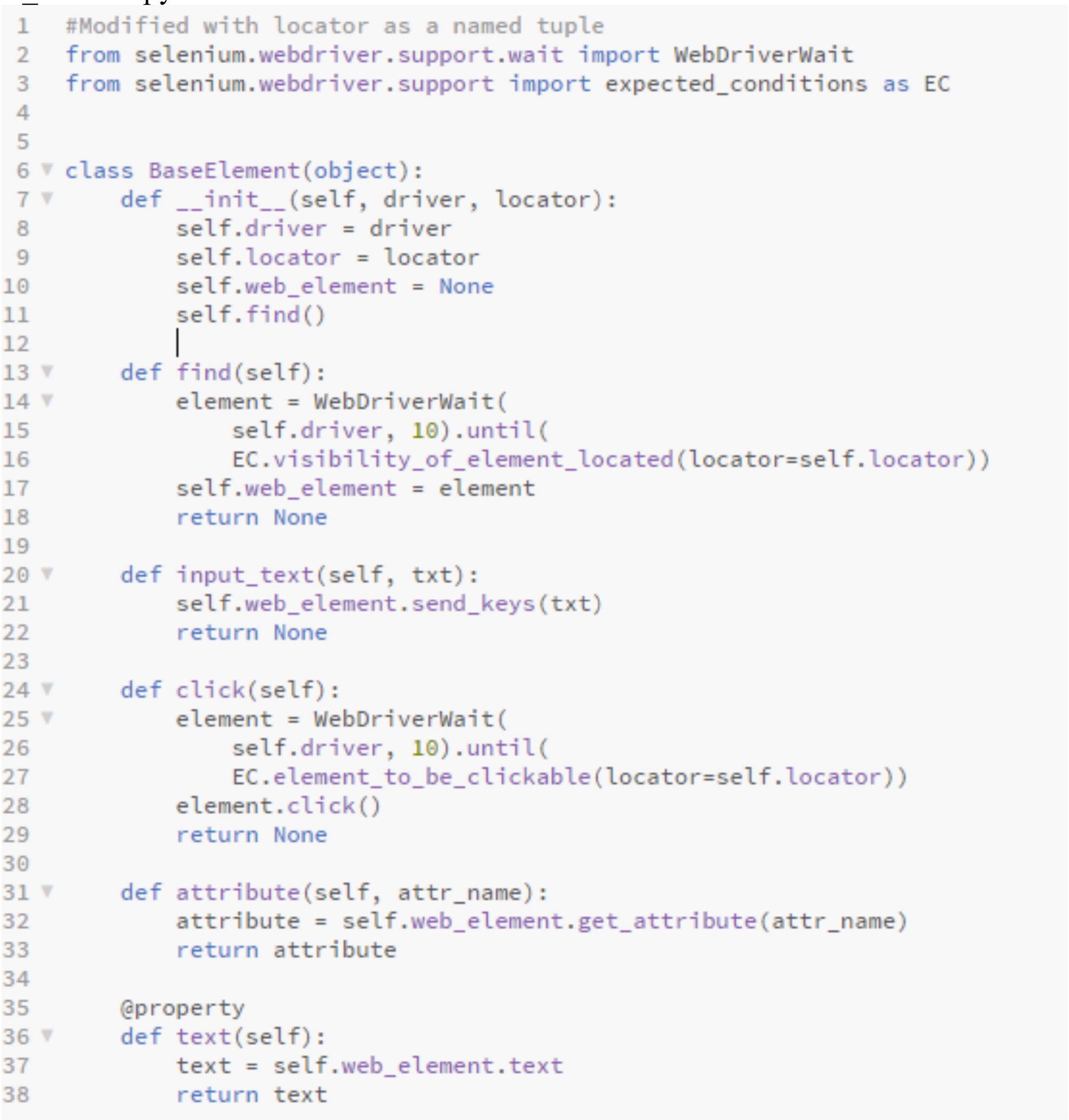
```python
class BasePage(object):

    url = None

    def __init__(self, driver):
        self.driver = driver

    def go(self):
        self.driver.get(self.url)
```

locator.py



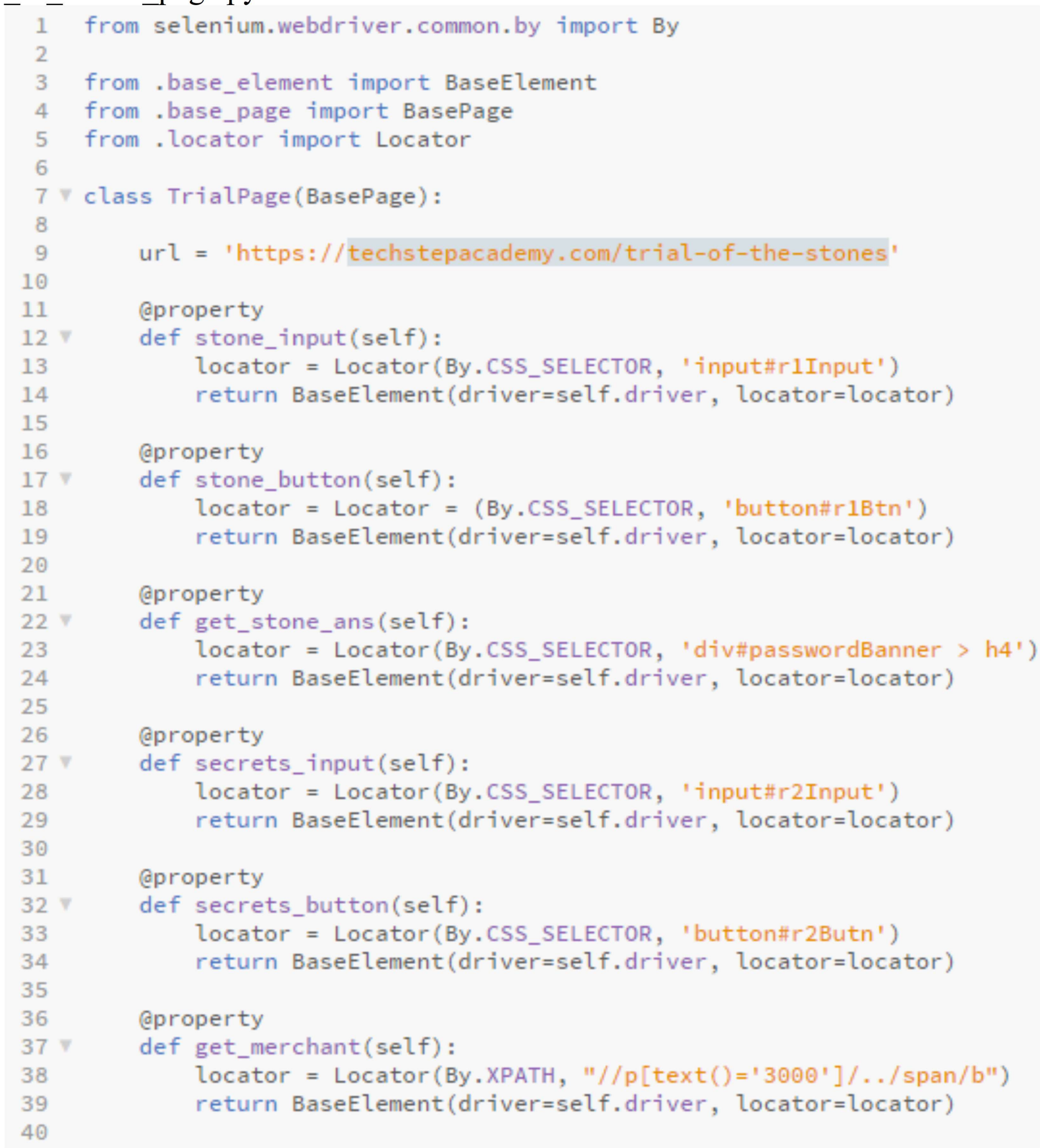
```python
from collections import namedtuple

Locator = namedtuple('Locator', ['by', 'value'])
```

base_element.py

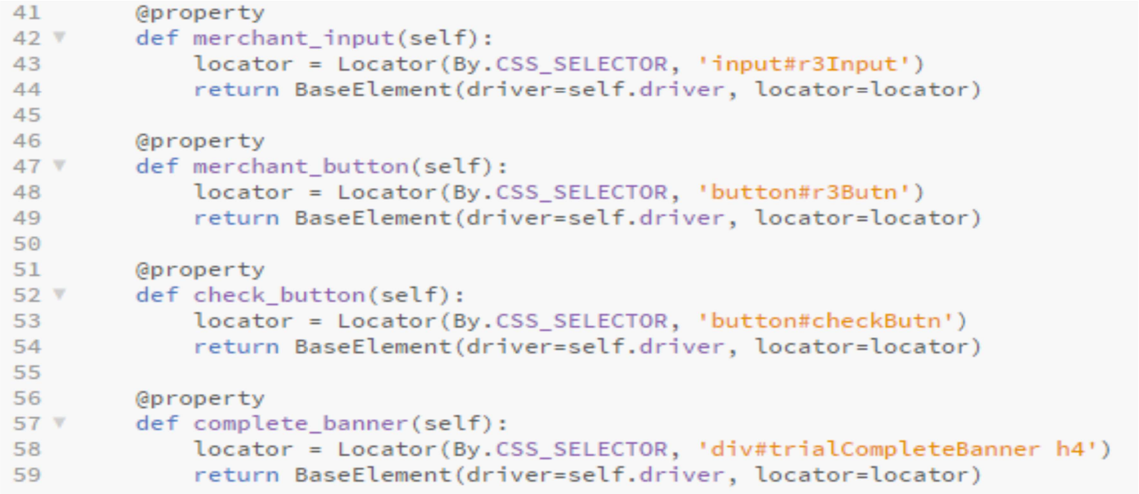


```python
#Modified with locator as a named tuple
from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC


class BaseElement(object):
    def __init__(self, driver, locator):
        self.driver = driver
        self.locator = locator
        self.web_element = None
        self.find()

    def find(self):
        element = WebDriverWait(
            self.driver, 10).until(
            EC.visibility_of_element_located(locator=self.locator))
        self.web_element = element
        return None

    def input_text(self, txt):
        self.web_element.send_keys(txt)
        return None

    def click(self):
        element = WebDriverWait(
            self.driver, 10).until(
            EC.element_to_be_clickable(locator=self.locator))
        element.click()
        return None

    def attribute(self, attr_name):
        attribute = self.web_element.get_attribute(attr_name)
        return attribute

    @property
    def text(self):
        text = self.web_element.text
        return text
```

trial_of_stones_page.py



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

from .base_element import BaseElement
from .base_page import BasePage
from .locator import Locator

class TrialPage(BasePage):

    url = 'https://techstepacademy.com/trial-of-the-stones'

    @property
    def stone_input(self):
        locator = Locator(By.CSS_SELECTOR, 'input#r1Input')
        return BaseElement(driver=self.driver, locator=locator)

    @property
    def stone_button(self):
        locator = Locator = (By.CSS_SELECTOR, 'button#r1Btn')
        return BaseElement(driver=self.driver, locator=locator)

    @property
    def get_stone_ans(self):
        locator = Locator(By.CSS_SELECTOR, 'div#passwordBanner > h4')
        return BaseElement(driver=self.driver, locator=locator)

    @property
    def secrets_input(self):
        locator = Locator(By.CSS_SELECTOR, 'input#r2Input')
        return BaseElement(driver=self.driver, locator=locator)

    @property
    def secrets_button(self):
        locator = Locator(By.CSS_SELECTOR, 'button#r2Butn')
        return BaseElement(driver=self.driver, locator=locator)

    @property
    def get_merchant(self):
        locator = Locator(By.XPATH, "//p[text()='3000']/../span/b")
        return BaseElement(driver=self.driver, locator=locator)

```



```python
    @property
    def merchant_input(self):
        locator = Locator(By.CSS_SELECTOR, 'input#r3Input')
        return BaseElement(driver=self.driver, locator=locator)

    @property
    def merchant_button(self):
        locator = Locator(By.CSS_SELECTOR, 'button#r3Butn')
        return BaseElement(driver=self.driver, locator=locator)

    @property
    def check_button(self):
        locator = Locator(By.CSS_SELECTOR, 'button#checkButn')
        return BaseElement(driver=self.driver, locator=locator)

    @property
    def complete_banner(self):
        locator = Locator(By.CSS_SELECTOR, 'div#trialCompleteBanner h4')
        return BaseElement(driver=self.driver, locator=locator)
```