edX (https://www.edx.org)
MITx: 6.00.1x Introduction to Computer Science and Programming Using Python
sandipan_dey (/dashboard)
Courseware (/courses/MITx/6.00.1_4x/3T2014/courseware) Updates & News (/courses/MITx/6.00.1_4x/3T2014/info)
Calendar (/courses/MITx/6.00.1_4x/3T2014/89309559b0414f6d8cbef9e48ca19f4b/) Wiki (/courses/MITx/6.00.1_4x/3T2014/course_wiki)
iscussion (/courses/MITx/6.00.1_4x/3T2014/discussion/forum) Progress (/courses/MITx/6.00.1_4x/3T2014/progress)
Help

Conflict with Grader

To avoid getting an error from the grader about using the class variables `title`, `subject`, or `summary`, rename any variable you have used with these names to something else. Thank you!

PART I: DATA STRUCTURE DESIGN (5/5 points)

First, let's talk about one specific RSS feed: Google News. The URL for the Google News feed is:

http://news.google.com/?output=rss (http://news.google.com/?output=rss)

If you try to load this URL in your browser, you'll probably see your browser's interpretation of the XML code generated by the feed. You can view the XML source with your browser's "View Page Source" function, though it probably will not make much sense to you. Abstractly, whenever you connect to the Google News RSS feed, you receive a **list of items**. Each **entry** in this list represents a single news item. In a Google News feed, every entry has the following fields:

- `guid` : A globally unique identifier for this news story.
- `title` : The news story's headline.
- `subject` : A subject tag for this story (e.g. 'Top Stories', or 'Sports').
- `summary` : A paragraph or so summarizing the news story.
- `link` : A link to a web-site with the entire story.

Generalizing the Problem

This is a little trickier than we'd like it to be, because each of these RSS feeds is structured a little bit differently than the others. So, our goal in Part I is to come up with a unified, standard representation that we'll use to store a news story.

Why do we want this? When all is said and done, we want an application that aggregates several RSS feeds from various sources and can act on all of them in the exact same way: we should be able to read news stories from various RSS feeds all in one place. If you've ever used an RSS feed reader, be assured that it has had to solve the exact problem we're going to tackle in this pset!

PROBLEM 1

Parsing is the process of turning a data stream into a structured format that is more convenient to work with. We have provided you with code that will retrieve and parse the Google and Yahoo news feeds.

Parsing all of this information from the feeds that Google/Yahoo/the New York Times/etc. gives us is no small feat. So, let's tackle an easy part of the problem first: Pretend that someone has already done the specific parsing, and has left you with variables that contain the following information for a news story:

- globally unique identifier (GUID) - a string that serves as a unique name for this entry
- title - a string
- subject - a string

- summary - a string
- link to more content - a string

We want to store this information in an *object* that we can then pass around in the rest of our program. Your task, in this problem, is to write a class, `NewsStory`, **starting with a constructor** that takes `(guid, title, subject, summary, link)` as arguments and stores them appropriately. NewsStory also needs to contain the following methods:

- `getGuid(self)`
- `getTitle(self)`
- `getSubject(self)`
- `getSummary(self)`
- `getLink(self)`

Each method should return the appropriate element of an instance. For example, if we have implemented the class and call

```
test = NewsStory('foo', 'myTitle', 'mySubject', 'some long summary', 'www.example.com')
```

then `test.getGuid()` will return `foo`.

The solution to this problem should be relatively short and very straightforward (please review what `get` methods should do if you find yourself writing multiple lines of code for each). Once you have implemented `NewsStory` all the `NewsStory` test cases should work.

To test your class definition, we have provided a test suite in `ps7_test.py`. You can test your code by loading and running this file. You should see an "OK" for the `NewsStory` tests if your code is correct. Because `ps7.py` contains code to run the full RSS scraping system, we suggest you do not try to run `ps7.py` directly to test your implementation. Instead, in IDLE, you can do the following:

```
>>> from ps7 import *
>>> test = ps7.NewsStory('foo', 'myTitle', 'mySubject', 'some long summary', 'www.example.com')
```

to load in then run your own tests on your class definitions.

Canopy Specific Instructions: If you are getting an error, type the following instead:

```
>>> cd [insert the full path of the directory where your code resides]
>>> from ps7 import *
>>> test = NewsStory('foo', 'myTitle', 'mySubject', 'some long summary', 'www.example.com')
```

```
# Enter your code for NewsStory in this box
class NewsStory(object):

    def __init__(self, guid, title, subject, summary, link):
        self.guid = guid
        self.title = title
        self.subject = subject
        self.summary = summary
        self.link = link

    def getGuid(self):
        return self.guid

    def getTitle(self):
        return self.title
```

Correct

Test results

Hide output

CORRECT

Test: Constructor

Can we construct a NewsStory object without error
s = NewsStory('guid', 'title', 'subj', 'sum', 'link')

Output:

```
```

Test: getGuid

Can we get a GUID?
s = NewsStory('guid', 'title', 'subj', 'sum', 'link')
s.getGuid()

Output:

```
guid
Testing a new story with a randomized guid
ymhipq
```

Test: getLink

Can we get a link?
s = NewsStory('guid', 'title', 'subj', 'sum', 'link')
s.getLink()

Output:

```
link
Testing a new story with a randomized link
www.xervoq.com
```

Test: getSubject

Can we get a subject?
s = NewsStory('guid', 'title', 'subj', 'sum', 'link')
s.getSubject()

Output:

```
subj
Testing a new story with a randomized subject
Canada, Hockey, obfio
```

Test: getSummary

Can we get a summary?
s = NewsStory('guid', 'title', 'subj', 'sum', 'link')
s.getSummary()

Output:

```
sum
Testing a new story with a randomized summary
It was the Queen Mum's birthday! London went crazy! nlhoz
```

`Test: getTitle`

Can we get a title?
s = NewsStory('guid', 'title', 'subj', 'sum', 'link')
s.getTitle()

Output:

```
title
Testing a new story with a randomized title
Obamny is sure in the US news a lot lately vxhdtvwies
```

`Test: multiple instances`

In this test we make three instances of NewsStory and ask
in turn for their guids and titles.

Output:

```
Testing: getting guids for each of three unique stories
jqqtdmzkf
nkqhaxo
fafpiglm
Testing: getting titles for each of three unique stories
Justin Bieber Fever!
One Direction tops the charts!
Rest of World Sick of Tween Pop Music
```

`Test: multiple instances 2`

In this test we make three instances of NewsStory and ask
in turn for their subjects, summaries, and links.

Output:

```
Testing: getting subjects for each of three unique stories
Science
Politics
Fashion
Testing: getting summaries for each of three unique stories
New temperature proposed for absolute zero: -1 degree Kelvin?! Find out what leading ultra cold
atom researchers have to say!
Neil de Grasse Tyson running for President of Mars, sources say.
Lady Gaga stuns fashion world by wearing Ann Taylor pantsuit. "She looks like Margaret Thatcher"
raves Giuliana Rancic
Testing: getting links for each of three unique stories
http://www.nature.com/news/quantum-gas-goes-below-absolute-zero-1.12146
www.example.politics.com
www.example.fashion.com
```

Hide output

Check Save *You have used 1 of 30 submissions*

Show related resources ▼

Show Discussion New Post