We can restrict the speed limits in `wget` by using the `--limit-rate` argument as follows:

```
$ wget  --limit-rate 20k http://example.com/file.iso
```

In this command, `k` (kilobyte) specifies the speed limit. You can also use `m` for megabyte.

We can also specify the maximum quota for the download. It will stop when the quota is exceeded. It is useful when downloading multiple files, limited by the total download size. This is useful to prevent the download from accidently using too much disk space.

Use `--quota` or `-Q` as follows:

```
$ wget -Q 100m http://example.com/file1 http://example.com/file2
```

Resume downloading and continue

If a download using `wget` gets interrupted before it is complete, we can resume the download from where we left off by using the `-c` option as follows:

```
$ wget -c URL
```

Copying a complete website (mirroring)

`wget` has an option to download the complete website by recursively collecting all the URL links in the web pages and downloading all of them like a crawler. Hence, we can completely download all the pages of a website.

In order to download the pages, use the `--mirror` option as follows:

```
$ wget --mirror --convert-links exampledomain.com
```

Or use the following command:

```
$ wget -r -N -l -k DEPTH URL
```

`-l` specifies the depth of web pages as levels. This means it will traverse only that many number of levels. It is used along with `-r` (recursive). The `-N` argument is used to enable time stamping for the file. `URL` is the base URL for a website for which the download needs to be initiated. The -k or `--convert-links` option instructs `wget` to convert the links to other pages in a downloaded page, to the local copy of those pages.

Exercise discretion when mirroring other websites. Unless you have permission, only perform this for your personal use and don't do it too frequently.

Accessing pages with HTTP or FTP authentication

Some web pages require authentication for HTTP or FTP URLs. It can be obtained by using the `--user` and `--password` arguments:

```
$ wget --user username --password pass URL
```

It is also possible to ask for a password without specifying the password inline. For this, use `--ask-password` instead of the `--password` argument.

Downloading a web page as plain text

Web pages are HTML pages that contain a collection of HTML tags, along with other elements, such as JavaScript and CSS. Of these, the HTML tags define the content of a web page, which we can parse to look for a specific content, and this is something Bash scripting can help us with. When we download a web page, we receive an HTML file, and in order to view the formatted page, it should be viewed in a web browser.

In most of the circumstances, parsing a text document will be easier than parsing HTML data because we aren't required to strip off the HTML tags. **Lynx** is an interesting command-line web browser, which can get the web page as plaintext. Let us see how to do it.

How to do it...

Let's download the webpage view, in ASCII character representation, in a text file by using the `-dump` flag with the `lynx` command:

```
$ lynx URL -dump > webpage_as_text.txt
```

This command will also list all the hyperlinks (`<a href="link">`) separately under a heading `References`, as the footer of the text output. This will help us avoid parsing of links separately by using regular expressions.

For example:

```
$lynx -dump http://google.com > plain_text_page.txt
```

You can see the plaintext version of `text` by using the `cat` command, as follows:

```
$ cat plain_text_page.txt
```