Dr. Mark Humphrys

School of Computing. Dublin City University.

Einstein - overloaded pages

Find web pages (among my local files on disk, not remote pages) that are overloaded with too many / too large embedded images. (Slowly-loading pages.)

---

totalimg

Write this script:

```
totalimg file.html
```

Add up total size of all embedded images in this HTML file.

Notes

Embedded images look like:

```
<img src="filename">
<img ....  src="filename">
<img width=...  src="filename">
<img style=...  src="filename">
<img ....  src="filename" .... >
```

To test it we will run it on pages in my test suite:

```
cd /users/tutors/mhumphrysdculab/share/testsuite
```

Assumptions

- We assume we are in the same directory as the file. (This is important, since a HTML link might be a relative link not an absolute link.)
- So we run it like this:

```
$ cd /users/tutors/mhumphrysdculab/share/testsuite/Cashel
$ totalimg george.html
```

- We assume `img` and `src` are on the same line.
- We assume `src` is lowercase.
- We assume image names are surrounded by double quotes.

- You could write a more robust script to allow for HTML that deviates from the above, but please do not. Or at least, do not submit that script to Einstein, or its marking might get confused. Follow the recipe below for the script you submit to Einstein.

Recipe

1. The filename is a command line argument.
2. grep the file for lines with an embedded image.
3. Put newlines **before and after every HTML tag**.
4. grep again for embedded images.
5. Use grep to *get rid of* lines with 'http'

You should now just have a list of embedded local images, like this:

```
$ cd /users/tutors/mhumphrysdculab/share/testsuite/Cashel
$ totalimg george.html

<img border=0 src="../Icons/pdf.gif">
<img border=0 src="../Icons/pdf.gif">
<img src="Bitmaps/ric.crop.2.jpg">
<img src="../Icons/me.gif">
<img width="98%" src="../Kickham/08.Mullinahone/SA400010.small.JPG">
<img width="98%" src="../Kickham/08.Mullinahone/SA400028.small.JPG">
<img width="95%" src="07.Carlow.Stn/SA400069.lores.jpg">
<img border=1  width="95%" src="07.Carlow.Stn/SA400070.lores.adjust.jpg">
```

Pipe the above into further commands to extract the image file names.

1. Use **sed** to delete everything from **start-of-line** to `src="`
2. Use sed to delete everything from `"` to end-of-line.

You should now have a better list of local images, like this:

```
$ cd /users/tutors/mhumphrysdculab/share/testsuite/Cashel
$ totalimg george.html

../Icons/pdf.gif
../Icons/pdf.gif
Bitmaps/ric.crop.2.jpg
../Icons/me.gif
../Kickham/08.Mullinahone/SA400010.small.JPG
../Kickham/08.Mullinahone/SA400028.small.JPG
07.Carlow.Stn/SA400069.lores.jpg
07.Carlow.Stn/SA400070.lores.adjust.jpg
```

Do files exist, and get sizes

1. Some of the files (like Kickham) do not actually exist.
2. So pipe the previous into a **Shell function** which will see if the files exist, and add up the file sizes.
3. Start with the following as the shell function. This just lists the files:

```
while read file
do
 if test -f $file
 then
  ls -l $file
 fi
done
```

4. Check this works before proceeding. Something like:

```
$ cd /users/tutors/mhumphrysdculab/share/testsuite/Cashel
$ totalimg george.html

-rwxr-xr-x 1 mhumphrysdculab tutors 426 Sep 17  2015 ../Icons/pdf.gif
-rwxr-xr-x 1 mhumphrysdculab tutors 426 Sep 17  2015 ../Icons/pdf.gif
-rwxr-xr-x 1 mhumphrysdculab tutors 39139 Sep 17  2015 Bitmaps/ric.crop.2.jpg
-rwxr-xr-x 1 mhumphrysdculab tutors 1005 Sep 17  2015 ../Icons/me.gif
-rwxr-xr-x 1 mhumphrysdculab tutors 339817 Sep 17  2015 07.Carlow.Stn/SA400069.lores.jpg
-rwxr-xr-x 1 mhumphrysdculab tutors 190968 Sep 17  2015 07.Carlow.Stn/SA400070.lores.adjust.jpg
```

5. (Note we have removed the files that do not exist.)
6. Now delete the `ls` line and insert:

```
stat --printf="%s" $file
echo
```

7. This prints the file size, plus new line.
8. Check this works before proceeding. Something like:

```
$ cd /users/tutors/mhumphrysdculab/share/testsuite/Cashel
$ totalimg george.html

426
426
39139
1005
339817
190968
```

Finish

1. Pipe the above to another Shell function which looks like this:

```
TOTAL=0

while read size
do
 TOTAL=`expr $TOTAL + $size`
done

echo "$TOTAL"
```

Test

Your finished script should work like this:

```
$ cd /users/tutors/mhumphrysdculab/share/testsuite/Cashel
$ totalimg george.html
571781

$ totalimg bushfield.html
3274461

$ cd /users/tutors/mhumphrysdculab/share/testsuite/ORahilly
$ totalimg the.orahilly.note.html
2515730

$ totalimg ballylongford.html
1654649
```

Imagine using this script to search thousands of pages for the most overloaded pages.

Upload to Einstein

- Rename it to totalimg.sh