Git

Git is the version control software used in the workshop.

Installation

- Go to the following link and choose the correct version for your operating system: https://git-scm.com/downloads.
- Following the download, run the installer as per usual on your machine.
- **Windows**: You may leave all selection widgets at their default values.

Check installation

You can check that git has been succefully installed:

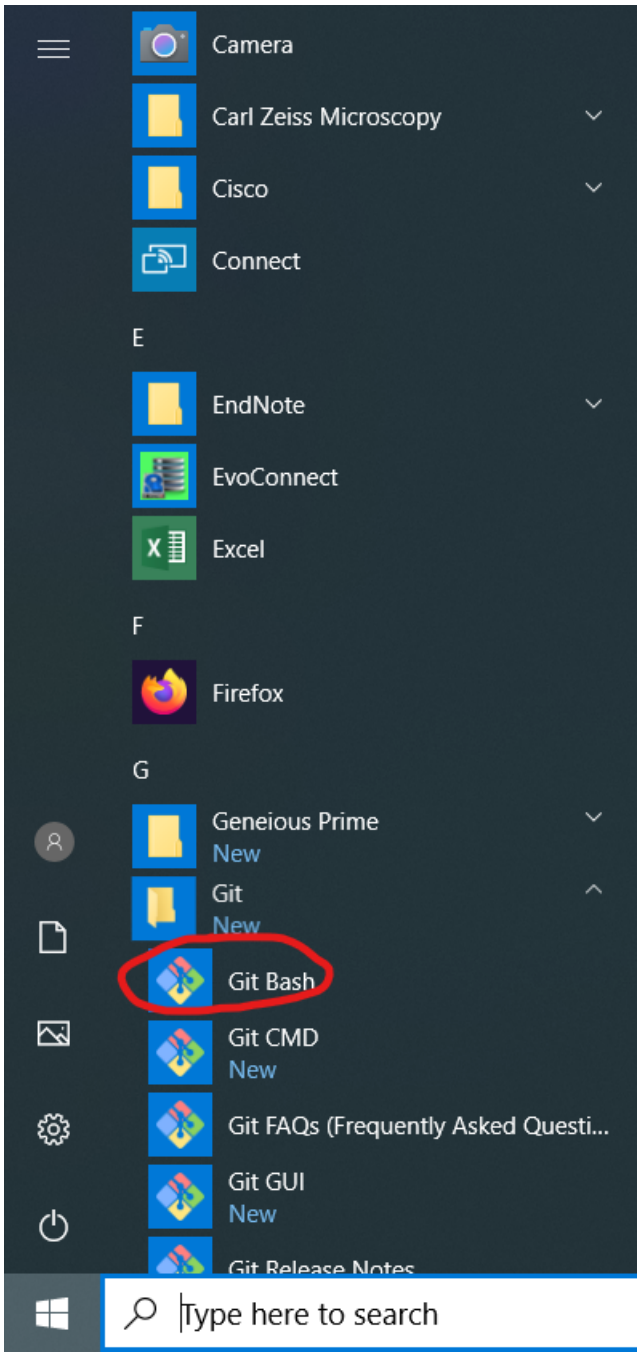
- Open a terminal/command prompt and type `git`. (Hit the `[Enter]` key to terminate the command entry.)

If git is installed then you should see the following:

```
(base) ~/rsc/rsd-workshop(prep-material-day-i ✗) git
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
   clone     Clone a repository into a new directory
   init      Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
   add       Add file contents to the index
   mv        Move or rename a file, a directory, or a symlink
   restore   Restore working tree files
   rm        Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
   bisect    Use binary search to find the commit that introduced a bug
   diff      Show changes between commits, commit and working tree, etc
   grep      Print lines matching a pattern
   log       Show commit logs
   show      Show various types of objects
   status    Show the working tree status

grow, mark and tweak your common history
   branch    List, create, or delete branches
   commit    Record changes to the repository
   merge     Join two or more development histories together
   rebase    Reapply commits on top of another base tip
   reset     Reset current HEAD to the specified state
   switch    Switch branches
   tag       Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)
   fetch     Download objects and refs from another repository
   pull      Fetch from and integrate with another repository or a local branch
   push      Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
See 'git help git' for an overview of the system.
```

Notes for Windows

In your Search bar, type `git-bash` and start the `git-bash` executable. This brings up a terminal window. If you cannot locate the `git-bash` launcher, go to "Start Menu -> Git -> Git Bash". You may want to create a shortcut on the Desktop or add it to the taskbar.


Camera
Carl Zeiss Microscopy
Cisco
Connect
E
EndNote
EvoConnect
Excel
F
Firefox
G
Geneious Prime
New
Git
New
Git Bash
Git CMD
New
Git FAQs (Frequently Asked Questi...
Git GUI
New
Type here to search

Enter

```
$ git
```

and observe the output given above.

Python

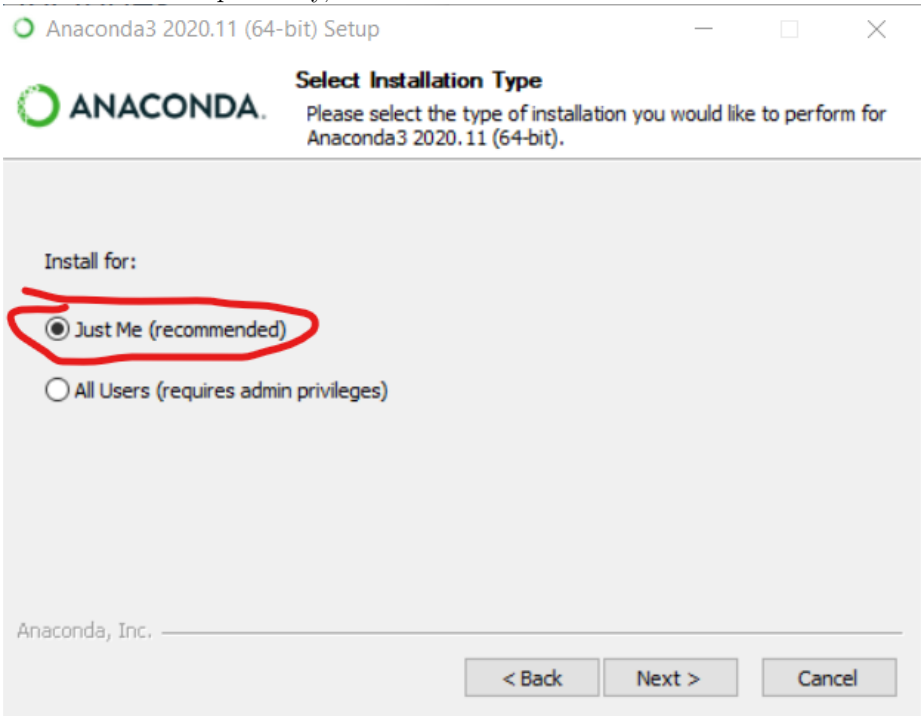
This workshop uses Python as the example programming language, however, the ideas and principles covered extend and apply to any language.

To install Python we will use **Anaconda**. Anaconda is a python distribution (similar as how Ubuntu is a distribution of Linux). It bundles the python interpreter with numerous libraries, addons, and development environments such as jupyter.

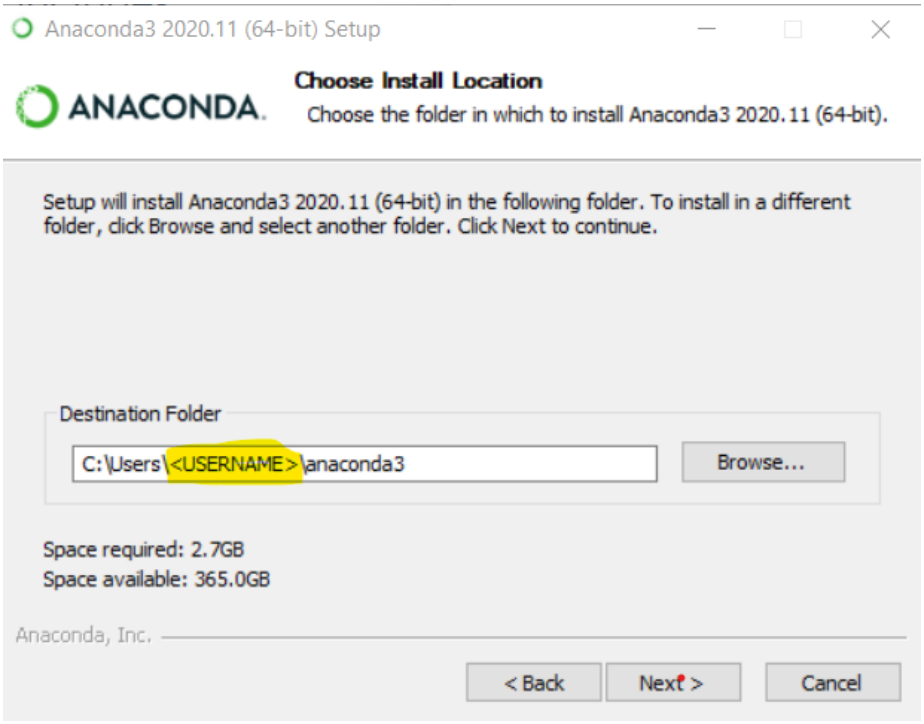
Installation

- Go to the following link https://www.anaconda.com/download/ and select Python 3.
- Following the download, run the installer as per usual on your machine.

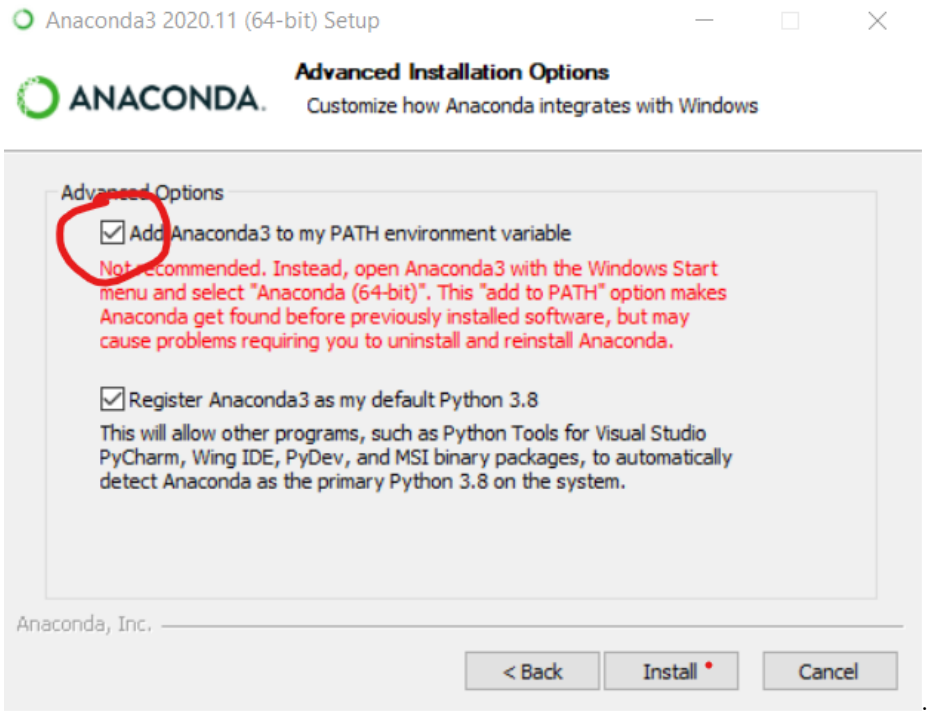
Windows users Below are some screenshots for installing anaconda under windows. Most importantly, select "Install for: Just me".



Next, accept the suggested installation path.



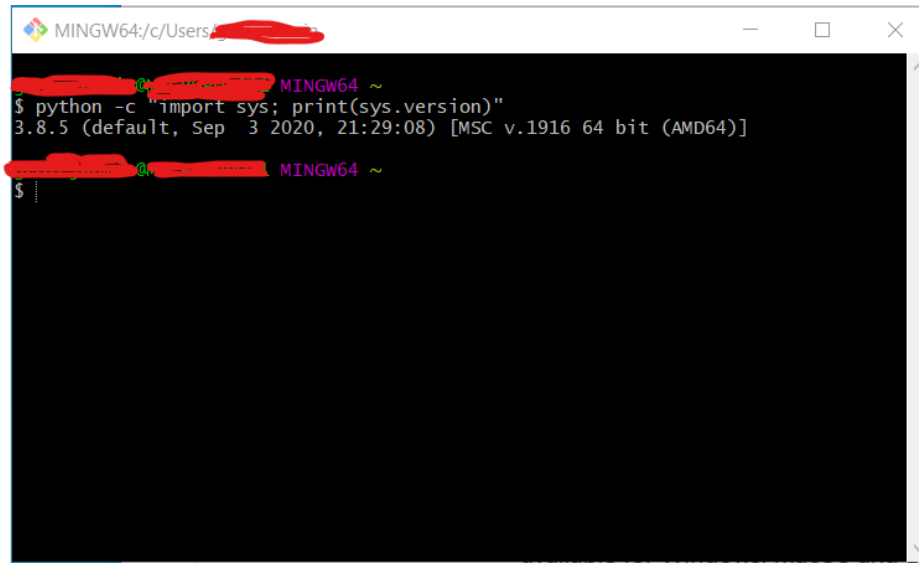
Finally, select "Add Anaconda3 to my PATH environment variable" despite the warning.



Test your python installation

In a terminal window (windows: `git-bash`), enter the command

```
$ python -c "import sys; print(sys.version)"
```



Editor

There are several editors that allow us to edit files such as python scripts, LaTex files etc. For this workhop the suggested editor is VS code.

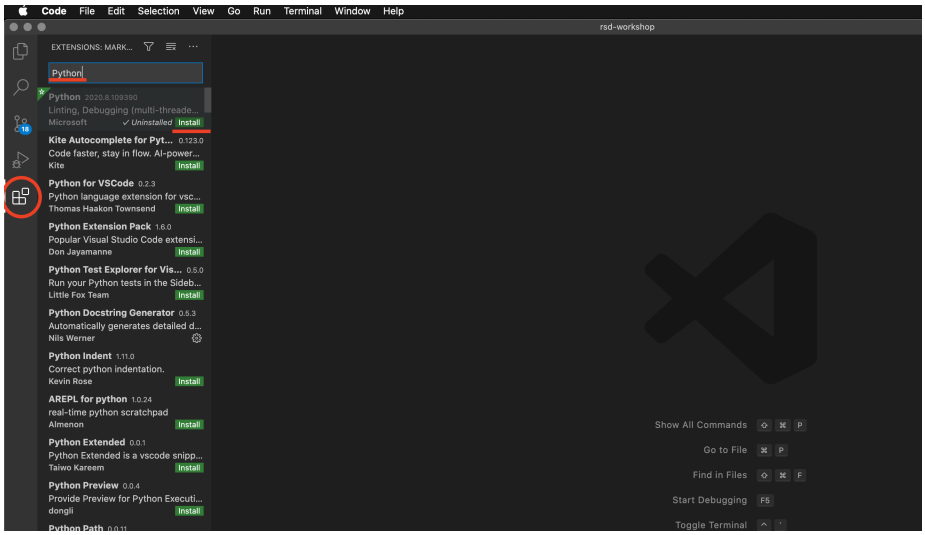
If you prefer to use a different editor skip this part.

Installation

- Go to the following link https://code.visualstudio.com and click download.

Install python extension for VSCode

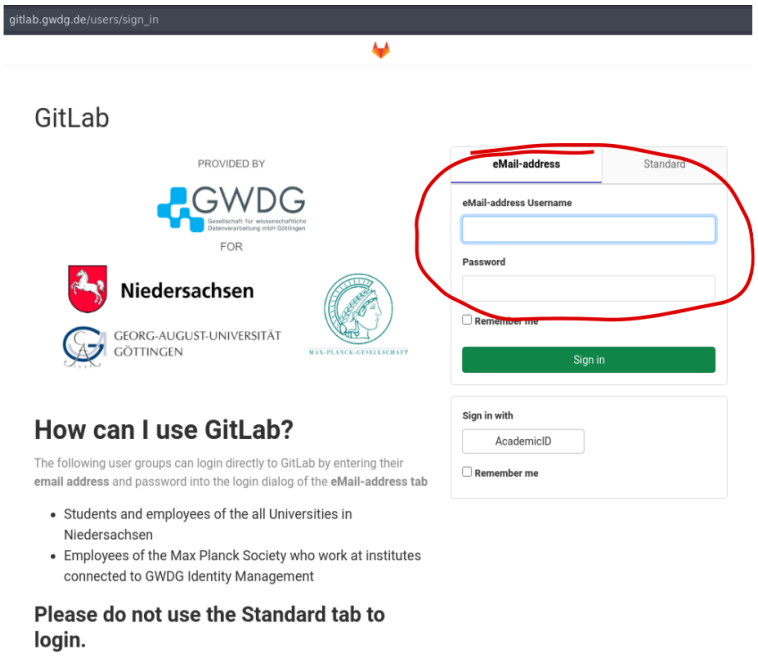
- Open VS code.
- Bring up the Extensions view by clicking on the Extensions icon in the Activity Bar on the side of VS Code.
- There type "Python" and select and install the first hit returned from the search.



GitLab

In this workshop, we will use GitLab, a code sharing service. The GWDG offers free GitLab accounts to all MPI employees. To interact with the code on GitLab, we will use `git`, the version control system.

To login to GitLab, use this website: https://gitlab.gwdg.de/users/sign_in. **Use the tab titled "eMail-address"**, do not use the tab "Standard". Enter your email address and password into the login dialog of the eMail-address tab.



Check list

- ☐ Python
- ☐ Git
- ☐ VS code (or any other editor)
- ☐ GitLab