

Setup Sage and GIT: (for Sage 8.7)

Note that building Sage will take several hours, so get at least as far as Step 5 well in advance of starting work.

1. Make sure you have a trac account or a github account.:

- To request a trac account go to trac.sagemath.org and follow the directions at the top of the page to request an account.
- For github go to <https://github.com/> and sign up for an account.

2. Make a folder in which you will do Sage development and testing:

- You want this to be a place that's different from your regular working Sage installation (if you have one).
- Move into that directory for the rest of the work

3. Set up GIT (only needs to be done one time on your computer): In your terminal:

- Make sure you are in the folder where you want to do Sage development.
- type `git config --global user.name < your name >`
- type `git config --global user.email < your email address >`
- generate a ssh key
 - (1) in the terminal type: `ssh-keygen`
 - (2) then (follow instructions)
- Link ssh key to trac server
 - (1) Go to <http://trac.sagemath.org>
 - (2) Log in with your trac username/password
 - (3) Click on "Preferences"
 - (4) Go to the "SSH Keys" tab
 - (5) Paste the content of your public key file (e.g. `~/.ssh/id_rsa.pub`)
 - (6) Click on "Save changes"
- in the terminal you can check if the key is linked by typing `ssh git@trac.sagemath.org info`

4. Initialize this directory as a git repository:

- In the terminal type: `git init`

5. Get the Sage source code:

This links to the most recent beta (the 'develop' branch). To get the latest released version use 'master' in place of 'develop'.

- type `git remote add trac git://trac.sagemath.org/sage.git -t develop`
if not try:
`git remote add -t develop trac git://trac.sagemath.org/sage.git`
- get the source code: `git pull trac`
- After a minute or two your folder will be synced with the latest source code

6. Compile ("build") Sage:

This links to the most recent beta (the 'develop' branch). To get the latest released version use 'master' in place of 'develop'.

- If this is your first time building sage you'll need to install a number of pre-requisites first. Which ones depends on your operating system. See the developer's guide for the current list: <http://doc.sagemath.org/html/en/installation/source.html#prerequisites>
- type `make` (or for multiple cpus: `make -j#` where # is how many threads to use)
- Go away and do something fun. This is going to take a while.

7. Make a symbolic link to make running sage easier:

- On Ubuntu this can be done as:
`sudo ln -s /<name of sage directory>/sage /usr/bin/<name you want to use>`

8. Get the development scripts to make working with tickets easier:

- Move to your home directory: `cd ~`
- type: `git clone https://github.com/sagemath/git-trac-command.git`
- type: `cd git-trac-command`
- add git-trac to your path:
`sudo ln -s 'pwd'/git-trac /usr/local/bin/` (where 'pwd' is the path to git-trac-command directory)
- type: `git trac config --user=<username> --token=<token>` (get your token from the token tab after logging on to trac)

9. Recommended additional packages:

- `sage -i ccache` to speed up future builds
- To get some of the documentation to build and format correctly, you need latex (on Ubuntu):
`sudo apt-get install texlive`
and possibly the stix fonts
`sudo apt-get install fonts-stix`
and possibly (for pdf docbuild)
`sudo apt-get install texlive-latex-recommended texlive-latex-extra texlive-fonts-recommended`

10. More information:

- You are now ready to create and review tickets. You may want to refer to the Sage Developers' Guide (<http://doc.sagemath.org/html/en/developer/index.html>)
- There is also a GIT/Sage cheat sheet if you forget the commands. In your terminal,
type: `git trac cheat-sheet`
- The `sage-devel` google group can help resolve build failures.

Installing Ubuntu on Windows: (for Sage 8.7)

One option for developing Sage on Windows is to install Ubuntu in a VirtualBox and develop Sage in the Ubuntu environment.

1. Install Virtual Box and set-up a virtual machine:

- install virtual box from <https://www.virtualbox.org>
- You may need to enable hardware virtualization in bios (to get a 64-bit virtual machine)
- choose RAM (4Gb) beware - swap size is equal to RAM unless you manually set partitions for linux. If you want more RAM you can go back and change this later.
- virtual harddrive → Virtual Disk Image → Fixed Size (say: 50 GB)
- Confirm settings → number of cpus (set to one less than the maximum)

2. Install Ubuntu LTS:

- Download Ubuntu LTS .iso <https://www.ubuntu.com>
- select .iso and start virtual machine
- Follow instructions to install ubuntu
- power off virtual machine
- remove .iso from optical drive under machine settings
- restart virtual machine

3. Install Guest Additions:

- in the virtual machine menu Devices → install Guest Additions
- restart virtual machine
- Devices menu → eject Guest Additions.

4. Set-up a shared folder:

- In the virtual box menu: Devices → Shared Folders → Machine Folders → add → auto-mount, make permanent
- in the terminal of the virtual machine `sudo adduser <name> vboxsf` (for permission to access folder)

5. Allow copy/paste:

- in the virtual box setting machines: settings → general → advanced. Make both “bidirectional”

Miscellaneous software for/on Ubuntu: (for Sage 8.7)

1. Install Sublime for code editing:

- `wget -q0 - https://download.sublimetext.com/sublimehq-pub.gpg`
| `sudo apt-key add`
- `sudo apt-add-repository "deb https://download.sublimetext.com/apt/stable/"`
- `sudo apt install sublime-text`
- tools → install package control
- preferences → package control → package control: install packages
- search for python/cython
- View → Syntax → Python 3 (on each file of new type)
- search for outline (follow directions to enable) (ctrl + shift + P)
- preferences → settings. On the right side enter

```

    ,
    "draw_white_space": "all"

```

2. Utility functions:

- (1) runsnake
- (2) importchecker
- (3) coverage
- (4) diff-so-fancy (install with npm)

In `~/ .gitconfig`

[pager]

```

diff = diff-so-fancy | less --tabs=1,5 -RFX
show = diff-so-fancy | less --tabs=1,5 -RFX

```

[color "diff"]

```

meta = "yellow bold"
commit = "green bold"
frag = "magenta bold"
old = "red bold"
new = "green bold"
whitespace = "red reverse"

```

[color "diff-highlight"]

```

oldNormal = "red bold"
oldHighlight = "red bold 52"
newNormal = "green bold"
newHighlight = "green bold 22"

```