HW2 requirements Testing

Source code:
https://github.com/wujiahui62/Homework/tree/master/Software_Engineering_Methodologies/HW2_SEM

1-a. Add a feature to the expression parsing code
I added a feature to parse the strings into decimals

```
def parse_decimal(s):
    whole, fraction = s.split('.')
    return int(whole) + int(fraction) / (10 ** len(fraction))
```

1-b. Add test case in the test_evaluate.py file

```
def test_parse_decimal():
    print("\ntest_parse_decimal\n")
    assert parse_decimal("11.23") == 11.23
    assert parse_decimal("0.23") == 0.23
    assert parse_decimal("999.0000000001") == 999.0000000001
```

We can test the new feature using pytest

```
Jiahuis-MacBook-Pro:HW2_SEM jiahuiwu$ pytest
============================= test session starts ==============================
platform darwin -- Python 3.6.1, pytest-3.4.0, py-1.5.2, pluggy-0.6.0
rootdir: /Users/jiahuiwu/Desktop/HW/HW2_SEM, inifile:
collected 3 items

test_evaluate.py ...                                                    [100%]

========================== 3 passed in 0.02 seconds ===========================
```

Or we call the function directly in the evaluate.py file to see if it can parse the string into decimal, they are the same

```
print(parse_decimal("11.23") == 11.23)
```

```
Jiahuis-MacBook-Pro:HW2_SEM jiahuiwu$ python3 evaluate.py
True
```

2. Verify using requests and pytest that some API on this list works correctly:
https://github.com/toddmotto/public-apis

I chose two API to test, the first one is called TacoFancy http://taco-randomizer.herokuapp.com/, visiting this page will get users a random taco. I

tested the connection, the headers content-type and keywords in the response text.

The second one is called LinkPreview, this API Get JSON formatted summary with title, description and preview image for any requested URL. I use this API to get the json data from the url https://www.google.com. The connection and header's content-type was tested. The properties of the returned data such as 'title', 'url' are also tested.

```
import requests
import json

"""
The app randomly pick a recipe, however, there is strict rule that the title of the
webpage should contain some keywords, so I tested if these keywords are there
"""
def test_taco_app_basic():
    response = requests.get("http://taco-randomizer.herokuapp.com/")
    assert response
    assert response.status_code == 200
    assert response.headers['Content-Type'] == 'text/html; charset=utf-8'
    text = response.text
    assert 'taco' in text
    assert 'garnished with' in text
    assert 'topped off with' in text
    assert 'and wrapped in delicious' in text
"""
This API is to Get JSON formatted summary with title, description and preview image
for any requested URL. I use this API to get data from google.com, I tested if the
returned format contains these attributes such as title, description, etc.
"""
def test_myjson_app_basic():
    url = 'https://www.google.com'
    response = requests.get("http://api.linkpreview.net/?key=123456&q=" + url)
    assert response
    assert response.status_code == 200
    assert response.headers['Content-Type'] == 'application/json; charset=utf-8'
    result = json.loads(response.text)
    assert type(result) is dict
    assert 'title' in result
    assert 'Google' in result['title']
    assert 'url' in result
    assert url in result['url']
    assert 'description' in result
    assert 'image' in result
```

Result of the pytest, these two APIs are works correctly.

```
Jiahuis-MacBook-Pro:HW2_SEM jiahuiwu$ pytest -vs
============================= test session starts ==============================
platform darwin -- Python 3.6.1, pytest-3.4.0, py-1.5.2, pluggy-0.6.0 -- /Users/jiahuiwu/anaconda/bin/p
ython
cachedir: .pytest_cache
rootdir: /Users/jiahuiwu/Desktop/HW/HW2_SEM, inifile:
collected 2 items

test_api.py::test_taco_app_basic PASSED
test_api.py::test_myjson_app_basic PASSED

=========================== 2 passed in 1.25 seconds ===========================
```

3. Verify using webdriver and pytest that either "Oster" or "Hamilton Beach" is one of the blenders brands sold on Amazon on the "blender" search page.

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

browser = None

def setup_module(module):
    global browser
    browser = webdriver.Chrome()
    browser.get("http://www.amazon.com")

def teardown_module(module):
    if browser:
        browser.close()

def test_go_to_amazon():
    assert "Amazon" in browser.title

def test_is_Oster_a_displayed_blender():
    id1 = "twotabsearchtextbox"
    searchbox = browser.find_element_by_id(id1)
    searchbox.clear()
    searchbox.send_keys("blender")
    searchbox.send_keys(Keys.RETURN)
    id2 = "s-results-list-atf"
    result_list = browser.find_element_by_id(id2)
    assert "Oster" in result_list.text
```

Test result shows that “Oster” is in the search result of blender on Amaozn

```
Jiahuis-MacBook-Pro:HW2_SEM jiahuiwu$ pytest -vs
============================= test session starts ==============================
platform darwin -- Python 3.6.1, pytest-3.4.0, py-1.5.2, pluggy-0.6.0 -- /Users/jiahuiwu/anaconda/bin/p
ython
cachedir: .pytest_cache
rootdir: /Users/jiahuiwu/Desktop/HW/HW2_SEM, inifile:
collected 2 items

test_blender_amazon.py::test_go_to_amazon PASSED
test_blender_amazon.py::test_is_Oster_a_displayed_blender PASSED
```