

HW6: CS 110X C 2014

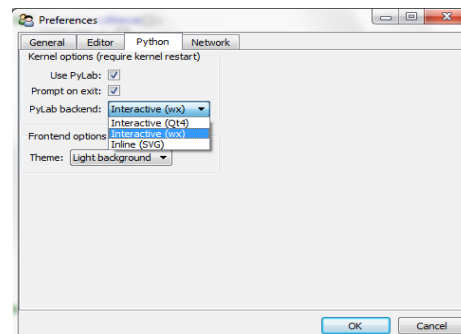
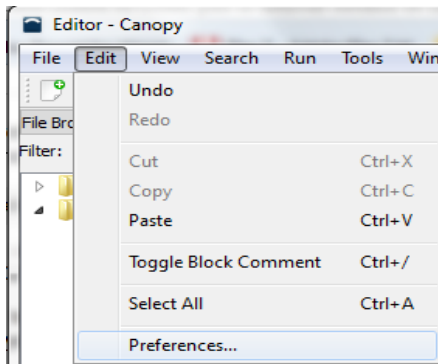
Note: This homework (and all remaining homework assignments) is a **partner homework** and must be completed by each partner pair. When you complete this assignment, you must not share your answers with any other student. Only one person from a partner pair needs to submit the assignment, but make sure that you submit before the deadline!

For this assignment, every function that you write must have a suitable documentation string as we present in class. Check the rubric to see the point values assigned for each question so you can maximize the points you get on this assignment.

Please make sure that when you submit your assignment, you submit a single “HW6.py” file that contains your entire assignment.

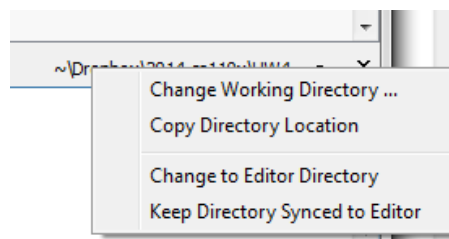
Canopy Issues

If you are running Canopy then you have to make a small configuration change for this homework to work properly. From within the Canopy Editor, select menu item **Edit | Preferences...**



Then in the Preferences window, select the **Python** tab and be sure that **PyLab backend** is set to “Interactive (wx)”. I have tested this on Windows and on a MacBook.

In Canopy, you need to set the working directory within the code editor. Near the right edge about one inch from the bottom you will see a small triangle that you can click on to change the working directory. Select the “Change to Editor Directory” so you will find the files that you need for this assignment.



Homework Themes

This homework will reinforce skills on **for** loops, **string** manipulation, and **CSV** files. You are also exposed to creating separate Python modules to store code you have written.

Homework Instructions

This Homework has **Five** questions, because of the lost day from Academic Advising.

For each question be sure you understand exactly the format of the output that is requested. You will lose points if you do not exactly follow the format of the output for the individual questions. Should you have any questions, be sure to review the HW4 rubric and post questions on the HW4 discussion forum.

Q1	Demonstrate ability to work with lists and dictionaries		
<table border="1"> <thead> <tr> <th>Skills</th> </tr> </thead> <tbody> <tr> <td>DT-11 CS-1 CS-9 CS-1 DT-10</td> </tr> </tbody> </table>	Skills	DT-11 CS-1 CS-9 CS-1 DT-10	<p>Write a function <code>summarize(values)</code> that returns a dictionary representing the counts of unique elements in <code>values</code>.</p> <p>That is, the dictionary will count the number of times each unique item in <code>values</code> appears in the list.</p> <p>This function could be used, for example, to generate a histogram of values.</p>
Skills			
DT-11 CS-1 CS-9 CS-1 DT-10			
Sample Output in IDLE	<pre>>>> summarize([2,3,4,2,4,3,2,4,2]) {2: 4, 3: 2, 4: 3} >>> summarize(['a', 'b', 'a', 'c', 'b']) {'a': 2, 'b': 2, 'c': 1}</pre>		

Q2	String manipulation		
<table border="1"> <thead> <tr> <th>Skills</th> </tr> </thead> <tbody> <tr> <td>CS-9 SM-3</td> </tr> </tbody> </table>	Skills	CS-9 SM-3	<p>Given a string literal containing arbitrary characters, one often wants to eliminate all characters that are not alphabetic. Write a function <code>sanitize(s)</code> that returns a new string literal that contains only alphabetic characters in <code>s</code> as processed from left to right.</p> <p>Note that an alphabetic character is a letter from a-z and A-Z.</p> <p>Hint: String concatenation will be useful for this function.</p>
Skills			
CS-9 SM-3			
Sample Output in IDLE	<pre>>>> sanitize("isn't this great?") 'isntthisgreat' >>> sanitize("Whatever") 'Whatever' >>> sanitize("Non-gratis") 'Nongratis'</pre>		

Q3	String manipulation
<div style="border: 1px solid black; padding: 2px;"> Skills SM-3 CS-6 DT-6 </div>	<p>Write a function <code>decodeChunks(s)</code> that returns a list containing the original string <code>s</code> subdivided into string literal chunks of seven characters at a time.</p> <p>Note that the original string can contain any number of characters, so the final string literal in the returned list may contain 1 to 6 characters.</p> <p>When solving this problem, first get it to work for strings whose length is an even multiple of seven (first example below) then consider solving the arbitrary case (second and third examples below)</p> <p>Hint: A while loop might be useful here.</p>
Sample Output in IDLE	<pre>>>> decodeChunks('Unintelligible') ['Unintel', 'ligible'] >>> decodeChunks('this is a test of how this works.') ['this is', ' a test', ' of how', ' this w', 'orks.'] >>> decodeChunks('Done') Out[27]: ['Done']</pre>

Q4	String manipulation
<div style="border: 1px solid black; padding: 2px;"> Skills SM-3 CS-6 CS-11 </div>	<p>Write a function <code>shuffle(str1, str2)</code> that returns a new string formed by interweaving alternating characters from each string to form a new string.</p> <p>Note that if one string is longer than the other, all remaining characters are just appended to the end of the new string.</p> <p>As you can see from the sample output, you first concatenate the first characters of the input strings, then the second characters and so on, until one string has run out of letters.</p>
Sample Output in IDLE	<pre>>>> shuffle('test', 'enough') 'teensotugh' >>> shuffle('this', 'that') 'tthhiast' >>> shuffle('abcd', 'wxyz') 'awbxcydz'</pre>

Q5	Demonstrate knowledge of CSV files and modules			
<table border="1" data-bbox="191 331 370 478"> <tr> <td data-bbox="191 331 370 365"><i>Skills</i></td> </tr> <tr> <td data-bbox="191 365 370 399">DT-11</td> </tr> <tr> <td data-bbox="191 399 370 478">IO-5</td> </tr> </table>	<i>Skills</i>	DT-11	IO-5	<p>For this question you are to create a helper module that contains the <code>extractAllRecords(filename)</code> function that you used on HW5. Create a file <code>helper.py</code> and insert the existing implementation of this function in that file (you can retrieve a suitable implementation from the posted HW5 solution).</p> <p>Make sure that <code>helper.py</code> is saved to the same directory as your <code>HW6.py</code> file. Note that your <code>helper.py</code> file must import <code>csv</code> to work properly.</p> <p>Now include the statement <code>import helper</code> at the front of your <code>HW6.py</code> file as you have seen me do with the <code>pylab</code> and <code>math</code> modules.</p> <p>Write a <code>generateDictionary(fileName, keyField, valueField)</code> function that processes the CSV file with the given <code>fileName</code> (header information and all) and returns a dictionary with entries whose string literal (key, value) pairs are drawn from each of the rows (excluding the header) in <code>listOfRows</code> as identified by the string literals <code>keyField</code> and <code>valueField</code>.</p> <p>Note: if two or more rows have the same key then the dictionary will store the (key, value) from the later row in the original <code>listOfRows</code> structure.</p> <p>Note that your function must invoke <code>helper.extractAllRecords(filename)</code> which is how Python invokes functions that exist within another module.</p>
<i>Skills</i>				
DT-11				
IO-5				

Sample Input File ([smallDataSet.csv](#))

```
Quarter,Stores,Stores Closed,New Stores
2014Q1,992,20,14
2013Q4,998,6,16
2013Q3,988,4,18
2013Q2,970,0,16
2013Q1,954,8,12
2012Q3,950,11,8
```

Sample Output

```
>>> generateDictionary ('smallDataSet.csv', 'Quarter', 'Stores')
{'2012Q3': '950',
 '2013Q1': '954',
 '2013Q2': '970',
 '2013Q3': '988',
 '2013Q4': '998',
 '2014Q1': '992'}
>>> generateDictionary ('smallDataSet.csv', 'Quarter', 'New Stores')
{'2012Q3': '8',
 '2013Q1': '12',
 '2013Q2': '16',
 '2013Q3': '18',
 '2013Q4': '16',
 '2014Q1': '14'}
```

Version: 2/20/2014

How To Get Started On This Assignment

A template HW6.py file is provided to you with some sample functions already provided.

Note: You don't have to submit the helper.py file because you will be graded assuming that your HW6.py file imports helper.

Submit your HW6.py file using the web-based turnin system. As we have mentioned in class, only one of the team members needs to submit the assignment. But just make sure that something gets submitted!

Change Log

- 1.