For this lab I'll leave X-windows on and you can use multiple terminal windows.

Everything in this lab should be done in your directory called webdocs.

Part 1 - Making a Web Page

We'll make an html page. Cd to your webdocs directory. You should have a file called index.html. It's probably not readable. Use chmod to make it readable by everyone.

Then point a browser at sites.mathcs.wilkes.edu/ user, where user is your username. You should see the page. It's extremely vanilla. Edit index.html to surround the page with:

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>

  ... Contents of your page

  </body>
</html>
```

Add whatever you would like to the page.

Save it and reload the page in the browser.

Turn this one in with the turnin name webpage

Part 2 - Making HTML Tables

Now your will Write a shell script that reads data with fields in columns from standard input and produces an html table.

In html, tables look like this:

```
  <table>
  <tr><td>Field 1</td><td>Field 2></td><td>Field 3</td></tr>
  <tr><td>Field 1</td><td>Field 2></td><td>Field 3</td></tr>
  <tr><td>Field 1</td><td>Field 2></td><td>Field 3</td></tr>
  <tr><td>Field 1</td><td>Field 2></td><td>Field 3</td></tr>
  </table>
```

Items are "marked up" with tags enclosed in angle brackets. Each tag should be closed with a corresponding tag that has an angle bracket followed by a slash.

The `<table>` tag starts a table and `</table>` ends it.

Table rows are indicated with the `tr` tag.

Each item on a row is indicated with the `td` tag.

You can write a loop that reads line by line like this:

```
while read line
do

...

done
```

`line` is a variable.

The input for this script will be a file with fields separated by colons. Like this:
field1:field2:field3

Use echo to print the open and close table tags.

For each line, use echo to print the open and close tr tags.

Use variable substitution to replace colons by `</td><td>`.

Reminder: variable substitution works like this: `${v//abc/def}` will replace abc by def in v. You'll also need to echo an open and close td tags on either end.

Call your script maketable1.sh

You can copy this test file: /home/mathcs/courses/cs246/people to your directory and test your script on it.

Part 3 - Another version of Making Tables

Make another script called maketable2.sh that uses sed and command substitution to replace the colons.

```
echo -n "$line" | sed -e 's/:/<\/td><td>/g'
```

Part 4 - Renaming Files

Write a script called replaceext.sh that takes two command line arguments that are file extensions (for example .jpg and .png). It will rename each file in the current directory that has the first extension, replacing it with the second extension.

Your script must check for two command line arguments, print a usage message, and exit if they aren't present.

To loop over all files in the current directory, you can use

```
for f in *
```

To remove the file extension from a name stored in `$f`, use `${f%$ext}` assuming the extension name is stored in `$ext`.