Task

Given a `string` representing the url for NY Times Today's,

you must write a python script using the `BeautifulSoup` and `requests` modules to scrap all the headlines of that page.

Example 1

Input:

```
url = "https://www.nytimes.com/issue/todayspaper/2021/11/01/todays-new-york-times"
```

Output:

```
result = [
    ...
    "Angling for a Merry ‘Fishmas’ Despite Global Shipping Delays",
    "Who Had Covid-19 Vaccine Breakthrough Cases?",
    ...
]
```

Explanation:

These are some of the articles in this web page. Since the list is quite extensive, we provide an excerpt of the articles found in the page.

Resources

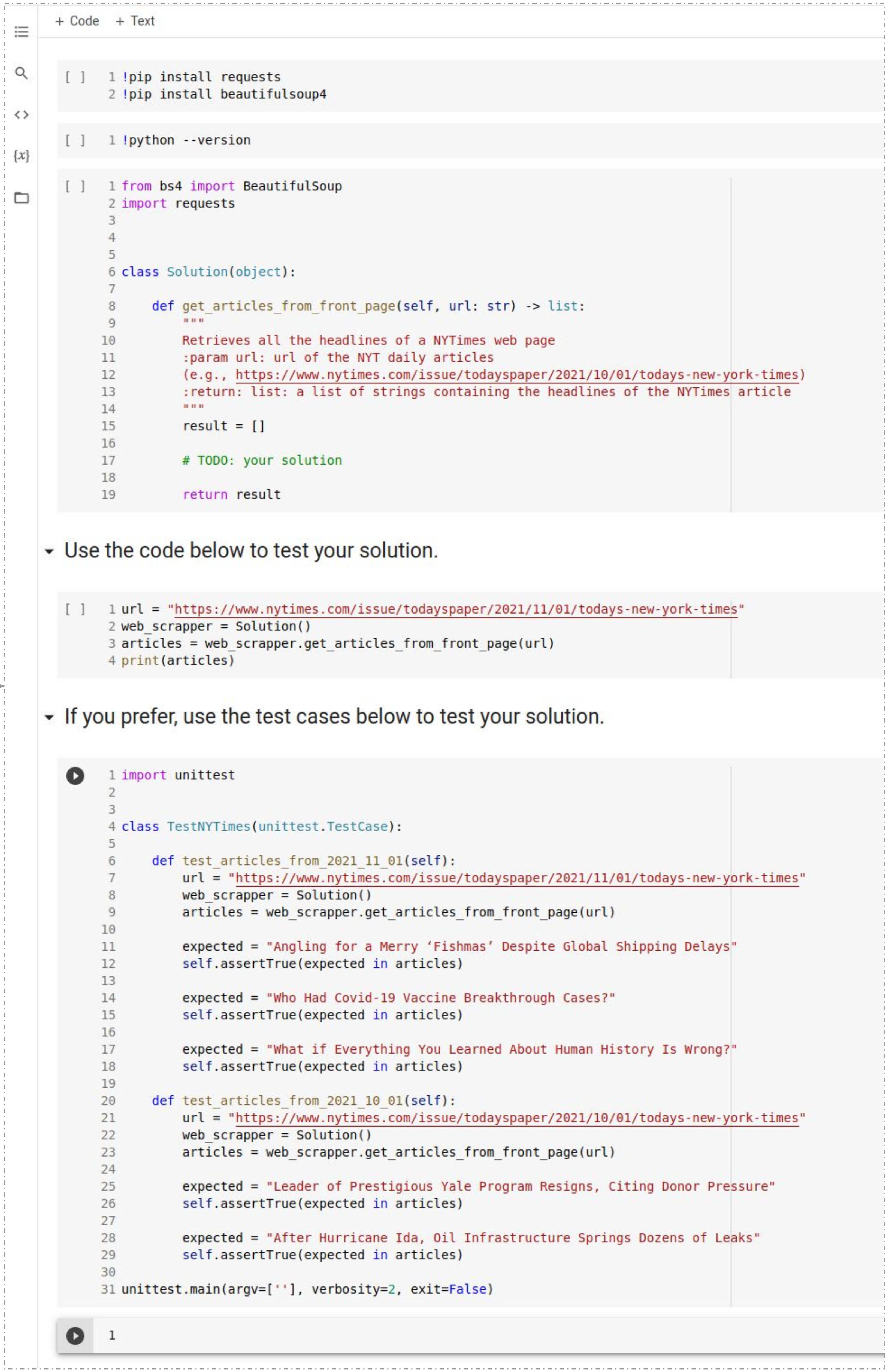
Please **use only** the following resources to find information that might assist you complete this task:

- Requests: HTTP for Humans™
- Requests API
- Beautiful Soup Documentation
- Tutorial: Web Scraping with Python Using Beautiful Soup
- Extracting an attribute value with beautifulsoup
- How to find children of nodes using BeautifulSoup
- How to find elements by class
- How to extract HTTP response body from a Python requests call?
- Beautiful Soup: Build a Web Scraper With Python
- Web Scraping with BeautifulSoup

tip: `ctrl + left mouse click` opens each link in a new tab

Colab

coding environment



```
[ ]
!pip install requests
!pip install beautifulsoup4
```

```
[ ]
!python --version
```

```
[ ]
from bs4 import BeautifulSoup
import requests



class Solution(object):

    def get_articles_from_front_page(self, url: str) -> list:
        """
        Retrieves all the headlines of a NYTimes web page
        :param url: url of the NYT daily articles
        (e.g., https://www.nytimes.com/issue/todayspaper/2021/10/01/todays-new-york-times)
        :return: list: a list of strings containing the headlines of the NYTimes article
        """
        result = []

        # TODO: your solution

        return result
```

Use the code below to test your solution.

```
[ ]
url = "https://www.nytimes.com/issue/todayspaper/2021/11/01/todays-new-york-times"
web_scrapper = Solution()
articles = web_scrapper.get_articles_from_front_page(url)
print(articles)
```

If you prefer, use the test cases below to test your solution.

```
import unittest


class TestNYTimes(unittest.TestCase):

    def test_articles_from_2021_11_01(self):
        url = "https://www.nytimes.com/issue/todayspaper/2021/11/01/todays-new-york-times"
        web_scrapper = Solution()
        articles = web_scrapper.get_articles_from_front_page(url)

        expected = "Angling for a Merry ‘Fishmas’ Despite Global Shipping Delays"
        self.assertTrue(expected in articles)

        expected = "Who Had Covid-19 Vaccine Breakthrough Cases?"
        self.assertTrue(expected in articles)

        expected = "What if Everything You Learned About Human History Is Wrong?"
        self.assertTrue(expected in articles)

    def test_articles_from_2021_10_01(self):
        url = "https://www.nytimes.com/issue/todayspaper/2021/10/01/todays-new-york-times"
        web_scrapper = Solution()
        articles = web_scrapper.get_articles_from_front_page(url)

        expected = "Leader of Prestigious Yale Program Resigns, Citing Donor Pressure"
        self.assertTrue(expected in articles)

        expected = "After Hurricane Ida, Oil Infrastructure Springs Dozens of Leaks"
        self.assertTrue(expected in articles)

unittest.main(argv=[''], verbosity=2, exit=False)
```