

CS 112 Project Assignment: Image Browser

Overview

The purpose of this project is to gain experience using functions, lists, string parsing and data persistence (pickling or shelves) and python packages. Additionally you will handle more user errors than in previous assignments.

In the project you will use Python to open a URL (website) and store the image locations (URLs) and alt-text description of the image in a data file.

You will then present a text-based menu of all images in your data file and allow the user to view any of them.

This project is due 4/13/2009 at 11:55PM for everyone in CS112. Programming projects are considered individual efforts, therefore no sharing of code and/or discussion of algorithms or problem solution is allowed with anyone except lab GTA, UTA or instructor.

Requirements

- The source code for this project must be submitted in a file named **imagebrowser.py**
 - The source code file must contain a **file header** formatted as previously specified.
 - The source code file should use **self-documenting code** and additional comments (as required) to improve code readability.
-

Procedure

Your menu will have options for each image in the data file

1. Ask the user for a URL. You must correctly accept a URL in the following forms:

<http://www.somewhere.com>

[HTTP://www.somewhere.com](http://www.somewhere.com)
www.somewhere.com

All of those URLs should take the user to the same place. Additionally, you must accept HTML files:

<http://www.somewhere.com/somedir/myfile.html>

2. Read the data (HTML) from the given URL. See the urllib module provided by Python for how to do this. If the URL does not exist it is okay to let the program crash. (Once we learn exceptions in the last part of the semester you can fix this problem.)

3. Create a list of all images and their alt tags. In HTML webpages an image is placed onto the screen using this format:

```

```

For this project you must find all image tags and store the image location and the alt (alternate) text. If the image does not have alternate text (which does happen) you should use the image name as alt text.

Issues to handle:

- Anything mixed case ``
- Images with a full URL: ``

4. Add that list of images, alttags, and the URL to your data file. It is REQUIRED that you use Python pickling or shelves in the creation, updating and usage of the datafile.

5. Display a menu to the user with a numbered list of all images currently in the datafile. The menu should show the alternate text. For example:

1. A student
 2. People walking
- q to quit
n to select new URL

Selection →

(If there is no alt text for the image, it should show the image file location.)

5. When the user selects:

- 'q' the program should end.
- 'n' the program should ask the user for another URL and show the menu

Any valid number from the menu: the program should display that image using :

On Mac use:

```
os.system("open http://www.somewhere.com/student.jpg")
```

On Windows use:

```
os.system("start http://www.somewhere.com/student.jpg")
```

Any other text (including numbers that are too high or too low) the program should say is invalid and ask the user for another option.

Coding Requirements

For this project you **MUST** use functions. You must have a function or functions that parse the URL and returns lists of images and alt tags.

Additionally you must **NOT** retrieve the URL multiple times. You should only retrieve a URL once when then user selects "n"ew URL. When the menu displays you should not need to go to the URL again, that information should be stored in a data structure.

Bonus Points

If you want to get a 10% bonus on this assignment you can add in the following requirements:

- Create another menu option for "thumbnail view".
 - o In this option you should download all pictures from the website (urllib may help)
 - o Use Python Imaging Library (PIL) to resize them to a thumbnail size
 - o Pack them all into one big image (using PIL)
 - o Open that image.
-

Hints

1. Things I used: shelves, urllib, String's find method, and upper case method.
2. Search for images by searching for "IMG" (handling upper/lowercase appropriately)
3. Make sure to check that you're finding things within the image tag by limiting most string searches to the end of your image tag by finding the ">" (closing of the IMG tag). See the find documentation for how to do that.

4. One simple way to store the information to a file is to store three variables (using either pickling or shelves). The variables are a list of image locations, alt tags and URLs. For example, my lists look like this:

URL List

<http://www.gmu.edu>
<http://www.gmu.edu>
<http://www.yahoo.com>
<http://www.yahoo.com>

File Location List

index_files/images/mason_logo.gif
index_files/images/students.gif
<http://l.yimg.com/a/i/ww/beta/y3.gif>
<http://l.yimg.com/a/i/ww/tbl/allys.gif>

Alt Tags

George Mason University
Students
Yahoo!
All Yahoo! Services

Using these three lists you can create a menu fairly easily because one index will let you find the URL, File location and Alt Tag. This does mean the URL is stored many times, but it is a small cost to have simpler code. You may use any technique you want, this is just a suggestion.

5. If you need to deal with mixed case issues, why not just convert everything to a known case so you are certain what case the text is in?

Sample Code and Test Page

There is a sample HTML with various tests on it here:

<http://cs.gmu.edu/~dfleck/classes/cs112/spring09/projects/imagebrowser/imagetest.html>

To help we'll also give you a working solution. This should help you understand what we're looking for. In your code you do not have to create the exact same menu, usage, etc... but it should show the same images, and be pretty close. You MUST read the specification though... simply mimicking my program may result in you not handling certain issues/errors. You should test all conditions in the specification! My program may also have errors...let us know if you find any ☺

If you use Python 2.5:

<http://cs.gmu.edu/~dfleck/classes/cs112/spring09/projects/imagebrowser/imagebrowser25.pyc>

If you use Python 2.6:

<http://cs.gmu.edu/~dfleck/classes/cs112/spring09/projects/imagebrowser/imagebrowser26.pyc>

You can run these using the Python on the command line (remember lab 1?)

Grading Rubric

	Excellent (85% or higher)	Average (60% or higher)	Needs Improving (Less than 60%)	Points Possible
Submission Instructions	<ul style="list-style-type: none"> Both filename and header meet the stated specifications 	<ul style="list-style-type: none"> Either is missing or has incorrect/missing information 	<ul style="list-style-type: none"> Both are missing or have incorrect details 	3
User interface	<ul style="list-style-type: none"> Menu is created and correctly displays options Menu accepts options and calls appropriate code System opens the required image upon menu selection System exist upon exit menu selection Formatting of menu options looks good 	<ul style="list-style-type: none"> Menu is created, but some options do not function Menu doesn't allow endless choices (not in a loop) Formatting of menu does not look good 	<ul style="list-style-type: none"> Menu is missing or non-functional System cannot open the image as required 	5
Use of functions	<ul style="list-style-type: none"> Functions are used as specified Comments are meaningful and professional. Code is clean and easy to read. Uses meaningful variable names 	<ul style="list-style-type: none"> Functions are used, but are not clear Comments are not helpful for understanding the code Variable names are not meaningful 	<ul style="list-style-type: none"> Code is written without appropriate use of functions Code is very difficult to read and understand due to poor coding style (missing comments/self-documenting code) 	8
Data persistence	<ul style="list-style-type: none"> Code correctly uses pickling or shelves to store the data into a file for later use. 	<ul style="list-style-type: none"> Code can store or retrieve data, but not both. Or has errors in one operation. 	<ul style="list-style-type: none"> Code does not store or retrieve data. 	8
Error handling	All errors and formats of user input are handled appropriately (as specified).	Some errors are handled, but others cause either incorrect output or system crashes.	Many/most errors are not handled and cause incorrect output or crashes.	8
Reading and parsing URL	<ul style="list-style-type: none"> The system correctly opens and reads URLs. The URLs are parsed and IMG and ALT tags are found and stored correctly. 	<ul style="list-style-type: none"> The system correctly opens and reads URLs. The URL parsing works in only some cases and fails in others 	<ul style="list-style-type: none"> The system does not correctly open and reads URLs. The URL parsing does not work and hinders the menu from being displayed 	8
Bonus Points	Code correctly displays a thumbnail image of all accessible pictures from a website.	None given.	None given.	(+4)
Final Score				40 (+4)