How to do it...

Let's write a Bash script by combining different commands, to track changes in a web page:

```
#!/bin/bash
#Filename: change_track.sh
#Desc: Script to track changes to webpage

if [ $# -ne 1 ];
then
  echo -e "$Usage: $0 URL\n"
  exit 1;
fi

first_time=0
# Not first time

if [ ! -e "last.html" ];
then
  first_time=1
  # Set it is first time run
fi

curl --silent $1 -o recent.html

if [ $first_time -ne 1 ];
then
  changes=$(diff -u last.html recent.html)
  if [ -n "$changes" ];
  then
    echo -e "Changes:\n"
    echo "$changes"
  else
    echo -e "\nWebsite has no changes"
  fi
else
  echo "[First run] Archiving.."

fi


cp recent.html last.html
```

Let's look at the output of the `track_changes.sh` script when changes are made to the webpage and when the changes are not made to the page:

- First, run the following command:

```
$ ./track_changes.sh http://web.sarathlakshman.info/test.html
[First run] Archiving..
```

- Second, run the following command:

```
$ ./track_changes.sh http://web.sarathlakshman.info/test.html
Website has no changes
```

- Third, run the following command after making changes to the web page:

```
$ ./test.sh http://web.sarathlakshman.info/test_change/test.html
Changes:

--- last.html 2010-08-01 07:29:15.000000000 +0200
+++ recent.html     2010-08-01 07:29:43.000000000 +0200
@@ -1,3 +1,4 @@
<html>
+added line :)
<p>data</p>
</html>
```

How it works...

The script checks whether the script is running for the first time by using `[ ! -e "last.html" ];`. If `last.html` doesn't exist, it means that it is the first time and, hence, the webpage must be downloaded and copied as `last.html`.

If it is not the first time, it should download the new copy (`recent.html`) and check the difference by using the `diff` utility. If there are changes, it should print the changes and finally, copy `recent.html` to `last.html`.

See also

- The *A primer on cURL* recipe in this chapter explains the `curl` command