

## CSC 104 Assignment 3, Winter 2008


Due by the end of Thursday March 20, 2008; no late assignments without written explanation.

### Part 1: Write some web pages in simple HTML

For this part of the assignment you need to write two web pages. One page, with file name `web.html`, should look like this:

*Surfing* The *World-Wide Web*

*W*hen you surf the web, you hope to catch that perfect wave! It might just look like this:



If you fall off your board, you may have to [run a new web browser](#) and start surfing all over again.

You may use any (inoffensive) picture you like for the picture of a wave. Perhaps find something interesting (but small) on the web.

You can use the following images with an `<img>` tag for your wave or for fancy text:

- The wave picture: `http://www.cs.toronto.edu/~faye/pics/bigWaves.gif`
- The fancy *W*: `http://www.cs.toronto.edu/~faye/pics/w.gif`
- The *Surfing*: `http://www.cs.toronto.edu/~faye/pics/surfing.gif`

Be sure to use suitable 'alt' text.

The text “[run a new web browser](#)” is a link to another page in a file named “`browser.html`”.

## CSC 104 Assignment 3, Winter 2008

The `browser.html` file should contain instructions about how to run a web browser at CDF (probably just one sentence), and also a link back to the `web.html` page.

In your HTML files, use **only structural tags**, not tags which directly specify the desired appearance. Structural tags include `<html>`, `<head>`, `<body>`, `<title>`, `<h1>`, `<p>`, `<ol>`, `<ul>`, `<sup>`, etc.

Appearance tags which you should not use include `<b>`, `<font>`, `<br>`.

(An exception, not relevant here, is that you could use `<i>...</i>` to indicate the *English language* use of italics, as traditionally used in prose, such as to indicate stress, or to indicate terminology or foreign-language words. For typographic emphasis, you would use `<strong>...</strong>`; for citations you would use `<cite>...</cite>`; etc.)

All applicable tags should be properly paired. Most tags we are using should be closed, except for `<p>`, `<li>`, and `<img>`.

All of the above means that you may *not* use a “web authoring” program.

You must compose the HTML yourself and you will be graded on the actual HTML file, including how it looks when viewed as plain text (e.g. with “cat”).

### Part 2: Publish your web pages through your CDF account

To publish these two files on the web through the CDF web server, make a subdirectory of your home directory called “`public_html`” and put them in that subdirectory. This allows you to view the file using the URL `http://www.cdf.toronto.edu/~yourname/filename`.

However, the web server will not be able to read these files yet, because your home directory is set up by default to be unreadable by other users. The web server runs with user permissions, under the special user name “`www`”.

So part 2 of this assignment is to modify your file permissions as needed to publish your web pages.

First, examine the output of the `ls` command when you give it the option “`-l`” (the letter ell, for “long” format output). The various fields will be described on a CSC 104 web page, and changing them with the `chmod` command will be described on a web page and/or in lecture or tutorial.

If you specify a directory on the `ls` command line, it shows you the files *inside* that directory. If you want the information about the directory instead, use the “`-d`” option in addition to the “`-l`” option. For example, you can see the “`ls -l`” information for the current directory by typing “`ls -l -d`”.

To make your home directory accessible to the web server, you will want to give it global ‘x’ permission by typing “`chmod a+x .`” when you are in your home directory. The `public_html` directory will similarly need to be accessible to the web server; type “`chmod a+x public_html`”. For plain files (the HTML files inside that directory), to make them readable by others you should use “`chmod a+r`”.

If you use the URL `http://www.cdf.toronto.edu/~yourname/`, you get a directory listing of your `public_html` directory, by default; but you can instead, if you like, supply an HTML file called “`index.html`” which will be sent in response to this URL. You don’t have to make an `index.html` file for this assignment.

## CSC 104 Assignment 3, Winter 2008

### Part 3: Python functions

The following python functions should reinforce your skill at using logic to control program flow. As before, you must create your functions in a single python file called `a3.py`. You must test your functions in a separate file `a3Test.py`.

Function	Specifications																												
<code>calcGrade(numericGrade)</code>	<p>Accept a number <code>numericGrade</code> which represents a numeric grade. Convert the numeric grade to an alphabetic grade using the following criteria:</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>lower threshold</th> <th>grade</th> </tr> </thead> <tbody> <tr><td>90</td><td>A+</td></tr> <tr><td>85</td><td>A</td></tr> <tr><td>80</td><td>A-</td></tr> <tr><td>77</td><td>B+</td></tr> <tr><td>73</td><td>B</td></tr> <tr><td>70</td><td>B-</td></tr> <tr><td>67</td><td>C+</td></tr> <tr><td>63</td><td>C</td></tr> <tr><td>60</td><td>C-</td></tr> <tr><td>57</td><td>D+</td></tr> <tr><td>53</td><td>D</td></tr> <tr><td>50</td><td>D-</td></tr> <tr><td>0</td><td>F</td></tr> </tbody> </table> <p>Return the alphabetic grade value calculated.</p>	lower threshold	grade	90	A+	85	A	80	A-	77	B+	73	B	70	B-	67	C+	63	C	60	C-	57	D+	53	D	50	D-	0	F
lower threshold	grade																												
90	A+																												
85	A																												
80	A-																												
77	B+																												
73	B																												
70	B-																												
67	C+																												
63	C																												
60	C-																												
57	D+																												
53	D																												
50	D-																												
0	F																												
<code>min3(val1, val2, val3)</code>	Accept three numbers: <code>val1</code> , <code>val2</code> , <code>val3</code> , and return the smallest value.																												
<code>calculator(val1, operator, val2)</code>	Accept two numeric values <code>val1</code> and <code>val2</code> , and an <code>operator</code> which may be one of: '+' (plus), '-' (minus), '*' (times) or '/' (divided by). Perform the operation specified ( <code>val1 operator-operation val2</code> ) and return the result of the calculation.																												
<code>average(numberList)</code>	Accept a list of numbers <code>numberList</code> . Calculate and return the average value of the numbers in the list. (use real numbers)																												
<code>longestWord(wordlist)</code>	Accept a list of words <code>wordlist</code> . Return the longest word in the list.																												

## CSC 104 Assignment 3, Winter 2008

### Submission

For part 1, you must make the HTML files work at the following URLs, replacing c7abcdef with your CDF logname:

```
http://www.cdf.toronto.edu/~c7abcdef/web.html
```

```
http://www.cdf.toronto.edu/~c7abcdef/browser.html
```

You may also, if you like, submit these two HTML files using the ‘submit’ command. I recommend this only if you are having trouble making them visible on the web.

**Important note:** If you submit the files using the ‘submit’ command, then those versions will take precedence over the web versions, and the web versions will only be consulted for visibility.

So make sure you re-submit them after making any changes. It’s best just to make them directly available over the web and not to ‘submit’ them at all.

Part 3 must be submitted with the ‘submit’ command. The file names must be **a3.py** and **a3Test.py**.

The submission commands are similar to those for assignments one and two, except that you will use “-a a3” where you used “-a a1” and “-a a2”. Also remember the “-f” option for resubmission, if needed.

### Hints and other questions and answers

Please watch the course web page for a “questions & answers” page, once the questions start coming in.