

# CMPUT 175 - Lab 4: Web Browser with a List

Goal: Use a built-in data structure (Python list) to simulate the forward and back button functionality of a web browser.

You are tasked with creating a web browser simulator. Your web browser simulator should **use one list** together with a variable containing the **current index** to enable both the **back button** functionality and the **forward button** functionality. Your simulator will not actually display any webpages: it will just display the URL address of the current page that the user is on. (See sample run at the end of this Exercise). When the user wishes to enter a new webpage address, '=' is entered. When the user wishes to go back, '<' is entered. When the user wishes to go forward, '>' is entered. The user can quit by entering 'q' when prompted.

**Ensure that you write proper docstrings and follow the Software Quality Requirements. See Deliverables on the last page.**

## Part 1:

1. To aid your understanding, complete the worksheet at the end of this lab to show the contents of the list of webpages and the value of the current index after every valid entry in the sample output. This will assist you greatly when writing the code.

## Part 2:

1. Download and save a copy of lab4\_browser.py from eClass. This file contains a *main()* function (done for you) which controls the flow of operation of a web browser simulation. In the following steps, you will complete the functions that this *main()* function calls.
2. Complete *getAction()*. This function prompts the user to enter either a '=' (to enter a new website address), '<' (back button), '>' (forward button), or 'q' to quit the browser simulation. If the user enters something other than these four characters, an error message is displayed before re-prompting for a valid entry. This function has no inputs. This function returns the valid character entered by the user (str).
3. Complete *goToNewSite()*. This function is called when the user enters '=' during *getAction()*. This function prompts the user to enter a new website address, and returns the index to the current site (int). It also updates the list (pages) and current index (current), as appropriate. (*Hint:* experiment with how the back and forward buttons work on a real web browser like Firefox or Chrome. After a new address is entered, can you still go forward?) Note that you do not need to explicitly return the list of websites because the list is a mutable object – so **pages** is actually just an alias for the list called **websites** in your main function. The inputs for this function are the index of the current

website (int), and a reference to the list holding the webpage addresses to go back and forward between.

4. Complete *goBack()*. This function is called when the user enters '<' during *getAction()*. An error message is displayed if there are no webpages stored in the back history, and the index to the current site is returned (int). Otherwise, the index of the previous webpage is retrieved (and returned as an int). The inputs for this function are the index of the current website (int), and a reference to the list holding the webpage addresses to go back and forward between.
5. Complete *goForward()*. This function is called when the user enters '>' during *getAction()*. An error message is displayed if there are no webpages stored in the forward history, and the index of the current site is returned (int). Otherwise, the index of the next website is retrieved (and returned as an int). The inputs for this function are the index of the current website (int), and a reference to the list holding the webpage addresses to go back and forward between.

**Please see the next page for a sample output**

### Sample run:

Currently viewing [www.cs.ualberta.ca](http://www.cs.ualberta.ca)

Enter = to enter a URL, < to go back, > to go forward, q to quit: 123

Invalid entry.

Enter = to enter a URL, < to go back, > to go forward, q to quit: >

Cannot go forward.

Currently viewing [www.cs.ualberta.ca](http://www.cs.ualberta.ca)

Enter = to enter a URL, < to go back, > to go forward, q to quit: <

Cannot go back.

Currently viewing [www.cs.ualberta.ca](http://www.cs.ualberta.ca)

Enter = to enter a URL, < to go back, > to go forward, q to quit: =

URL: [www.google.ca](http://www.google.ca)

Currently viewing [www.google.ca](http://www.google.ca)

Enter = to enter a URL, < to go back, > to go forward, q to quit: <

Currently viewing [www.cs.ualberta.ca](http://www.cs.ualberta.ca)

Enter = to enter a URL, < to go back, > to go forward, q to quit: >

Currently viewing [www.google.ca](http://www.google.ca)

Enter = to enter a URL, < to go back, > to go forward, q to quit: =

URL: [docs.python.org](http://docs.python.org)

Currently viewing [docs.python.org](http://docs.python.org)

Enter = to enter a URL, < to go back, > to go forward, q to quit: <

Currently viewing [www.google.ca](http://www.google.ca)

Enter = to enter a URL, < to go back, > to go forward, q to quit: <

Currently viewing [www.cs.ualberta.ca](http://www.cs.ualberta.ca)

Enter = to enter a URL, < to go back, > to go forward, q to quit: =

URL: [www.beartracks.ualberta.ca](http://www.beartracks.ualberta.ca)

Currently viewing [www.beartracks.ualberta.ca](http://www.beartracks.ualberta.ca)

Enter = to enter a URL, < to go back, > to go forward, q to quit: >

Cannot go forward.

Currently viewing [www.beartracks.ualberta.ca](http://www.beartracks.ualberta.ca)

Enter = to enter a URL, < to go back, > to go forward, q to quit: <

Currently viewing [www.cs.ualberta.ca](http://www.cs.ualberta.ca)

Enter = to enter a URL, < to go back, > to go forward, q to quit: >

Currently viewing [www.beartracks.ualberta.ca](http://www.beartracks.ualberta.ca)

Enter = to enter a URL, < to go back, > to go forward, q to quit: q

Browser closing...goodbye.

### **Worksheet to Match Sample Run:**

**The first 3 steps are already completed to get you started...**

Step 1: Web browser opened; home page displayed

webpages [ www.cs.ualberta.ca ]

currentIndex [ 0 ]

Cannot go forward, cannot go back. No change to webpages or currentIndex.

Step 2: Go to new site; www.google.ca is displayed

webpages [ www.cs.ualberta.ca, www.google.ca ]

currentIndex [ 1 ]

Step 3: Go back to previous page; www.cs.ualberta.ca displayed

webpages [ www.cs.ualberta.ca, www.google.ca ]

currentIndex [ 0 ]

**Complete the rest (draw extra boxes for list elements, as needed)...**

Step 4: Go forward to next page; www.google.ca is displayed

webpages [ ]

currentIndex [ ]

Step 5: Go to new site; www.docs.python.org displayed

webpages [ ]

currentIndex [ ]

Step 6: Go back to previous page; www.google.ca displayed

webpages [ ]

currentIndex [ ]

Step 7: Go back to previous page; www.cs.ualberta.ca is displayed

webpages [ ]

currentIndex [ ]

Step 8: Go to new site; www.beartracks.ualberta.ca displayed

webpages [ ]

currentIndex [ ]

Step 9: Try to go forward, but can't; www.beartracks.ualberta.ca displayed

```
webpages      [          ]  
currentIndex  [  ]
```

Step 10: Go back to previous page; www.cs.ualberta.ca displayed

```
webpages      [          ]  
currentIndex  [  ]
```

Step 11: Go forward to next page; www.beartracks.ualberta.ca displayed

```
webpages      [          ]  
currentIndex  [  ]
```

---

## Deliverables

You will produce and submit one file for this lab on eClass by the submission deadline:

- **lab4\_browser.py**: Python solution to Part 2 of this lab

*(Note: Part 1 should not be submitted, it is for your practice only)*