More key Git techniques

There are a few more key Git commands and techniques that you need to know about.

.gitignore

When you're working with Git, there are lots of things you won't want to push, including:

- hidden system files
- .pyc files
- rendered documentation files
- ... and many more

`.gitignore`, https://github.com/evildmp/afraid-to-commit/blob/master/.gitignore, is what controls this. You should have a `.gitignore` in your projects, and they should reflect *your* way of working. Mine include the things that my operating system and tools throw into the repository; you'll find soon enough what yours are.

With a good `.gitignore`, you can do things like:

```
git add docs/
```

and add whole directories at a time without worrying about including unwanted files.

Starting a new Git project

You've been working so far with an existing Git project. It's very easy to start a brand new project, or turn an existing one into a Git project. On GitHub, just hit the **New repository** button and follow the instructions.

Combining Git and pip

When you used pip to install a package inside a virtualenv, it put it on your Python path, that is, in the virtualenv's `site-packages` directory. When you're actually working on a package, that's not so convenient - a Git project is the most handy thing to have.

On the other hand, cloning a Git repository doesn't install it on your Python path (assuming that it's a Python application), so though you can work on it, you can't actually use it and test it as an installed package.

However, pip is Git-aware, and can install packages *and* put them in a convenient place for editing - so you can get both:

```
cd ~/
virtualenv git-pip-test
source git-pip-test/bin/activate
pip install -e git+git@github.com:python-parsley/parsley.git#egg=parsley
```

The `-e` flag means editable; `git+` tells it to use the Git protocol; `#egg=parsley` tells it what to call it.

(Should you find that this causes an error, try using quotes around the target:

```
pip install -e "git+git@github.com:python-parsley/parsley.git#egg=parsley"
```

)

You can also specify the branch:

```
pip install -e git+git@github.com:python-parsley/parsley.git@master#egg=parsley
```

And now you will find an editable Git repository installed at:

    ~/git-pip-test/src/parsley

which is where any other similarly-installed packages will be, and just to prove that it really is installed:

```
$ pip freeze
-e git+git@github.com:python-
parsley/parsley.git@e58c0c6d67142bf3ceb6eceffd50cf0f8dae9da1#egg=Parsley-master
wsgiref==0.1.2
```