bioengineering-toolbox

Mar 13, 2021

Contents:

CHAPTER 1

Code development

1.1 Git FAQ

1.1.1 General

How to ignore specific folders

1.1.2 GitHub

How to link to a specific line number on GitHub

First get a permanent link to files you want to link to on on GitHub. You can do this by viewing the file and pressing the "y" key to update the URL to a permalink to the exact version of the file you see (see this link for more information).

Then click on the line number you want (like line 18), and the URL in your browser will get a #L18 tacked onto the end. You can now copy and use this url.

1.1.3 CLI (command line interface)

Backup untracked files

To copy untracked files into a separate directory outside the git repository before cleaning up, use

```
git ls-files --others --exclude-standard -z -m - d | cpio -pmd0 .
./untracked-backup/
```

Options used:

- `--others` lists untracked files
- `--exclude-standard` uses .gitignore and the default git excludes

- `-z` null-delimited output
- `-p` pass-through (=copy)
- `-m` preserve modification times
- `-d` create missing directories
- `-0` null-delimited input

These commands might be useful when migrating repositories e.g. from bitbucket to github. This allows the untracked changes to be move to a separate folder which can then be copied over to the new location where the new repo is located.

Getting git on a server

Determine the URL that a local Git repository was originally cloned from?

- If referential integrity is intact, use: `git remote show origin`
- If referential integrity has been broken, use `git config --get remote .origin.url`

Change an existing local git repository's remote URL from HTTPS to SSH

Cloning/pushing to a repository using a HTTPS remote URL's (e.g. https://github.com/PrasadBabarendaGamage/iron.git) requires a password. This can be avoided by cloning any new repository using a SSH remote URL (e.g. git@github.com:PrasadBabarendaGamage/iron.git), or by changing the remote URL for an existing local repository to use the SSH remote URL.

To change a existing local git repository's remote URL from HTTPS to SSH:

1. Edit the .git/config file under the root directory of the repository.
2. Find `url=entry` under section `[remote "origin"]`
3. Change it from `url=https://github.com/PrasadBabarendaGamage/iron .git` to `url=ssh://git@github.com:PrasadBabarendaGamage/iron.git`. that is, change all the texts before the `@` symbol to `ssh://git`
4. Save config file and quit.

Now you can use `git push origin your_branch` to sync with your remote repo.

Note that before you can push to your remote repo, you will need to have setup a ssh key pair e.g. in on github.

Adding ssh key to GitHub

Make a git submodule to track remote branch

```
# add submodule to track master branch
git submodule add -b master [URL to Git repo];

# Make sure the parent repo knows that its submodule now tracks a branch:
cd /path/to/your/parent/repo
git config -f .gitmodules submodule.<path>.branch <branch>

# Make sure your submodule is actually at the latest of that branch:
cd path/to/your/submodule
```

(continues on next page)

(continued from previous page)

```
git checkout -b branch --track origin/branch
# if the master branch already exist:
git branch -u origin/master master
# (with 'origin' being the name of the upstream remote repo the submodule␣
↪has been cloned from.
#  A git remote -v inside that submodule will display it. Usually, it is
↪'origin')

# Don't forget to record the new state of your submodule in your parent repo:
cd /path/to/your/parent/repo
git add path/to/your/submodule
git commit -m "Make submodule tracking a branch"
# Subsequent update for that submodule will have to use the --remote option:
# update your submodule
# --remote will also fetch and ensure that
# the latest commit from the branch is used
git submodule update --remote

# to avoid fetching use
git submodule update --remote --no-fetch
```

When cloning a repository with a submodule, use the following command to initialise and update the submodules.

```
git submodule update --init --remote
```

How do I make Git ignore file mode (chmod) changes?

```
git config core.fileMode false
```

```
git config core.fileMode false
```

```
core.fileMode
   If false, the executable bit differences between the index and the
   working copy are ignored; useful on broken filesystems like FAT.
   See git-update-index(1). True by default.
```

1.2 Github FAQ

1.2.1 Adding badges to repositories

1.2.2 Renaming repositories and updating remote paths of clones

```
# Get the remote origin url of the old repository.
git remote get-url origin
# This will produce output like e.g.:
# git@github.com:UOA-Heart-Mechanics-Research/heart_metadata.git

# Set the new remote origin url to the new repository name.
git remote set-url origin git@github.com:UOA-Heart-Mechanics-Research/heart-metadata.
↪git
```

1.3 Pycharm FAQ

Click on the links in the headings for more information.

1.3.1 How to open interactive python console by default?

CHAPTER 2

Documentation

2.1 Creating documentation

This section describes how sphinx documentation can be created for your project. Typically this documentation would be generated within a code repository.

2.1.1 Getting started

Follow the quick-start instructions on the official sphinx documentation page During this process, it will ask `Separate source and build directories (y/n)`. Select yes.

2.1.2 Example repositories

https://so-tools.readthedocs.io/

2.1.3 Markdown

By default, sphinx recognises RestructuredText. Markdown is an alternative, and relatively simpler plain text format for writing structured documents. Sphinx supports an unambiguous version of Markdown called CommonMark which addresses many of Markdown's limitations.

Markdown best practices

A guide to best practices when using Markdown can be found in the following link.

Adding support for Markdown files

CommonMark can be enabled in Sphinx by including the `Recommonmark` extension in the Sphinx `conf.py`. See official sphinx documentation page for instructions on how to enable this sphinx extension.

Adding new markdown files

A new markdown file can be added to your project as described below (ensure that the `Recommonmark` Sphinx extension has been enabled as described in the previous section).

1. Create a new text file with the `.md` extension in appropriate folder within the standard sphinx `docs/source` folder, e.g. `docs/source/my/folder/file.md`.
2. This new file can be added to the main Sphinx table of contents by adding the filename to the `index.rst` file in the `docs/source/` folder as shown below:

```
==========================
Welcome to my documentation
==========================

.. toctree::
   :maxdepth: 2
   :caption: Contents:

   my/folder/file
```

2.1.4 Building documentation

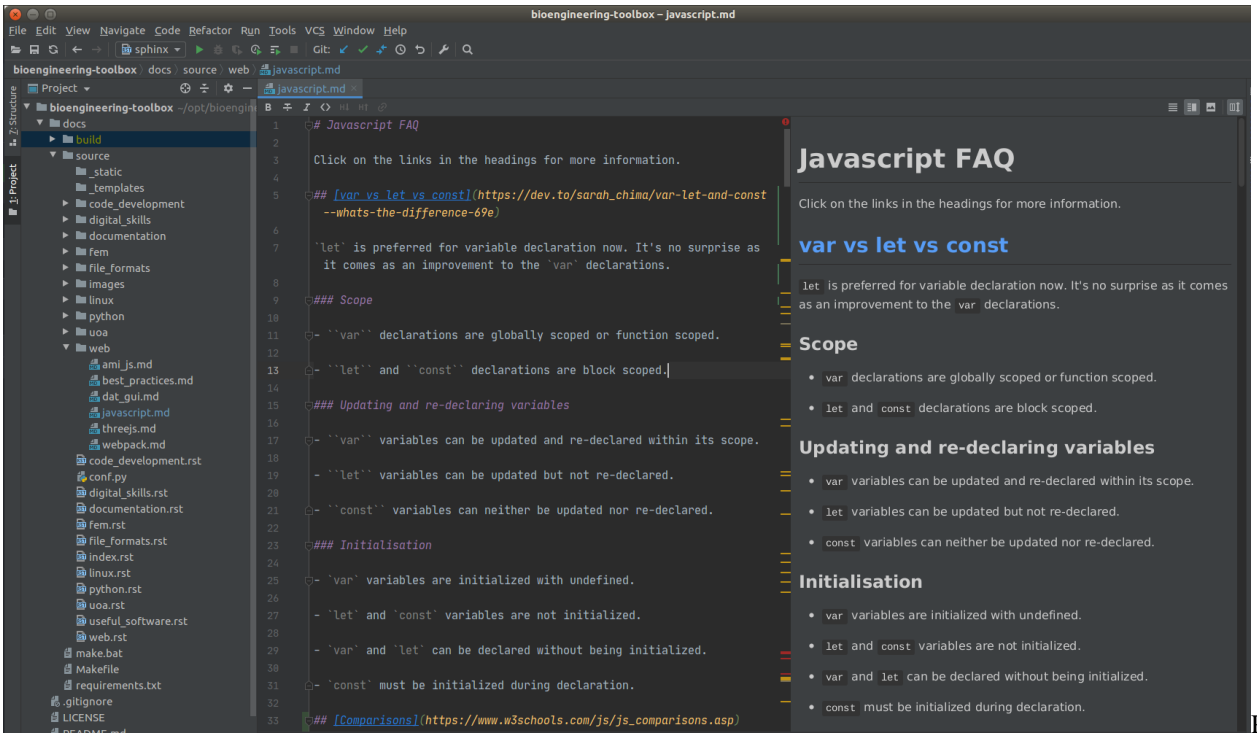
Building documentation from the command line

Change directory to the folder where the sphinx documentation was generated (typically `/path/to/docs/`) in the terminal, and run the following command:

```
make html
```

Building documentation using Pycharm

Follow the instructions in the following link. Python provides a markdown plugin that allows for dynamic previewing of markdown files as shown in the figure below:



Pycharm dynamic markdown preview!

2.1.5 Additional features

Linking to heading sections in other markdown files

When writing documentation, it might be useful to reference headings sections in other markdown files. This can be enabled using the following instructions.

For example, to link to a section named `# My subtitle` in a file located in `/path/to/file.md`, relative to the `docs/source` folder.

```
[text for the link](/path/to/file:My%20Subtitle)
```

NOTE:

1. Any spaces in headings need to be replaced by `%20`.
2. The .md file extension should not be included in the path to the file.

2.1.6 restructuredText

Add ability to reference sections

See this Section

2.2 Sphinx FAQ

2.2.1 Force rebuild of html, including autodoc

```
html -E -a
```

2.2.2 Adding a sphinx build configuration to pycharm

1. Open Pycharm.
2. File -> Open -> Navigate to your project folder -> Click Ok.
3. Run -> Edit configurations.
4. Click the + button and selected `python docs` -> `sphinx` and complete the following fields:
    1. Name: `Sphinx`
    2. Input: Select the source directory in the documentation folder
    3. Output: Select the build directory in the documentation folder
    4. Python interpretor: Select your interpreter
    5. Options: `-E -a` (This options forces rebuild of html)
5. Click ok.

2.2.3 Table width fix for Read the Docs Sphinx theme

2.2.4 Sphinx and Jupyter notebooks

Linking to Jupyter notebooks in Sphinx

https://nbsphinx.readthedocs.io/en/0.7.1/

Embedding Jupyter code into Sphinx Restructured Text

https://jupyter-sphinx.readthedocs.io/en/latest/

```
.. jupyter-execute::

  name = 'world'
  print('hello ' + name + '!')
```

Linking to notebooks outside of sphinx doc folder

2.2.5 Updating logo for sphinx_rtd_theme

```
html_theme = 'sphinx_rtd_theme'
html_static_path = ['_static']
html_logo = 'logo.svg'
```

2.2.6 Updating sphinx_rtd_theme colours

You can change the theme colors by adding a custom CSS file to _static. To actually have Sphinx use that file, add this to your `conf.py`:

```
# Update navigation background colour.
def setup (app):
    app.add_css_file('custom.css')
```

Create a custom.css file to the `docs/source/_static` folder with the following:

```
.wy-side-nav-search, .wy-nav-top {
    background: #0b750a;
}
```

2.2.7 Graphs in Sphinx/RestructuredText

Add `sphinx.ext.graphviz` to extensions in `conf.py` Add `graphviz_output_format = "svg"` in `conf.py` Example (see output here)

```
.. graphviz::

   digraph foo {
      rankdir="BT";
      graph [fontname="avenir", fontsize=10];
      node [fontname="avenir", fontsize=10, target="_blank" shape=rectangle,␣
↪style=filled, fillcolor=darkseagreen2];
      edge [fontname="avenir", fontsize=10, style=dashed, arrowhead=onormal];
      Thing [label="SO:Thing", href="https://schema.org/Thing"];
      CreativeWork [href="https://schema.org/CreativeWork"];
      Dataset [href="https://schema.org/Dataset"];
      MediaObject [href="https://schema.org/MediaObject"];
      DataDownload [href="https://schema.org/DataDownload"];
      Intangible [href="https://schema.org/Intangible"];
      PropertyValue [href="https://schema.org/PropertyValue"];
      Place [href="https://schema.org/Place", target="_blank"];
      Person [href="https://schema.org/Person", target="_blank"];
      Organization [href="https://schema.org/Organization"];

      CreativeWork -> Thing;
      Intangible -> Thing;
      Place -> Thing;
      Person -> Thing;
      Organization -> Thing;
      Dataset -> CreativeWork;
      MediaObject -> CreativeWork;
      DataDownload -> MediaObject;
      PropertyValue -> Intangible;
   }
```

2.3 Markdown FAQ

2.3.1 Best practices

2.3.2 Cheatsheet

2.3.3 Table generator

2.3.4 Escape characters

| Character to escape | Code | |——————————— |———- |||| | |

2.3.5 Create note

```
> **_NOTE:_** The note content.
```

2.4 Restructured text FAQ

restructuredText is a lightweight markup language (file format) that is commonly used to create documentation. The langauge is designed to be both: (a) processable by documentation-processing software such as Docutils, and (b) easily readable by human programmers who are reading and writing source code.

See these links for a guide to the rst file format.

- guide
- cheatsheet

To find out about the differences between reST, docutils, readthedocs, sphinx, and extensions, see this link.

Guide to setting up a sphinx project

2.4.1 Examples of well documented code repositories

- http://libcellml.readthedocs.io/
- http://morphic.readthedocs.io/

2.4.2 Commenting restructuredText files

```
..
   _This: is a comment!

..
   [and] this!

..
   this:: too!

..
   |even| this:: !
```

Avoid putting comments on the same line as the double dots:

```
.. Avoid this type of comment
```

This is considered bad practice since it may lead to unintended consequences if the comment matches a proper markup construct.

2.4.3 Hyperlinks and cross-referencing

Hyperlinking

```
`anchor text <url>`__
```

Referencing section headings

Add `sphinx.ext.autosectionlabel` to extensions in :file:`conf.py`. Add the following to :file:`conf.py`.

```
# True to prefix each section label with the name of the document it is in, followed␣
↪by a colon. For example, index:Introduction for a section called Introduction that␣
↪appears in document index.rst. Useful for avoiding ambiguity when the same section␣
↪heading appears in different documents.
autosectionlabel_prefix_document = True
```

Example usage:

```
A Plain Title
-------------

This is the text of the section.

It refers to the section title, see :ref:`A Plain Title`.

To reference section headings in other files:
:ref:`path/to/file:Section heading`
```

Cross-Referencing within a document

```
:ref:`anchor text <link-target>`
```

Cross-Referencing between documents

```
:ref:`anchor text <document-tag:link-target>`
```

Whenever you use the cross-referencing mechanism, you should create a link target for a section chapter, by adding a label before the section header:

```
.. _link-target:

Headline
========
```

Referencing external files

Reference files and paths Use the following syntax to reference files and paths:

```
:file:`myfile.txt`
```

This will output: :file:`myfile.txt`.

You can reference paths in the same way:

```
:file:`path/to/myfile.txt`
```

This will output: :file:`path/to/myfile.txt`.

Reference an element in a GUI

Use the following syntax to direct a user to click a link or look to a certain area of the GUI:

```
:guilabel:`Main Menu`
```

This will output: :guilabel:`Main Menu`.

External files

Text snippets, large blocks of downloadable code, and even zip files or other binary sources can all be included as part of the documentation. To include files as part of the build process, use the following syntax:

```
:download:`An external file <readme.txt>`
```

2.4.4 Showing example code

Code blocks

```
.. code-block:: language

   code
```

Inline code block with syntax highlighting

First define a custom role. e.g.:

```
.. role:: bash(code)
   :language: bash
```

This then allows use inline code e.g.

```
Here is some awesome bash code :bash:`a = b + c`.
```

Forcing syntax highlighting for a code snippet.

e.g. for a partial snippet of json code:

```
.. code-block:: json-object
   :force:

    "scripts": {
      "build": "webpack",
      "serve": "webpack-dev-server"
    },
```

2.4.5 Add images

Use the image directive, for example:

```
.. image:: example.png
```

The path to the image is relative to the file. See the Sphinx documentation for more information.

2.4.6 Creating tables

- Programtically within .rst files
- Load from csv files

2.4.7 Notes and warnings

2.5 ReadTheDocs FAQ

2.5.1 Adding Github organisation repos to readthedocs

Your ReadTheDocs account may not be able to see repositories within a Github organisation you belong. To address this issue:

1. If you have not already done so, connect your github account to ReadTheDocs by logging into ReadTheDocs and click sign-in via github.
2. Go to your github account settings page.
3. Select :guilabel:`Applications` from the left hand side menu.
4. Click on the :guilabel:`Read The docs Community` row.
5. Under the `Organisation access` section, click the :guilabel:`grant` button next to the organisation that you want ReadTheDocs access.
6. Go to :guilabel:`My Projects` on ReadTheDocs.org.
7. Click :guilabel:`Import a Project`.
8. CLick the refresh button.
9. Navigate through the repositories and you should see repositories under your github organisation (note that your organisation may not show on the organisation filter list, however, they will show up in the list of repositories section of the page.)

2.5.2 Troubleshooting

contents.rst not found

By default, readthedocs looks for a contents.rst file. However, by default sphinx creates an index.rst file. Readthedocs will therefore raise the following error:

```
Sphinx error:
master file /home/docs/checkouts/readthedocs.org/user_builds/bioengineering-toolbox/
↪checkouts/latest/docs/source/contents.rst not found
```

To address this include the following line in the conf.py file:

```
# Set the root rst to load. This is required to be named contents to allow
# readthedocs to host the docs using its default configuration.
master_doc = 'index'
```

Table width fix for Read the Docs Sphinx theme

My ReadTheDocs project isn't building

First, you should check out the Builds tab of your project. That records all of the build attempts that RTD has made to build your project. If you see `ImportError` messages for custom Python modules, you should enable the virtualenv feature in the Admin page of your project, which will install your project into a virtualenv, and allow you to specify a `requirements.txt` file for your project.

You can create this file using the following steps

> ***NOTE:*** The following steps assume you have successfully built the Sphinx documentation on your local machine (i.e. that all the required modules are installed on your machine).

Open the conf.py of your sphinx documentation (usually located inside the `docs` folder), and navigate to the extensions section (an example of which is shown below)

```
# Add any Sphinx extension module names here, as strings. They can be
# extensions coming with Sphinx (named 'sphinx.ext.*') or your custom
# ones.
extensions = [
    'sphinx.ext.autodoc',
    'sphinx.ext.doctest',
    'sphinx.ext.intersphinx',
    'sphinx.ext.todo',
    'sphinx.ext.coverage',
    'sphinx.ext.imgmath',
    'sphinx.ext.viewcode'
]
```

The Extensions (or in other words - the python modules) that inbuilt into Sphinx are denoted by `the sphinx.` prefix. The other python modules will need to be installed when ReadTheDocs builds the documentation on their server.

To see what python modules are currently installed on your machine use the following command in a terminal:

```
pip freeze > requirements.txt
```

Search in this file for the items that match the other python modules required by ReadTheDocs (in the example above, search for `sphinxcontrib`). Move these lines to the top of the file and delete the remaining lines.

Move the `requirements.txt` file in the documentation directory of your repository (commonly labelled `docs`). Add the location of this file to the ReadTheDocs project admin page and rebuild the documentation on ReadTheDocs.

CHAPTER 3

Documentation

3.1 Docker FAQ

3.1.1 Common commands

```
# List available images
sudo docker images

# List running containers
sudo docker container ls

# Build docker image from a dockerfile in the current directory
sudo docker build --tag image_name:image_tag .

# Remove all images
docker image prune -a
or
docker rmi -f $(docker images -a -q)

# Acess a bash shell within a running container
sudo docker run --rm -it 6c8bbe6e0ffe /bin/bash

# Echo output when running docker build
RUN echo $(mpirun --version)
RUN echo $(ls ~)
RUN echo $($PATH)
```

3.1.2 Activating a Conda environment in your Dockerfile

```
# Make RUN commands use the new environment:
SHELL ["conda", "run", "-n", "myenv", "/bin/bash", "-c"]
```

3.1.3 Tips to Reduce Docker Image Sizes

3.1.4 Create Docker Images for Docker Hub

3.1.5 Debugging applications with totalview in dockers

CHAPTER 4

FEM

4.1 OpenCMISS FAQ

4.1.1 Building OpenCMISS

```
~/hpc/opt/OpenCMISS/cercmissprd01
mkdir main
cd main/
git clone https://github.com/OpenCMISS/setup.git
mkdir setup-build
cd setup-build
cmake -DOPENCMISS_ROOT=../ -DOPENCMISS_PERFORM_INITIAL_BUILD=false -
↪DOPENCMISS_CONFIG_BUILD_TYPE=Debug ../setup
make
cd ../build/manage/release/
pluma OpenCMISSInstallationConfig.cmake
# set(IRON_BRANCH devel)
# set(IRON_DEVEL git@github.com:PrasadBabarendaGamage/iron)
# option(OPENCMISS_USE_ARCHITECTURE_PATH "Use architecture path to enable␣
↪multiple configs in the same installation." YES)
# option(OPENCMISS_DEVEL_ALL "Download/checkout development branches of all␣
↪components of the OpenCMISS build." YES)
cmake -DOPENCMISS_MPI=mpich -DOPENCMISS_MPI_USE_SYSTEM=NO -DOPENCMISS_
↪TOOLCHAIN=gnu -DOPENCMISS_BUILD_TYPE=debug .
make create_config
pluma configs/x86_64_linux/gnu-C5.4-gnu-F5.4/mpich_release/
↪OpenCMISSLocalConfig.cmake
# set(OC_SYSTEM_LIBXML2 OFF)
# set(OC_PYTHON_BINDINGS_USE_VIRTUALENV YES)
# set(OC_USE_ZINC OFF)
# set(IRON_WITH_Python_BINDINGS YES)
make build_config
cd /home/psam012/hpc/opt/OpenCMISS/cercmissprd01/main/install/x86_64_linux/
↪gnu-C5.4-gnu-F5.4/mpich_release/python/Release
```

(continues on next page)

(continued from previous page)

```
python setup.py install --force
source /home/psam012/hpc/opt/OpenCMISS/cercmissprd01/main/install/x86_64_
↪linux/gnu-C5.4-gnu-F5.4/mpich_release/virtual_environments/oclibs_venv_
↪py27_release/bin/activate
```

4.1.2 Using python bindings on hpc3

```
# To use it, open a terminal and log into the hpc3 machine:
ssh hpc3 # or
ssh upi@hpc3

# Make sure you are running the bash shell
bash

# Setup environmental variables for OpenCMISS
source /people/cmiss/develop_opencmiss.sh

# Tell python where the OpenCMISS libraries are located
source /people/cmiss/opencmiss/install/x86_64_linux/gnu-C4.4-gnu-F4.4/
↪openmpi_release/virtual_environments/bin/activate

# Go to the directory where the Laplace example is located (if needed,␣
↪download from the OpenCMISS-examples repo)
cd python/example/location/

# Run the OpenCMISS example.
python LaplaceExample.py
```

4.1.3 Debugging OpenCMISS examples

Set diagnostics on and off before/after the line of interest.

Use Totalview.

4.1.4 Hydrostatic pressure export

Element constant hydrostatic pressure fields are stored in the exelem files.

4.2 CMGUI FAQ

4.2.1 Replicating cmgui 3.01 visualisation using gfx commands

```
# For each region you need to list commands to recreate computed fields
# (this orders them so that source fields are defined first, otherwise
# its alphabetical):
gfx list field "/" commands; # for root region
gfx list field REGION_PATH commands; # for any other region
# Images (textures) are now fields in each region so for each region with
# images list:
```

(continues on next page)

(continued from previous page)

```
gfx list texture commands region REGION_PATH;
# tessellation, material, spectrum are needed if you created any of your
# own (including tessellations automatically created for old commands:
gfx list tessellation;
gfx list material commands;
gfx list spectrum commands;
# there are a bunch of other objects such as lights, graphics_filter which
# you probably don't change
# graphics are simply listed for all regions
gfx list g_element comm;
# list window commands (for any windows you have):
gfx list window 1 commands;
```

CHAPTER 5

File formats

5.1 JSON FAQ

5.1.1 JSON examples

5.1.2 Schema

Understanding schema

Tutorial for creating schema

Python module for schema validation

https://python-jsonschema.readthedocs.io/en/stable/

CHAPTER 6

Linux

6.1 Linux FAQ

Click on the links in the headings for more information.

6.1.1 Find files in a terminal

```
find / -type f -iname "*postgis-2.0.0*"
```

where:

- `./` indicates that you want to search in the current directory. This can be replaced with the directory you want to start your search from.
- `f` can be replaced with d if you're searching for a directory instead of a file
- `-iname` can be replaced with -name if you want the search to be case sensitive
- `*` in the search term can be omitted if you don't want the wildcards in the search. In this case, `*` indicates that any number of different characters could be present where the `*` is located in the string. See this link for more information on wildcards

The `grep` command can be used to simplify things further:

```
find . | grep -i "screen"
```

where `-i` indicates the search term is case insensitive. see this link for further details about the `grep` command.

6.1.2 Change file permissions

Change permissions :

```
chmod -R 755 your_directory
```

Value	Description
755	(rwxr-xr-x) The file's owner may read, write, and execute the file. All others may read and execute the file. This setting is common for programs that are used by all users.

6.1.3 tar/untar or zip/unzip

Compress:

```
tar -zcvf archive_name.tar.gz directory_to_compress
tar -jcvf archive_name.tar.bz2 directory_to_compress
zip -r archive_name.zip directory_to_compress
```

Uncompress:

```
tar -zxvf archive_name.tar.gz
tar -jxvf archive_name.tar.bz2 -C /tmp/extract_here/
unzip archive_name.zip
```

6.1.4 Search for files in directory

```
find / -name 'program.c'
```

6.1.5 Find size of directory

```
du -sh folder
du -h -d1 folder # Only the top level directories.
```

6.1.6 Print all environmental variables

```
printenv
```

To show a list including the "shell variables":

```
( set -o posix ; set ) | less
```

6.1.7 Download file from the internet

```
wget http://www.openss7.org/repos/tarballs/strx25-0.9.2.1.tar.bz2
```

6.1.8 Restart network manager (Ubuntu 16.04)

```
sudo systemctl restart NetworkManager
# or
sudo service network-manager restart
```

This is useful if you VPN disconnects and you get the following error when trying to connect.

```
psam012@pc:~$ ssh psam012@bioeng10.bioeng.auckland.ac.nz
ssh: Could not resolve hostname bioeng10.bioeng.auckland.ac.nz: Temporary␣
↪failure in name resolution
```

6.1.9 GUI utility to mount remote filesystems over SSH

Older versions of Ubuntu (<11.10)

By default the GVFS in GNOME supports SSH connections. Click on the nautilus icon (the folder with the house on it) it will launch the file manager.

`linux/images/ssh-remote_mount-file-manager.png`

Then go to File -> "Connect to server":

`linux/images/ssh-remote_mount-dialog.png`

Paste in the following:

Sever:	bioeng10.bioeng.auckland.ac.nz
Folder:	/hpc/upi
UserName:	upi

Click connect and put in your password (ask it to forget immediately for now)

This will mount the remote SSH server in a folder you can access from the file manager. If you need to access the mount in the command line it's in .gvfs.

You can also mount it in Nautilus. Just hit CTRL+L and in the address bar type: `ssh://server-ip/somepath/`

Newer versions of Ubuntu (>11.10)

press Alt+F2 and type nautilus-connect-server as shown below.

linux/images/ssh-remote_mount-unity-step1.png

linux/images/ssh-remote_mount-unity-step2.png

Adding menu items in gnome ^^^

6.2 Screen FAQ

Install screen on remote machine. After you ssh in, type:

```
screen -S insert_name_of_session_here
```

You'll get a welcome message, then you'll be back at a prompt. From there type your command to start whatever app you want. Once it's running, press Ctrl-a (control and a at the same time), then press d. These keystrokes does what's called detaching the screen session. After detaching, you'll be back at a command prompt. From there you can logout. The app you started in the screen session keeps on running in the background.

To re-attach to your screen session: ssh in, then at a terminal prompt type:

```
screen -list
```

This will show all the remote terminal sessions. Copy the session name of the screen you are interested it. Re-attach the screen by typing in

```
screen -r session_name
```

6.3 Photo and video FAQ

6.3.1 Video software

6.3.2 Create gifs from images (commandline)

```
sudo apt-get install imagemagick
convert -delay 20 -loop 0 *.jpg myimage.gif
```

CHAPTER 7

Python

7.1 Python FAQ

General tips can be found here: http://book.pythontips.com/en/latest/index.html

Click on the links in the headings for more information.

7.1.1 Python virtual environments

- Data camp tutorial
- Real Python tutorial

7.1.2 Strings

```
# Replace substring in string (where the item in [] is an optional argument.
str.replace(old_substring, new_substring [, count])
```

7.1.3 Displaying docstrings to see function arguments and info

```
help(function)
```

See Doc string conventions (PEP257) for information about how to write docstrings.

7.1.4 Comparisons

Check if variable is a certain type

```
# Check if a local variable exists.
if 'myVar' in locals():
    pass  # myVar exists.
# Check if a global variable exists.
if 'myVar' in globals():
    pass  # myVar exists.
# Is variable a list?
isinstance(variable, list)
# Check if variable is a function.
callable(obj)
```

7.1.5 Objects

```
# Check if an object has an attribute.
if hasattr(obj, 'attr_name'):

# Set an attribute.
setattr(obj, 'attr_name', 'attr_value')

# Set an attribute within a class.
setattr(self, 'attr_name', 'attr_value')
```

7.1.6 Dictionaries

```
# Creating dictionaries.
dict = {}
# The key is 'color', the item is 'blue'.
dict['color'] = 'blue'
# or
dict = {'color': 'blue'}

# Iterating through keys.
for key in dict:
  print(key)

# Listing items (as a tuple).
items = a_dict.items()
#prints: dict_items([('color', 'blue')])

# Iterating through items.
for item in dict.items():
  print(item)

# Iterating through key-item pairs.
for key, value in dict.items():
  print(key, '->', value)
```

7.1.7 Path and IO functions

See os.path and shutil for more information.

```
import os
os.path.exists(path) # Returns true for directories.
os.path.isfile(path) # Returns false for directories.
os.path.isdir(path)
os.path.join(path, *paths)
os.path.split(path) # Returns (head, tail) where tail is the last pathname component
os.path.dirname(path) # Returns the head of the path.
os.path.basename(path) # function returns the tail of the path.
# Get file extension.
filename, file_extension = os.path.splitext('/path/to/somefile.ext')

# Check if a directory exists and create it if necessary
if not os.path.exists(directory):
    os.makedirs(directory)

import shutil
shutil.copyfile(src, dst) # dst must be the complete target file name.
shutil.copy(src, dst) # dst can be a directory.
shutil.rmtree() # will delete a directory and all its contents.

# Find module directory.
import my_module
module_path = os.path.dirname(my_module.__file__)
```

7.1.8 Load/save json

```
# Load.
import json
with open('strings.json') as json_data:
    d = json.load(json_data)
    print(d)

# Save.
import json
with open('data.txt', 'w') as outfile:
    json.dump(data, outfile, indent=4)
```

7.1.9 Understanding decorators

7.1.10 Manually throw/raise an exception

```
raise ValueError('A very specific bad thing happened')
```

The class hierarchy for built-in exceptions is:

```
    BaseException
     +-- SystemExit
     +-- KeyboardInterrupt
     +-- GeneratorExit
     +-- Exception
          +-- StopIteration
          +-- StopAsyncIteration
          +-- ArithmeticError
```

(continues on next page)

(continued from previous page)

```
          |    +-- FloatingPointError
          |    +-- OverflowError
          |    +-- ZeroDivisionError
          +-- AssertionError
          +-- AttributeError
          +-- BufferError
          +-- EOFError
          +-- ImportError
               +-- ModuleNotFoundError
          +-- LookupError
          |    +-- IndexError
          |    +-- KeyError
          +-- MemoryError
          +-- NameError
          |    +-- UnboundLocalError
          +-- OSError
          |    +-- BlockingIOError
          |    +-- ChildProcessError
          |    +-- ConnectionError
          |    |    +-- BrokenPipeError
          |    |    +-- ConnectionAbortedError
          |    |    +-- ConnectionRefusedError
          |    |    +-- ConnectionResetError
          |    +-- FileExistsError
          |    +-- FileNotFoundError
          |    +-- InterruptedError
          |    +-- IsADirectoryError
          |    +-- NotADirectoryError
          |    +-- PermissionError
          |    +-- ProcessLookupError
          |    +-- TimeoutError
          +-- ReferenceError
          +-- RuntimeError
          |    +-- NotImplementedError
          |    +-- RecursionError
          +-- SyntaxError
          |    +-- IndentationError
          |         +-- TabError
          +-- SystemError
          +-- TypeError
          +-- ValueError
          |    +-- UnicodeError
          |         +-- UnicodeDecodeError
          |         +-- UnicodeEncodeError
          |         +-- UnicodeTranslateError
          +-- Warning
               +-- DeprecationWarning
               +-- PendingDeprecationWarning
               +-- RuntimeWarning
               +-- SyntaxWarning
               +-- UserWarning
               +-- FutureWarning
               +-- ImportWarning
               +-- UnicodeWarning
               +-- BytesWarning
               +-- ResourceWarning
```

7.2 Python bindings

Click on the links in the headings for more information.

7.2.1 Manually building h5py bindings with local version of hdf5

```
python setup.py build_ext --include-dirs=/hpc/psam012/usr/hdf5/hdf5-1.8.12/include/ --
↪library-dirs=/hpc/psam012/usr/hdf5/hdf5-1.8.12/lib/
python setup.py install --home=/hpc/psam012/opt/summer-projects/2017/python-modules/
```

Alternative is to set the HDF5_DIR environmental variable

```
    export HDF5_DIR=/hpc/psam012/usr/hdf5/hdf5-1.8.12/
```

Then update your ~/.bashrc

```
    #export PYTHONPATH="/hpc/psam012/opt/summer-projects/2017/python-modules/lib/
↪python:$PYTHONPATH"
```

7.3 Virtualenv FAQ

7.3.1 Create virtual environments for python with conda

7.4 Pandas FAQ

Click on the links in the headings for more information.

7.4.1 Create empty dataframe

```
df_empty = pd.DataFrame()
```

7.4.2 Creating & editing entries

```
# Replacing pandas dataframe column values with another value.
# Values to replace = ['ABC', 'AB']
# Replacement value = 'A'
df['BrandName'] = df['BrandName'].replace(['ABC', 'AB'], 'A')

# Turn off warnings where overwriting dataframe values.
pd.options.mode.chained_assignment = None  # default='warn'
```

7.4.3 Concatenating dataframes

```
df_empty = pd.DataFrame()
```

7.4.4 Renaming headers

```
# Provide a dictionary with "before":"after" names of the items to be␣
↪renamed.
df.rename(columns={"A": "a", "B": "c"})
```

7.4.5 Concatenating

```
# Concatenating dataframes by row, ie appending rows with the same header.
result = df1.append(df4, sort=False)
# or
result = pd.concat([df1, df4], axis=0, sort=False)

# Concatenating dataframes by column, ie appending additional header columns.
result = pd.concat([df1, df4], axis=1, sort=False)
```

This may require reindexing each dataframe that needs to be appended - see this Concatenating for more information.

7.4.6 Find number of rows in dataframe

```
len(df)
```

7.4.7 Indexing data

```
df.loc[row_indexer,column_indexer]
```

7.5 Jupyter notebook FAQ

7.5.1 Guide

7.6 Matplotlib FAQ

7.6.1 Plotting tutorials

7.7 Morphic FAQ

Click on the links in the headings for more information.

7.7.1 Using groups

```
cranial_elem = range(11)
for element in mesh.elements[cranial_elem.tolist()]:
    element.add_to_group('cranial')
for element in mesh.elements.groups['cranial']:
    print node.id
```

7.7.2 Build wxPython on Ubuntu 16.04 (required by mayavi2, which is used by morphic)

sudo apt-get install libgstreamer-plugins-base0.10-dev

7.8 Mayavi FAQ

Click on the links in the headings for more information.

7.8.1 Speed up mayavi rendering

You've just created a nice Mayavi/mlab script and now want to generate an animation or a series of images. You realize that it is way too slow rendering the images and takes ages to finish. There are two simple ways to speed up the rendering. Let's assume that obj is any Mayavi pipeline object that has a scene attribute:

```
obj.scene.disable_render = True
# Do all your scripting that takes ages.
# ...
# Once done, do the following:
obj.scene.disable_render = False
```

This will speed things up for complex visualizations sometimes by an order of magnitude.

While saving the visualization to an image you can speed up the image generation at the cost of loosing out on anti-aliasing by doing the following:

```
obj.scene.anti_aliasing_frames = 0
```

The default value is typically 8 and the rendered image will be nicely anti-aliased. Setting it to zero will not produce too much difference in the rendered image but any smooth lines will now appear slightly jagged. However, the rendering will be much faster. So if this is acceptable (try it) this is a mechanism to speed up the generation of images.

7.9 Goolge API FAQs

7.9.1 First steps

1 Install the google-api-python-client

```
pip install --upgrade google-api-python-client
```

or locally using:

```
pip install --upgrade google-api-python-client --user
```

2 Turn on Google Sheets API by following the instructions on step 1 of the Quick start guide for python

- Add a new project e.g. test-api
- Add credentials to your project (Google Sheets API, calling API from Other UI, processing user data)
- Create an OAuth 2.0 client ID (Add you name, email address, and product name)

- Download credentials json file (it will be called 'client_id.json' and move this file to your working directory and rename it client_secret.json

3 Run examples here: https://github.com/PrasadBabarendaGamage/google_apis

CHAPTER 8

University of Auckland

8.1 UoA eResearch Virtual machine FAQ

Click on the links in the headings for more information.

8.1.1 Add user on eReserach VM

```
useradd -c "" -m -s /bin/bash xzho001
```

CHAPTER 9

Web

9.1 Best practices

Folder structure

9.2 Javascript FAQ

Click on the links in the headings for more information.

9.2.1 var vs let vs const

`let` is preferred for variable declaration now. It's no surprise as it comes as an improvement to the `var` declarations.

Scope

- `var` declarations are globally scoped or function scoped.
- `let` and `const` declarations are block scoped.

Updating and re-declaring variables

- `var` variables can be updated and re-declared within its scope.
- `let` variables can be updated but not re-declared.
- `const` variables can neither be updated nor re-declared.

Initialisation

- `var` variables are initialized with undefined.
- `let` and `const` variables are not initialized.
- `var` and `let` can be declared without being initialized.
- `const` must be initialized during declaration.

9.2.2 Comparisons

Comparison operators

| Operator | Description | Comparing | Returns | |———- |——————————— |———— |———— | | == | equal to | x == 8 | false | | === | equal value and equal type | x === 5 | true | | != | not equal | x != 8 | true | | !== | not equal value or not equal type | x !== 5 | false | | > | greater than | x > 8 | false | | < | less than | x < 8 | true | | >= | greater than or equal to | x >= 8 | false | | <= | less than or equal to | x <= 8 | true |

Logical operators

| Operator | Description | Example | |————— |————- |————————————- | | && | and | (x < 10 && y > 1) is true | | || | or | (x == 5 || y == 5) is false | | ! | not | !(x == y) is true |

9.2.3 Strings

String concatenation

In javascript the + operator is used to add numbers or to concatenate strings. if one of the operands is a string + concatenates, and if it is only numbers it adds them, for example:

```
1+2+3 == 6
"1"+2+3 == "123"
```

Finding strings

```
// Check if substring in string.
'a nice string'.includes('nice')
// Check if substring in string starting at third letter of the string.
'a nice string'.includes('nice', 3)
```

convert string to other formats

See converting between data types section.

9.2.4 Arrays

```
// Adding elements to arrays using push.
var array = [];
array.push(5);

// Create array [0, 1, 2, 3, 4]
[...Array(5).keys()];

//Flatten array of arrays.
const arrays = [[1], ["2"], [3]];
const merged = [].concat(arrays);

// Get subset of values from an array without modifying original array.
const array  = [1, 2, 3, 4, 5];
// slice from 1..3 - add 1 as the end index is not included
const arraySubset = array.slice(1, 3 + 1);
console.log(arraySubset);

// Check if value is an array.
Array.isArray(your_array)
```

9.2.5 Converting between data types

```
// The parseInt() method converts a string into an integer (a whole number).
// You can even pass in strings with random text in them, but the number needs
// to be the first part of the string.
var text = '42px';
var integer = parseInt(text, 10); // returns 42

// The parseFloat() method converts a string into a point number (a number
// with decimal points). You can even pass in strings with random text in them,
// but the number needs to be the first part of the string.
var text = '3.14someRandomStuff';
var pointNum = parseFloat(text); // returns 3.14

// The Number() method converts a string to a number.
// Sometimes it's an integer. Other times it's a point number. And if you
// pass in a string with random text in it, you'll get NaN, an acronym for
// "Not a Number."
// Less safe than parseInt() and parseFloat()
Number('123'); // returns 123
Number('12.3'); // returns 12.3
Number('3.14someRandomStuff'); // returns NaN
Number('42px'); // returns NaN
```

9.2.6 Conditional operator (if and ?)

```
// If statement.
if (condition1) {
  console.log("condition1 is true");
} else if (condition2) {
  console.log("condition2 is true");
} else {
  console.log("both conditions not true");
```

(continues on next page)

(continued from previous page)

```
}

// Check if object attribute exists.
if (object.attribute) {
  console.log("exists");
}

// ? conditional operator.
let result = condition ? value1 : value2;
// The condition is evaluated: if it's truthy then value1is returned, otherwise
// - value2, e.g:
let accessAllowed = (age > 18) ? true : false;
```

9.2.7 Classes

```
// Constructor is similar to __init__ in python.
class Car {
  constructor(brand) {
    this.carname = brand;
  }
  present(x) {
    return x + ", I have a " + this.carname;
  }
}
mycar = new Car("Ford");
mycar.present('Hello');
```

9.2.8 Objects

```
// Declaring new objects.
// 1. Object literal.
// 1.a  Object with no properties or methods.
var emptyObject = {};
// 1.b  Object with properties and methods.
let person = {
  firstName: "John",
  lastName : "Doe",
  id       : 5566,
  getFullName : function() {
    return this.firstName + " " + this.lastName;
  }
}
// 2. Object constructor.
let person = new Object();
// Attach properties and methods to person object.
person.firstName = "James";
person["lastName"] = "Bond";
person.age = 25;
person.getFullName = function () {
        return this.firstName + ' ' + this.lastName;
};
```

(continues on next page)

(continued from previous page)

```
// Accessing - objectName.propertyName
person.firstName
person["lastName"]

// Accessing object methods - objectName.methodName()
let name = person.getFullName();

// Find number of keys.
Object.keys(person).length

// List value of an object.
Object.values(person)

// Check if object attribute exists.
if (object.attribute) {
  console.log("exists");
}

/// Return first value of an object.
Object.values(person)[0]

// Looping - see looping section.
```

9.2.9 Looping

```
// For each value in array.
const foobar = [1, 2, 3];
for (const value of foobar) {
  console.log(value);
}
// For each index and value in array.
const foobar = ['A', 'B', 'C'];
for (const [index, element] of foobar.entries()) {
  console.log(index, element);
}

// Looping through object values.
const object = {'a': 1, 'b': 2, 'c' : 3};
for (const value of Object.values(object)) {
  console.log(value);
}

// Looping through object keys.
const object = {'a': 1, 'b': 2, 'c' : 3};
for (const key of Object.keys(object)) {
  console.log(key);
}

// Looping through objects keys and values.
const object = {'a': 1, 'b': 2, 'c' : 3};
for (const [key, value] of Object.entries(object)) {
  console.log(key, value);
}
```

9.2.10 TypedArray

```
// Create a TypedArray with a size in bytes.
const typedArray1 = new Int8Array(8);
const typedArray2 = new Float32Array(11);

// Set TypedArray values by direct assignment.
typedArray1[0] = 32;
typedArray2[0] = 32.34;

// Set TypedArray values using set - typedarray.set(array[, offset])
typedArray1.set([1, 2, 3], 3);
```

9.2.11 ES6 Modules

- Everything inside an ES6 module is private by default, and runs in strict mode (there's no need for 'use strict').
- Public variables, functions and classes are exposed using export.
- Exposed modules are called into other modules using import
- Modules must be included in your HTML with type="module", which can be an inline or external script tag.

```
// Define src in a different file;
<script type="module" src="main.js"></script>

// or an inline script.
<script type="module">

</script>
```

- Modules are deferred, and only run after a document is loaded

9.2.12 Mastering javascript callbacks/bind/apply/call

9.2.13 Fetching

```
addLandmarks() {
let server = '127.0.0.1:5000'
var module = this; // Allows variables to be accessed within then().
fetch(`http://${server}/return_landmarks?participant_id=${this.participant}`)
  .then(function (response) {
    return response.json();
  })
  .then(function (json) {
    console.log(json)
  });
}
```

9.2.14 Adding and removing event listeners with access to 'this'

```
class Test {
  constructor(){
    this.success = false;
    this.onMouseUp = () => {
      // Access attributes of this class.
      this.success = true;
    }
    document.addEventListener('mouseup', this.onMouseUp);
    document.removeEventListener('mouseup', this.onMouseUp);
  }
}
```

9.2.15 Alerts

```
alert('Hi');
```

9.2.16 Disable right-click context menu

```
window.addEventListener('contextmenu', function (e) {
  // do something here...
  e.preventDefault();
}, false);
```

9.3 Dat.Gui FAQ

9.3.1 Accessing object methods and attributes inside an event callback.

Use the following es6 style functions. It will allow accessing all attributes and methods from class that creates the event callback.

```
class Modules {
  constructor() {
    this.showModel = true;
  }
  addGui(gui) {
    this.showModelControl = gui.add(this, 'showModel').listen();

    this.showModelControl.onChange((showModel) => {
      if (showModel) {
        this.showModel = false;
      } else {
        this.showModel = true;
      }
    });
  }
}
```

9.4 Three.js FAQ

9.4.1 Tips and tricks

- General tips
- Optimising performance

9.4.2 Vector to array conversion

```
vertex.toArray()
```

9.4.3 View axes

```
var axesHelper = new THREE.AxesHelper( 5 );
scene.add( axesHelper );
```

9.4.4 Find absolute position of a vertex

9.4.5 View point cloud

View vertices of a point cloud.

```
// Get vertex point cloud.
var getPointsObject = function (points) {
  let geometry = new THREE.Geometry();
  for (const point of points) {
    geometry.vertices.push(point);
  }
  let material = new THREE.PointsMaterial({
    color: "white",
    size: 3,
    sizeAttenuation: false
  });
  return new THREE.Points(geometry, material);
};

// Generate points.
var cube = new THREE.BoxGeometry(1, 1, 1);
const points = cube.vertices

// Get point object and add to scene. ;
scene.add(getPointsObject(points));
```

9.4.6 View vertices of an existing geometry

```
var points = new THREE.Points(geometry, new THREE.PointsMaterial({
  size: 0.25,
  color: "yellow"
```

(continues on next page)

(continued from previous page)

```
}));
scene.add(points);
```

9.4.7 Create BufferGeometry

9.4.8 Dynamic visualisations using BufferGeometry

```
// Setup a buffer attribute for position variable.
const positionAttribute = new THREE.BufferAttribute(
  positions, positionNumComponents);
// Mark positionAttribute as dynamic if its contents change often.
positionAttribute.setUsage(THREE.DynamicDrawUsage);
```

9.4.9 Copying attributes for a buffer geometry

```
var nodeGeometry = new THREE.SphereBufferGeometry( 0.1, 32, 32 );
const bufferedNodeGeometry = new THREE.InstancedBufferGeometry();

// Method 1 (may not copy over all required attributes)
bufferedNodeGeometry.index = nodeGeometry.index;
bufferedNodeGeometry.attributes = nodeGeometry.attributes;

// Method 2
bufferedNodeGeometry.copy( nodeGeometry );
```

9.4.10 Duplicating objects using THREE.InstancedMesh

```
// Add mesh nodes as spheres.
var nodeGeometry = new THREE.SphereBufferGeometry( 0.1, 32, 32 );
var transform = new THREE.Object3D();
// Define a new material instance for the spheres (required)
var nodeMaterial = new THREE.MeshNormalMaterial();
var nodeMesh = new THREE.InstancedMesh( nodeGeometry, nodeMaterial, numVertices );
posNdx = 0;
for (const vertex of unbufferedGeometry.vertices) {
    transform.position.set( vertex.x, vertex.y, vertex.z );
    transform.updateMatrix();
    nodeMesh.setMatrixAt( posNdx ++, transform.matrix );
}
scene.add( nodeMesh );
```

9.4.11 Add stats.js

```
// Import stats.
import * as Stats from 'stats.js';
```

9.4.12 Get camera world coordinates

```
let vector = camera.getWorldDirection();
```

9.4.13 Enable or disable orbit controls

```
orbitControls.enabled = false;
orbitControls.enabled = true;
```

9.5 Webpack 4 FAQ

9.5.1 Getting started

```
# 1. Make project folder.
mkdir webpacktut
cd webpacktut
# 2. Create an empty package.json file.
npm init -y
# 3. Install webpack and it's commandline interface.
npm i -D webpack webpack-cli
```

CHAPTER 10

Visualisations

10.1 Graphviz FAQ

10.1.1 Examples

CHAPTER 11

Indices and tables

- genindex
- modindex
- search