Wget: download whole or parts of websites with ease

wget is a nice tool for downloading resources from the internet. It can be used to fetch images, web pages or entire websites. It can be used with just a URL as an argument or many arguments if you need to fake the user-agent, ignore robots.txt files, rate limit it or otherwise tweak it.

The basic way to use it is `wget url`: `wget https://example.org/`

Therefore, wget and less is all you need to surf the internet.

Naming the output file with -O

WGet's `-O` option for specifying output file is one you will use *a lot*. Let's say you want to download an image named 2039840982439.jpg. That is not very useful. Thus; you could ask wget to name the saved file something useful,

```
wget https://example.org/iamges/2039840982439.jpg -O
images/anime-girls-with-questionmarks/cute-blond-girl.jpg
```

Downloading recursively

The power of wget is that you may download sites recursive, meaning you also get all pages (and images and other data) linked on the front page:

```
wget -r https://example.org//
```

But many sites do not want you to download their entire site. To prevent this they typically check how browsers identify. Many sites refuses you to connect or sends a blank page if they detect you are not using a web-browser. You might get a message like:

Sorry, but the download manager you are using to view this site is not supported. We do not support use of such download managers as flashget, go!zilla, or getright

The trick to ignoring sites blacklisting wget in robots.txt

Many sites have a `robots.txt` file which includes `wget`.

`wget` *will respect* a listing in a `robots.txt` file which tells `wget` to not download parts of a website or anything at all if that is what the `robots.txt`file asks. `wget` will respect `robots.txt` even if you override the user-agent.

The command-line option `-e robots=off` will tell `wget` to ignore the `robots.txt` file.

The trick that fools sites and webservers blocking Wget by User-Agent

Wget has a very handy -U option for sites that don't like wget. Use -U My-browser to tell the site you are using some commonly accepted browser:

```
wget -r -p -U Mozilla http://www.example.com/restricedplace.html
```

You will, of course, want to use a complete string which looks plausible for `-U such as`:

```
wget -U "Mozilla/5.0 (iPhone; CPU iPhone OS 12_3 like Mac OS X)
AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.1.1
Mobile/15E148 Safari/604.1"
```

This can actually be quite useful for testing how well your mobile code adaptation code works with various user-agents[1]

It is possible to make an *alias* in your `$HOME/.bashrc` which will always use a given user-agent,

File: $HOME/.bashrc

```
alias wgetmobile='wget -U "Mozilla/5.0 (iPhone; CPU iPhone OS 12_3 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.1.1 Mobile/15E148 Safari/604.1"'
```

Be polite!

The most important command line options for being polite with wget are `--limit-rate=` and `--wait=`. You *should* add `--wait=20` to pause 20 seconds between retrievals - this ensures that you are not manually added to a blacklist. `--limit-rate` defaults to bytes, add K to set KB/s. Example:

```
wget --wait=20 --limit-rate=20K -r -p -U Mozilla
http://www.example.com/restricedplace.html
```

A web-site owner will probably get upset if you attempt to download their entire site using a simple `wget http://foo.bar` command and it is very noticeable in the logs too. However, the web-site owner will not even notice you if you limit the download transfer rate and pause 20 seconds between fetching files.

Use --no-parent

--no-parent is a very handy option that guarantees wget will not download anything from the folders beneath the folder you want to acquire. Use this to make sure wget does not fetch more than it needs to if just just want to download the files in a folder.