ATAD | Basic GitHub CLI usage

This tutorial shows the basic *git* commands over the CLI (command line interface). However, *VS Code* has a very competent *git integration*; documentation at code.visualstudio.com.

This document does not cover *pull requests* nor *issues*.

Default credentials for git

This is not a required step, as you will be prompted for your credentials everytime they are necessary; either on the CLI or VS Code.

But if you use a single *GitHub* account it may be preferred as it will not ask for your username later on (password is always asked):

The values of `user.name` and `user.email` **must match** your *GitHub* information:

```
$> git config --global user.name "brunomnsilva-estsetubal"
$> git config --global user.email "bruno.silva@estsetubal.ips.pt"
```

> Apparently, starting from August 21st *GitHub* will require *token authentication*. If needed, check github.blog for instructions.

Clone from upstream

This enables you to *clone* the upstream repository into your local filesystem.

> If you are not the owner of the upstream repository you'll not be able to *commit* and *push* changes.

The *repository url* is available through the **Code** green button in the project's *GitHub* page.

```
$> git clone https://github.com/estsetubal-atad/CProgram_Template.git
 ↪  MyCProgram
$> cd MyCProgram
```

Here, `MyCProgram` is the diretory that will be created with the repository code. If you ommit this argument, then the directory will be named after the repository name.

Fork a repository at GitHub

1. Navigate to the repository webpage and *fork* the project into your own account (requires *sign in*).
2. You'll now have a *forked* repository that you own and can commit changes to.
3. Use the "Clone from upstream" instructions.

Start a new Git repository for an existing code base

It should be pointed out that your project **should** have a `README.md` *markdown* file with the repository description, if you intend to go public with it.

```
$> cd /path/to/my/codebase
$> git init      (1)
$> git add .     (2)
$> git commit -m "Initial commit"   (3)
```

1. Create a `/path/to/my/codebase/.git` directory.
2. Add all existing files to the index.
3. Record the pristine state as the first commit in the history.

Upload to github

First you need to create a *git repository* at GitHub, from where you'll get the repository URL to use below:

```
$> git remote add origin
 ↪  https://github.com/estsetubal-atad/CProgram_Template.git   (1)
$> git branch -M master        (2)
$> git push -u origin master   (3)
```

1. Set your *remote* repository URL.
2. Move code to `master` *branch*.
3. Push code to *remote* `master` *branch*.

Commit and push existing repository

You can commit and push changes to any repository that you own by using:

```
$> git add .
$> git commit -m "Commit message"
$> git push -u origin master
```

you should replace “Commit message” with a brief description of the content of your *commit*. This also assumes that the *remote branch* is called `master`.

Author and support

Bruno Silva (bruno.silva@estsetubal.ips.pt)

You should ask your PL teacher for any help regarding these contents and procedures.