Python 01

Elias Wachmann

2024

“The magic of computing begins with 0, a simple binary digit that serves as a powerful reminder that even the smallest building blocks can create wonders.”

Content

Introduction

Starting to code

This course is designed to give you a basic understanding of the Python programming language. But to get you all up to speed we will first have a look at the Tools you need to get started...

Tools for the course

- Python - python.org/downloads
- VsCode - code.visualstudio.com/download
- Git - git-scm.com/downloads
- Gitlab - see Mail - personal account

Python & VsCode

Python in VsCode

Ok, so you have installed Python and VsCode.

Let’s take a look ...

Git & GitLab

Git [video]

Git is a version control system.

Version control system: A system that records changes to a file or set of files over time so that you can recall specific versions later.

But why should I use it?
No more `report_v1_final_final_final_final.pdf`!

You can download git here.

Git – initializes a new repository (init)

Repository: A directory where git has been initialized to start version controlling your files.

To initializes a new repository use `git init`.

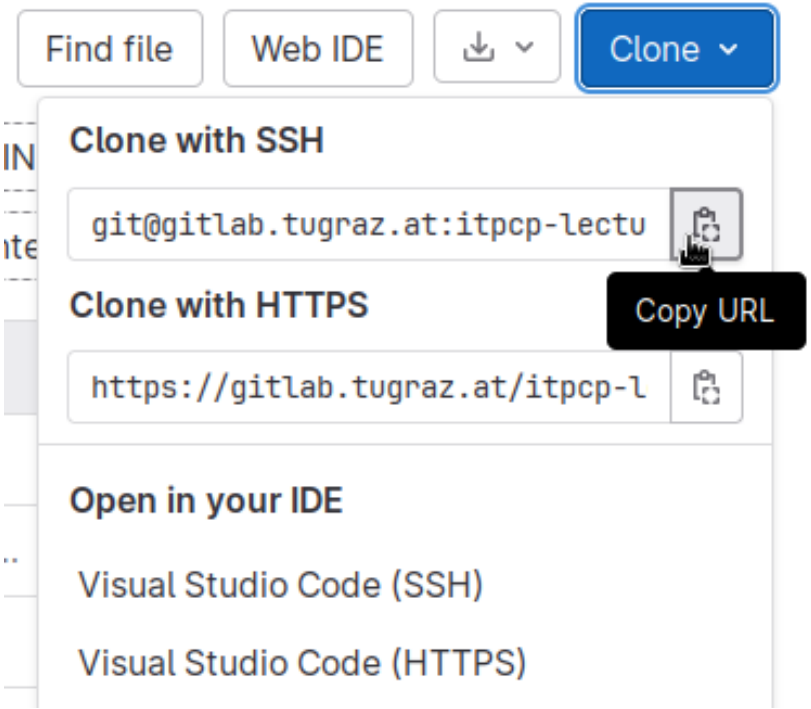
```
etschgi1@Deep-thought:/mnt/c/REPOS/example_git$ git init
Initialized empty Git repository in /mnt/c/REPOS/example_git/.git/
etschgi1@Deep-thought:/mnt/c/REPOS/example_git$
```

This creates a hidden folder `.git` which contains all the information about the repository.

Git – clone a repository (clone)

Navigate to your Gitlab repository and copy the url.

Click on the `clone` button and copy the `Clone with SSH` url.



Git – clone a repository (clone) cont.

To clone a repository use `git clone` followed by the `url` in the shell.

Or open the VsCode palette usind `Ctrl+Shift+P` and type `Git: Clone` followed by the `url`.

Choose a directory to clone the repository into.

Note: The repository is now available on your local machine.

Git – add files (add)

To add files to the repository use `git add`.
This stages the files for the next commit.

```
etschgi1@Deep-thought:/mnt/c/REPOS/example_git$ echo "Hello World" > testfile.txt
etschgi1@Deep-thought:/mnt/c/REPOS/example_git$ git add testfile.txt
etschgi1@Deep-thought:/mnt/c/REPOS/example_git$ git add .
etschgi1@Deep-thought:/mnt/c/REPOS/example_git$
```

Here ”Hello World” is redirected into a file called `testfile.txt` using the > operator.
All files in the current folder and in subfolder bellow can be added using `git add .` (with a fullstop).

Git – see current status (status)

To see the current status of the repository use `git status`.

```
etschgi1@Deep-thought:/mnt/c/REPOS/example_git$ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   testfile.txt

etschgi1@Deep-thought:/mnt/c/REPOS/example_git$
```

`testfile.txt` is staged for commit.

Git – commit changes (commit)

To commit the changes use `git commit`.

```
etschgi1@Deep-thought:/mnt/c/REPOS/example_git$ git commit -m "This is my message"
Author identity unknown

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: empty ident name (for <etschgi1@Deep-thought.localdomain>) not allowed
etschgi1@Deep-thought:/mnt/c/REPOS/example_git$
```

That didn’t work → setup your git user name & email.

Git – setup user name & email (config)

To setup your user name & email use `git config`.
The `--global` flag sets the configuration (globally) for the current user.

```
etschgi1@Deep-thought:/mnt/c/REPOS/example_git$ git commit -m "This is my message"
[main b71f6fb] This is my message
 1 file changed, 1 insertion(+), 1 deletion(-)
```

After setting up the user name & email you can commit the changes.
Commit messages should be short and meaningful and can be added using the `-m` flag.

Git – see commit history (log)

To see the commit history use `git log`.

```
etschgi1@Deep-thought:/mnt/c/REPOS/example_git$ git log
commit b71f6fb5ecc50bbd65861a10d060fa37deffd5a9 (HEAD -> main)
Author: max huber <e.wachmann@student.tugraz.at>
Date:   Wed Apr 26 16:56:48 2023 +0200

    This is my message
```

Each commit has a unique identifier called a **SHA**.
This allows you to go back to a specific commit.

Git – Remote repositories

Remote repository: A repository that is hosted on the Internet or another network.

It is a good idea to have a remote repository to backup your work or collaborate with others.

GitLab: A website that hosts git repositories (relevant for the exercises later).

Git – Remote (push & pull)

When cloning a repository the remote alias `origin` is automatically added.
All changes are loaded into the local repository.

After working on the project locally you can push the changes to the remote repository using `git push`.

Note: You can also pull changes from a remote repository using `git pull`.

Git – SSH keys?

SSH: Secure Shell (SSH) is a cryptographic network protocol for operating network services securely over an unsecured network.

SSH keys: An SSH key is an access credential in the SSH protocol.

You need to setup SSH keys to push changes to a remote repository.

Use this Guide to setup link your SSH keys to GitLab.

Create your SSH keys

Create the SSH keys using the `ssh-keygen` command:

```
ssh-keygen -t rsa -b 4096 -C "
   your_name@student.tugraz.at"
```

Windows:

```
Enter file in which to save the key (C:\
   Users\your_username/.ssh/id_rsa):
```

Linux:

```
Enter file in which to save the key (/home/
   your_username/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
```

Add your SSH keys to GitLab

Go to your GitLab account and add the SSH keys to your account.

Go to `Profile-Picture (upper left corner)` → `Preferences` → `SSH Keys` → `Add SSH Key`

Copy the **whole** content of the public key file (id_rsa.pub) into the key field.

Choose a title. Leave usage type as is. You don't have to set an expiration date.
Click `Add key` - Now you are ready to push & pull.

Additional Resources

Additional Resources

- Codewars
- Leetcode
- Oh my git!
- Interactive python tutorials
- Learn Python 3
- CS50