

Zawiki

Release v0.1-8-ga43f0f4

tschinz

Content

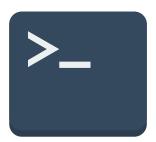
| 1 | Linu | X | 1 |
|---|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| | 1.1 | | 1 |
| | 1.2 | Scripts | 2 |
| | 1.3 | Tools | 2 |
| | 1.4 | Ubuntu | 4 |
| | 1.5 | Filesystem | 11 |
| 2 | Mac | | 12 |
| 3 | Wind | lows | 13 |
| | 3.1 | Firewall | 13 |
| | 3.2 | Group Policies | 13 |
| | 3.3 | Registry | 14 |
| 4 | Git | | 16 |
| | 4.1 | Git Commands | 16 |
| | 4.2 | Git Flow | 17 |
| | 4.3 | Git General | 18 |
| | 4.4 | Git Submodules | 21 |
| 5 | Jupy | ter 2 | 23 |
| | 5.1 | Common Functions | 23 |
| | 5.2 | Extensions | 25 |
| | 5.3 | | 25 |
| | 5.4 | y and the state of | 27 |
| | 5.5 | Installation | 28 |
| 6 | Pand | loc | 31 |
| | 6.1 | Pandoc | 31 |
| 7 | Auto | HotKey AHK | 33 |
| | 7.1 | | 33 |
| | 7.2 | Key Definitions | 35 |
| | 7.3 | Tips & Tricks | 36 |
| 8 | Pytho | on | 38 |
| | 8.1 | | 38 |
| | 8.2 | General | 45 |
| | 8.3 | Flake 8 | 46 |
| | 8.4 | PIP | 47 |

| 9 | | 48 |
|----|--------------------------------------|----|
| | 8 | 48 |
| | | 48 |
| | 9.3 WTFPL | 49 |
| 10 | ROS - Robot Operating System | 50 |
| | | 50 |
| | 10.2 Basics | 51 |
| | 10.3 Books summary | 55 |
| | | 57 |
| | | 60 |
| | | 65 |
| | | 69 |
| | | 73 |
| | $\boldsymbol{\varepsilon}$ | 73 |
| | | 80 |
| | | 85 |
| | 10.12 Transform Frames | 87 |
| 11 | LaTeX | 91 |
| 11 | | 91 |
| | | 92 |
| | 11.2 Insulation Euror | 72 |
| 12 | | 96 |
| | 12.1 Introduction | 96 |
| | 12.2 RST and Sphinx Cheatsheet | 96 |
| 10 | W 11 | 44 |
| 13 | | 11 |
| | 13.1 Github Markdown | |
| | 13.2 H1 | 12 |
| 14 | Multimedia 1 | 18 |
| | | 40 |
| 15 | Security 1 | 19 |
| 16 | Past Time 1 | 20 |
| | 16.1 Programmer Jokes | 20 |
| 15 | | |
| 17 | | 23 |
| | 17.1 About | |
| | 17.2 Credits | |
| | 17.3 How to use Sphinx Documentation | |
| | 17.5 License | |
| | 17.5 Electise | 20 |
| 18 | Welcome to Zawiki Knowledge Database | 29 |
| | 18.1 Site purpose and structure | 29 |

Linux



1.1 Commandline



1.1.1 General Shell Commands

- Change permissions on type
- SSH relia

Change permissions on type

```
sudo find /var/www -type d -print0 | sudo xargs -0 chmod 0755
sudo find /var/www -type f -print0 | sudo xargs -0 chmod 0644
```

SSH relia

```
ssh -p 2222 -L 5900:localhost:5900 -L 19999:localhost:19999 zas@relia.zapto.org
```

1.2 Scripts



1.2.1 General

• Shell Bang

Shell Bang

#!/bin/sh

1.3 Tools



1.2. Scripts 2

1.3.1 Let's Encrypt

- Version
- Renew Certificates

Version

```
certbot --version
```

Renew Certificates

```
# Stop Webserver Service
sudo service apache2 stop

# Update Certificates
sudo certbot renew
sudo certbot renew --dry-run

# Restart Webserver Service
sudo service apache2 start
```

1.3.2 Systemd

- List services
- Status service
- Start Stop Service
- Add Service
- Add in vim file
- Start Service manually
- Start Service on boot

List services

```
systemctl --type=service
```

1.3. Tools

Status service

```
systemctl status firewalld.service
```

Start Stop Service

```
systemctl stop firewalld.service
systemctl status firewalld.service
```

Add Service

```
od /etc/systemd/system
sudo vim jupyterlab.service
```

Add in vim file

```
[Unit]
Description = Jupyterlab service
After = network.target
StartLimitIntervalSec=0

[Service]
Type=simple
User=zas
ExecStart=/home/zas/Documents/jupyterlab_start.bash

[Install]
WantedBy = multi-user.target
```

Start Service manually

```
systemctl start jupyterlab
```

Start Service on boot

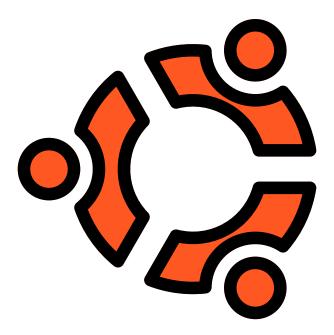
```
systemctl enable jupyterlab
```

1.4 Ubuntu

1.4.1 Installation and Config

```
• Installation
```

- Default Tools
- ZSH
- Oh My ZSH



- SublimeText 3
- SublimeMerge
- Krusader
- Yakuake
- FSearch
- Anaconda
- QT-Creator
- Visual Studio Code
- Configuration
 - Oh My ZSH Config
 - SublimeText 3 Config
 - SublimeMerge Config
- How To Use Ubuntu Tools
 - SSH
 - * SSH connection without password
 - * Open SSH Connection
 - VNC
 - * Create password
 - * Launch x11vnc

Installation

This installation is based on Ubuntu 18.4 LTS and ROS Melodic Morenia.

Default Tools

```
sudo apt-get install git curl vim openssh-server krename rar unrar kget diffutils.

→ kate x11vnc
eche "Configure Firewall and Port for ssh"
sudo ufw allow ssh
sudo ufw enable
sudo ufw status
sudo service ssh restart
```

ZSH

```
sudo apt-get install zsh
sudo chsh -s /bin/zsh SUSER
```

Oh My ZSH

```
ad ~/Downloads
sh -c "$(curl -fsSL https://raw.github.com/robbyrussell/oh-my-zsh/master/tools/

install.sh)"
```

SublimeText 3

```
wget -q0 - https://download.sublimetext.com/sublimehq-pub.gpg | sudo apt-key add - sudo apt-get install apt-transport-https

suho "deb https://download.sublimetext.com/ apt/stable/" | sudo tee /etc/apt/

sucres.list.d/sublime-text.list
sudo apt-get update
sudo apt-get install sublime-text
```

SublimeMerge

```
wget -q0 - https://download.sublimetext.com/sublimehq-pub.gpg | sudo apt-key add - sudo apt-get install apt-transport-https

suho "deb https://download.sublimetext.com/ apt/stable/" | sudo tee /etc/apt/

sudo apt-get update
sudo apt-get install sublime-merge
```

Krusader

```
sudo apt-get install krusader
```

Yakuake

```
sudo apt-get install yakuake
```

FSearch

```
sudo add-apt-repository ppa:christian-boxdoerfer/fsearch-daily
sudo apt update
sudo apt-get install fsearch-trunk
```

Anaconda

```
~/Downloads
wget https://repo.anaconda.com/archive/Anaconda3-2019.10-Linux-x86_64.sh
bash Anaconda3-2019.10-Linux-x86_64.sh
```

QT-Creator

Visual Studio Code

```
curl https://packages.microsoft.com/keys/microsoft.asc | gpg --dearmor > packages.

--microsoft.gpg
sudo install -o root -g root -m 644 packages.microsoft.gpg /usr/share/keyrings/
sudo sh -c 'echo "deb [arch=amd64 signed-by=/usr/share/keyrings/packages.microsoft.

---gpg] https://packages.microsoft.com/repos/vscode stable main" > /etc/apt/sources.

---list.d/vscode.list'

sudo apt-get install apt-transport-https
sudo apt-get update
sudo apt-get install code # or code-insiders
```

Configuration

Oh My ZSH Config

Listing 1: ~/.zshrc additions

```
echo"#----
--" >> ~/.zshrc
echo "# Program in Path" >> ~/.zshrc
echo "#" >> ~/.zshrc
---" >> ~/.zshrc
echo "# Special zsh config" >> ~/.zshrc
echo "# Show hidden files and folders" >> ~/.zshrc
echo "setopt globdots" >> ~/.zshrc
→--" >> ~/.zshrc
    "# Goto Alias" >> ~/.zshrc
    "# Common home locations" >> ~/.zshrc
    "alias home='cd ~'" >> ~/.zshrc
   "alias root='cd /'" >> ~/.zshrc
   o "alias dtop='cd ~/Desktop'" >> ~/.zshrc
 cho "alias dwld='cd ~/Downloads'" >> ~/.zshrc
 cho "alias docs='cd ~/Documents'" >> ~/.zshrc
cho "alias www='cd /var/www/html'" >> ~/.zshrc
echo "alias workspace='cd ~/Workspace'" >> ~/.zshrc

→frontend'" >> ~/.zshrc

echo "# Common commands" >> ~/.zshrc
echo "alias o=open" >> ~/.zshrc
echo "alias ..='cd ..'" >> ~/.zshrc
echo "alias ...='cd ..; cd ..'" >> ~/.zshrc
echo "alias ....='cd ..; cd ..; cd ..'" >> ~/.zshrc
echo "# Common command shortcuts" >> ~/.zshrc
echo "alias cls=clear" >> ~/.zshrc
  ho "alias ll='ls -la'" >> ~/.zshrc
```

SublimeText 3 Config

Listing 2: ~/.zshrc additions

SublimeMerge Config

Listing 3: ~/.zshrc additions

```
echo "#Sublime Merge" >> ~/.zshrc
echo "export PATH=SPATH:/opt/sublime_merge" >> ~/.zshrc
echo "#Sublime Merge" >> ~/.bashrc
echo "export PATH=SPATH:/opt/sublime_merge" >> ~/.bashrc
```

How To Use Ubuntu Tools

SSH

SSH connection without password

Open SSH Connection

```
# Just ssh
ssh <user>@<remoteip>

# ssh with portforwarding
ssh -L <local-port>:localhost:<remote-port> <user>@<remoteip>
# ssh with vnc port forwarding
ssh -L 5900:localhost:5900 spl@<remoteip>
```

VNC

On remote PC x11vnc needs to be installed and launched. Prefereable add to startup commands

Create password

Only needed if not only localhost used.

```
x11vnc -storepasswd
```

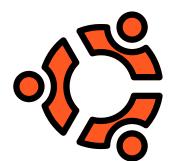
Launch x11vnc

1.4.2 Introduction

• Additional Informations

Additional Informations

- https://ubuntu.com/ Ubuntu Webpage
 - https://ubuntu.com/#download Ubuntu Download
- https://www.osboxes.org/ubuntu/ Virtual Box images
- · Additional Tools
 - ZSH
 - * Oh My ZSH
 - Sublime Text
 - Sublime Merge
 - Krusader
 - Yakuake
 - FSearch
 - Anaconda
 - QT Creator
 - Visual Studio Code
- Hitachi SDK
- * Hitachi LiDaR SDK
- * Hitachi LiDaR ROS Driver
- ROS Installation



1.5 Filesystem

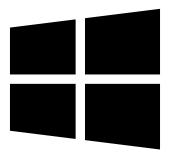


1.5. Filesystem 11

Mac



Windows



3.1 Firewall

• SSH Over FTP Port

3.1.1 SSH Over FTP Port

 $Ff\ FTP\ Port\ is\ used\ for\ SSH\ connections\ disable\ \texttt{Statefulftp}\ in\ the\ Windows\ firewall$

netsh advfirewall set global statefulftp disable

3.2 Group Policies

- Modify Policies
- See all modified Group Policies

3.2.1 Modify Policies

Search for Edit group policy

3.2.2 See all modified Group Policies

Search for rsop.msc

3.3 Registry

- Login
- DateTime
- Shell Overlay Icons
- Context Menu
- New Context Menu
- SAP Shortcut Password
- PowerPoint Options

3.3.1 Login

Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon

3.3.2 DateTime

 $\verb|HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\DateTime\Servers|\\$

3.3.3 Shell Overlay Icons

 $\label{local_MACHINE} Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Current\Version\Explorer\Gamma \Gamma \Gam$

3.3.4 Context Menu

 ${\tt Computer} \\ {\tt HKEY_CLASSES_ROOT} \\ {\tt \times} \\ {\tt Shellex} \\ {\tt ContextMenuHandlers} \\$

3.3. Registry

3.3.5 New Context Menu

Computer\HKEY_CLASSES_ROOT\

3.3.6 SAP Shortcut Password

Computer\HKEY_CURRENT_USER\Software\SAP\SAPShortcut\Security

3.3.7 PowerPoint Options

 ${\tt Computer} \\ {\tt HKEY_CURRENT_USER} \\ {\tt Software} \\ {\tt Microsoft} \\ {\tt Office} \\ {\tt 16.0} \\ {\tt PowerPoint} \\ {\tt Options} \\ {$

- * ExportBitmapResolution = DWORD 32bit => 300 (ppi)
- * AutomaticPicturesCompressionDefault = DWORD = 0

3.3. Registry



4.1 Git Commands

- Start a working area
- Work on the current change
- Examine the history and state
- Grow, mark and tweak your common history
- Collaborate

4.1.1 Start a working area

| Command | Description |
|---------|----------------------------------------------------------------|
| clone | Clone a repository into a new directory |
| init | Create an empty Git repository or reinitialize an existing one |

4.1.2 Work on the current change

| Command | Description |
|---------|-------------------------------------------------------|
| add | Add file contents to the index |
| mv | Move or rename a file, a directory, or a symlink |
| reset | Reset current HEAD to the specified state |
| rm | Remove files from the working tree and from the index |

4.1.3 Examine the history and state

| Command | Description |
|---------|-------------------------------|
| log | Show commit logs |
| show | Show various types of objects |
| status | Show the working tree status |

4.1.4 Grow, mark and tweak your common history

| Command | Description |
|----------|-------------------------------------------------------------|
| branch | List, create, or delete branches |
| checkout | Switch branches or restore working tree files |
| commit | Record changes to the repository |
| diff | Show changes between commits, commit and working tree, etc |
| merge | Join two or more development histories together |
| rebase | Reapply commits on top of another base tip |
| tag | Create, list, delete or verify a tag object signed with GPG |

4.1.5 Collaborate

| Command | Description |
|---------|--------------------------------------------------------------------|
| 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 |

4.2 Git Flow

• Branches

4.2.1 Branches

- $\bullet \ \, \text{master-protected branch-Production releases}$
- develop protected branch main developement merge of all feature branches
- feature/* for each feature a separate feature branch is created fork from develop
- release preparing development branch for release on master branch, mainly for bugfixes
- hotfix quick and dirty hotfix directly into develop and master branch

4.2. Git Flow 17

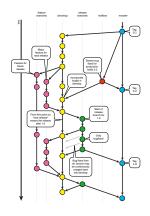


Fig. 1: Git Flow

4.3 Git General

- Global setup
 - Check setup
 - Save Credentials
 - Not verify https Certificates
- Git Repo Creation / Cloning
 - Create new repository
 - Clone existing repository
 - Existing folder
 - Switch to new Remote
 - Get Remote Information
 - Change Push Remote URL
- Git Repo information
- Add Files
- Checkout
- Push
- Branch
 - Merge

4.3. Git General

4.3.1 Global setup

```
git config --global user.name "Silvan Zahno"
git config --global user.email "silvan.zahno@hevs.ch"
```

Check setup

```
git config --list
```

Save Credentials

```
git config credential.helper store
```

Not verify https Certificates

```
git config --global http.sslVerify false
```

4.3.2 Git Repo Creation / Cloning

Create new repository

```
git init
```

Clone existing repository

```
git clone git@gitlab.hevs.ch:course/ElN/eln_labs.git
cd eln_labs
touch README.md
git add README.md
git commit -m "add README"
git push -u origin master
```

Existing folder

```
existing_folder
git init
git remote add origin git@gitlab.hevs.ch:course/ElN/eln_labs.git
git add .
git commit -m "Initial commit"
git push -u origin master
```

4.3. Git General

Switch to new Remote

```
existing_repo
git remote rename origin old-origin
git remote add origin git@gitlab.hevs.ch:course/ElN/eln_labs.git
git push -u origin --all
git push -u origin --tags
```

Get Remote Information

```
git remote show origin
```

Change Push Remote URL

```
git remote set-url --push <new_repo_push_url>
```

4.3.3 Git Repo information

```
# Status about current files ion folder
git status

# Status about last commits
git log --oneline
```

4.3.4 Add Files

```
# Stage a File
git add README.md
# Commit file
git commit -m "Initial commit, add README file"
```

4.3.5 Checkout

```
# Checkout certain commit
git checkout e006db0 -b inspectingPrev

# Checkout given branch
git chekout master
```

4.3.6 Push

```
git push origin master
```

4.3. Git General 20

4.3.7 Branch

```
# Create new branch
git branch dev_branch_1

# List all existing branches
git branch

# Checkout certain branch
git branch dev_branch_1

# Delete certain branch
git branch -d dev_branch_1
```

Merge

```
# Checkout branch you want to merge into
git checkout master
# Merge the two branches
git merge dev_branch_1
```

4.4 Git Submodules

- Clone Repo with submodules
- Pull changes
 - Pull all changes in the repo including changes in the submodules
 - Pull all changes for the submodules
- Add submodule and define the master branch as the one you want to track
- Move Submodule

4.4.1 Clone Repo with submodules

```
git clone --recursive [URL to Git repo]
```

4.4.2 Pull changes

Pull all changes in the repo including changes in the submodules

```
git pull --recurse-submodules
```

4.4. Git Submodules 21

Pull all changes for the submodules

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

4.4.3 Add submodule and define the master branch as the one you want to track

```
git submodule add -b master [URL to Git repo]
git submodule init
```

4.4.4 Move Submodule

```
git mv a b
```

4.4. Git Submodules 22

5

Jupyter



5.1 Common Functions

- Common Jupyterlab and Nodejs functions
 - install nvm
 - Install nodejs via conda
 - update npm
 - Rebuild Jupyterlab
 - Remove nodejs and npm
- Auto import of Libraries
- Check

5.1.1 Common Jupyterlab and Nodejs functions

install nvm

```
https://github.com/creationix/nvm
curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.34.0/install.sh     zsh
### Install latest nodejs
nvm install node # "node" is an alias for the latest version
```

Install nodejs via conda

```
conda install -c conda-forge nodejs
```

update npm

```
sudo npm install -g npm
```

Rebuild Jupyterlab

```
jupyter lab build
```

Remove nodejs and npm

```
conda remove nodejs npm
```

5.1.2 Auto import of Libraries

- Navigate to ~/.ipython/profile_default
- Create a folder called startup if it's not already there
- Add a new Python file called start.py
- Put your favorite imports in this file
- Launch IPython or a Jupyter Notebook and your favorite libraries will be automatically loaded every time!

Example start.py

```
import pandas as po
   import numpy as np
   # Pandas options
   pd.options.display.max_columns = 30
   pd.options.display.max_rows = 20
   from IPython import get_ipython
   # If in ipython, load autoreload extension
11
   if 'ipython' in globals
12
13
        ipython.magic('load_ext autoreload')
ipython.magic('autoreload 2')
14
15
16
   # Display all cell outputs in notebook
17
   from IPython.core.interactiveshell import InteractiveShell
18
19
   # Visualization
21
   import plotly.plotly as py
22
   import plotly.graph_objs as go
   from plotly offline import iplot,
24
25
```

(continues on next page)

(continued from previous page)

```
27  cf.go_offline(connected=True)
28  cf.set_config_file(theme='pearl')
29  
30  print('Your favorite libraries have been loaded.')
```

5.1.3 Check

Confirm that Libraries are loaded with

```
globals()
```

5.2 Extensions

• Installed extensions

5.2.1 Installed extensions

```
jupyter labextension list
```

5.3 General

- Anaconda / Conda Update
- nbconvert
 - Convert to python for linting
 - Convert to html
 - Convert to pdf

5.3.1 Anaconda / Conda Update

```
# Update all Conda packages
conda update --all

# Update Anaconda only
conda update conda
conda update anaconda
```

5.2. Extensions 25

5.3.2 nbconvert

Converts jupyter notebook to other formats

```
jupyter nbconvert --to <format> notebook.ipynb
```

formats are:

- --to html HTML
 - --template full (default)
 - --template basic
- ullet --to latex LaTeX
 - --template article(default)
 - --template report
 - --template basic
- --to pdf PDF
 - --template article (default)
 - --template report
 - --template basic
- --to sildes Reveal.js HTML slideshow
- --to markdown Markdown
- --to rst -reStructuredText
- --to script executable script (.py)
- --to notebook -

Convert to python for linting

```
jupyter nbconvert --to script test.ipynb
```

Convert to html

```
jupyter nbconvert --to html test.ipynb
```

Convert to pdf

needs Latex installed see: LaTeX

```
jupyter nbconvert --to latex test.ipynb
```

5.3. General 26

5.4 Plotly Dash

- Install Plotly Dash
 - Checking Versions
 - Getting help
 - Jupyter integration install
 - To rebuild the package and the JupyterLab app
 - Additional Packages
 - Install Dash DAQ

5.4.1 Install Plotly Dash

```
# The core dash backend
pip install dash==0.43.0
# DAQ components (newly open-sourced!)
pip install dash-daq==0.1.0
```

Checking Versions

```
import dash_core_components
print(dash_core_components.__version__)
```

Getting help

```
help(dcc.Dropdown)
```

Jupyter integration install

```
git clone https://github.com/plotly/jupyterlab-dash

d jupyterlab-dash

npm install

npm run build

jupyter labextension link .

~/anaconda3/bin/./python -m pip install -e .
```

To rebuild the package and the JupyterLab app

```
npm run build
jupyter lab build
```

5.4. Plotly Dash

Additional Packages

```
pip install aiohttp
pip install django_plotly_dash
pip install jupyter_plotly_dash
```

Install Dash DAQ

```
pip install dash_daq
```

5.5 Installation

- My Extension list
 - All in one install
 - Add install R to jupyter
 - Add install pandoc and inkscape to conda
- Install Python Additional Stuff
 - Graphviz
 - Install python Libraries
- Problems
 - Anaconda Navigator not starting
- Install Plotly and Plotly Express
- Better PDF Export

see also jupyter config

5.5.1 My Extension list

```
jupyter labextension install @jupyter-widgets/jupyterlab-manager
 jupyter labextension install @jupyterlab/statusbar-extension
  jupyter labextension install @jupyterlab/geojson-extension
  jupyter labextension install @jupyterlab/git
  pip install -e git+https://github.com/jupyterlab/jupyterlab-git.git#egg=jupyterlab_
   ⊶ait
  jupyter serverextension
                           enable --py jupyterlab_git --sys-prefix
   jupyter labextension install @jupyterlab/plotly-extension
   jupyter labextension install @jupyterlab/toc
   jupyter labextension install @deathbeds/jupyterlab_graphviz
   jupyter labextension install @ryantam626/jupyterlab_sublime
10
   jupyter labextension install jupyter-matplotlib
   jupyter labextension install jupyterlab_bokeh
   jupyter labextension install @mflevine/jupyterlab_html
   jupyter labextension install jupyterlab-drawio
   jupyter labextension install jupyterlab-flake8
   # jupyter labextension install jupyterlab_nbmetadata
  jupyter labextension install jupyterlab_hidecode
  jupyter labextension install @krassowski/jupyterlab_go_to_definition
  jupyter labextension install @lckr/jupyterlab_variableinspector
```

5.5. Installation 28

All in one install

```
jupyter labextension install @lckr/jupyterlab_variableinspector @krassowski/

jupyterlab_go_to_definition @jupyter-widgets/jupyterlab-manager @jupyterlab/

statusbar-extension @jupyterlab/geojson-extension @jupyterlab/plotly-extension.

jupyterlab/toc @deathbeds/jupyterlab_graphviz jupyterlab_hidecode @ryantam626/

jupyterlab_sublime jupyter-matplotlib jupyterlab_bokeh @mflevine/jupyterlab_html.

jupyterlab-drawio jupyterlab-flake8

pip install -e git+https://github.com/jupyterlab/jupyterlab-git.git#egg=jupyterlab_

git
jupyter serverextension pnable --py jupyterlab_git --sys-prefix
```

Add install R to jupyter

```
conda install -c r r-essentials
```

Add install pandoc and inkscape to conda

```
conda install -c conda-forge pandoc conda install -c conda-forge inkscape
```

5.5.2 Install Python Additional Stuff

Graphviz

Install Graphviz from https://graphviz.gitlab.io/download/ put Graphviz/bin in your PATH

```
pip install graphviz
```

Install python Libraries

```
pip install pixiedust
pip install SchemDraw
pip install nbwavedrom
pip install flake8
pip install pyflakes
pip install nbconvert
pip install watermark
```

oneline

pip install pixiedust SchemDraw nbwavedrom flake8 pyflakes nbconvert graphviz

5.5. Installation 29

5.5.3 Problems

Anaconda Navigator not starting

When starting anaconda-navigator produces the following error.

```
$ anaconda-navigator.exe
   Traceback (most recent call last):
    File "c:\Users\silvan.zahno\AppData\Local\Continuum\anaconda3\lib\site-packages\
   →qtpy\__init__.py", line 202, in <module>
       from PySide import __version__ as PYSIDE_VERSION # analysis:ignore
   ModuleNotFoundError: No module named 'PySide'
   During handling of the above exception, another exception occurred:
   Traceback (most recent call last):
    File "c:\Users\silvan.zahno\AppData\Local\Continuum\anaconda3\Scripts\anaconda-
10
   →navigator-script.py", line 6, in <module>
       from anaconda_navigator.app.main import main
11
    File "c:\Users\silvan.zahno\AppData\Local\Continuum\anaconda3\lib\site-packages\
12
   →anaconda_navigator\app\main.py", line 22, in <module>
       from anaconda_navigator.utils.conda import is_conda_available
13
    File "c:\Users\silvan.zahno\AppData\Local\Continuum\anaconda3\lib\site-packages\
   →anaconda_navigator\utils\__init__.py", line 15, in <module>
15
       from qtpy.QtGui import QIcon
    File "c:\Users\silvan.zahno\AppData\Local\Continuum\anaconda3\lib\site-packages\
           __init__.py", line 208, in <module>
   →qtpy\
       raise PythonQtError('No Qt bindings could be found')
17
   qtpy.PythonQtError: No Qt bindings could be found
18
```

```
pip uninstall PyQt5
conda update conda
conda update anaconda-navigator
anaconda-navigator.exe
```

5.5.4 Install Plotly and Plotly Express

```
conda install -c plotly_plotly_express plotly-orca
```

5.5.5 Better PDF Export

```
sudo apt-get install texlive-xetex
pip install jupyter_contrib_nbextensions
pip install cite2c
```

5.5. Installation 30

Pandoc



6.1 Pandoc

- Additional Arguments
 - Highlight Styles
 - PDF Output
 - * For my template needed packages
 - Template
 - * Windows
 - * Linux

If you need to convert files from one markup format into another, pandoc is your swiss-army knife.

- Pandoc Online
- Pandoc Download

6.1.1 Additional Arguments

Highlight Styles

```
# List all Highlight Styles
pandoc --list-highlight-styles
pygments
tango
espresso
zenburn
```

(continues on next page)

(continued from previous page)

```
kate
monochrome
breezedark
haddock

## Pandoc Argument
--highlight-style breezedark
```

PDF Output

```
--pdf-engine=xelatex
```

For my template needed packages

- cm-super
 - Error no Scalable font
- koma-script
 - ! LaTeX Error: File scrartcl.cls not found.

Template

Latex Template needs to be in the following folders

Windows

```
{\tt C:/\backslash Users/\backslash susername}\backslash App {\tt Data/\backslash Roaming\backslash pandoc/\backslash templates}
```

Linux

```
~/.pandoc/templates/
```

```
--template=<template>.latex
```

6.1. Pandoc 32

AutoHotKey AHK



7.1 Tips & Tricks

- Comment
- Performance and Compatility
- Warnings
- Enable Regex for Title mach Mode
- Tray Icon and ToolTip
- Examples
 - For Win10 Hibernate
 - For Win10 Sleep
 - Home and End Hotkey
- Check for AHK Version and output message
- Supend a script via Hotkey

My ahk scripts can be found in the config repo

7.1.1 Comment

```
;-----;
;-- Comment
;--
```

7.1.2 Performance and Compatility

```
; Recommended for performance and compatibility with future AutoHotkey releases
```

7.1.3 Warnings

```
; Enable warnings to assist with detecting common errors
```

7.1.4 Enable Regex for Title mach Mode

```
SetTitleMatchMode,RegEx ; then IfWinExist, Total Commander.*
```

7.1.5 Tray Icon and ToolTip

```
Menu, TRAY, Icon, Favicon.ico
Menu, TRAY, Tip, Tooltip Text
```

7.1.6 Examples

For Win10 Hibernate

```
; Wait for Hotkey
; Ctrl + Win + Alt + l
; Send Hotkey
; Ctrl + Win + x + u + s
^*<!l::Send #xuh</pre>
```

For Win10 Sleep

```
; Wait for Hotkey
; Ctrl + Win + l
; Send Hotkey
; Win + x + u + s
^#l::Send #xus
```

Home and End Hotkey

```
; Ctrl + Left
^Left::Send (Home)
; Ctrl + Right
^Right::Send (End)
```

7.1. Tips & Tricks 34

7.1.7 Check for AHK Version and output message

7.1.8 Supend a script via Hotkey

```
fl::suspend
```

7.2 Key Definitions

- Raw Keys
- Double Keypress Detection

7.2.1 Raw Keys

```
^ ; Ctrl
# ; Win

| ; Right Win
| ; Alt
| ; Right Altt

| ; Right Altt

| ; Right Shift

<pr
```

7.2.2 Double Keypress Detection

Alt Key in the example

```
~Alt::
DoubleAlt := A_PriorHotkey ="~Alt" AND A_TimeSincePriorHotkey < 400
Sleep 0
KeyWait Alt ; This prevents the keyboard's auto-repeat feature from interfering.
return
```

7.2. Key Definitions

7.3 Tips & Tricks

- Comment
- Performance and Compatility
- Warnings
- Enable Regex for Title mach Mode
- Tray Icon and ToolTip
- Examples
 - For Win10 Hibernate
 - For Win10 Sleep
 - Home and End Hotkey
- Check for AHK Version and output message
- Supend a script via Hotkey

My ahk scripts cna be found in the config repo

7.3.1 Comment

```
;-----;-- Comment;--
```

7.3.2 Performance and Compatility

; Recommended for performance and compatibility with future AutoHotkey releases $\# \mathsf{Norm}_{\mathsf{N}}$

7.3.3 Warnings

```
; Enable warnings to assist with detecting common errors #Warn
```

7.3.4 Enable Regex for Title mach Mode

```
SetTitleMatchMode,RegEx ; then
IfWinExist, Total Commander.*
```

7.3. Tips & Tricks 36

7.3.5 Tray Icon and ToolTip

```
Menu, TRAY, Icon, Favicon.ico
Menu, TRAY, Tip, Tooltip Text
```

7.3.6 Examples

For Win10 Hibernate

```
; Wait for Hotkey
; Ctrl + Win + Alt + 1
; Send Hotkey
; Ctrl + Win + x + u + s
^#<!1::Send #xuh</pre>
```

For Win10 Sleep

```
; Wait for Hotkey
; Ctrl + Win + l
; Send Hotkey
; Win + x + u + s
^*l::Send #xus
```

Home and End Hotkey

```
; Ctrl + Left
^Left::Send (Home)
; Ctrl + Right
^Right::Send (End)
```

7.3.7 Check for AHK Version and output message

7.3.8 Supend a script via Hotkey

```
fl::suspend
```

7.3. Tips & Tricks



Python



8.1 Docstring

- Python begin file
- Variables
- Functions
 - Function with types
 - Function with pep484 type annotations
 - Function modules level
 - Function other examples
- Class

8.1.1 Python begin file

```
#!/usr/bin/python3
# -*- coding: utf-8 -*-
"""Example NumPy style docstrings.

This module demonstrates documentation as specified by the `NumPy
Documentation HOWTO`_. Docstrings may extend over multiple lines. Sections
are created with a section header followed by an underline of equal length.
```

(continues on next page)

```
Examples can be given using either the 'Example' or 'Examples' sections. Sections support any reStructuredText formatting, including literal blocks:

$ python example_numpy.py

Section breaks are created with two blank lines. Section breaks are also implicitly created anytime a new section starts. Section bodies *may* be indented:

Notes

This is an example of an indented section. It's like any other section, but the body is indented to help it stand out from surrounding text.

If a section is indented, then a section break is created by resuming unindented text.

Attributes

module_level_variable1 : int

Module level variables may be documented in either the 'Attributes' section of the module docstring, or in an inline docstring immediately following the variable.

Either form is acceptable, but the two should not be mixed. Choose one convention to document module level variables and be consistent with it.

.._NumPy Documentation HOWTO:

https://github.com/numpy/numpy/blob/master/doc/HOWTO_DOCUMENT.rst.txt

"""
```

8.1.2 Variables

```
module_level_variable1 = 12345

module_level_variable2 = 98765
"""int: Module level variable documented inline.

The docstring may span multiple lines. The type may optionally be specified on the first line, separated by a colon.
"""
```

8.1.3 Functions

Function with types

```
def function_with_types_in_docstring(param1, param2):
    """Example function with types documented in the docstring.

    `PEP 484`_ type annotations are supported. If attribute, parameter, and return types are annotated according to `PEP 484`_, they do not need to be included in the docstring:

Parameters
------
param1 : int
    The first parameter.
param2 : str
    The second parameter.

Returns
------
bool
    True if successful, False otherwise.

.. _PEP 484:
    https://www.python.org/dev/peps/pep-0484/
    """
```

Function with pep484 type annotations

```
def function_with_pep484_type_annotations(paraml: int, param2: str) -> bool:
    """Example function with PEP 484 type annotations.

The return type must be duplicated in the docstring to comply with the NumPy docstring style.

Parameters
------
param1
    The first parameter.
param2
    The second parameter.

Returns
-----
bool
    True if successful, False otherwise.
"""
```

Function modules level

```
def module_level_function(param1, param2=None, *args, **kwargs)
```

(continues on next page)

```
raise ValueError('param1 may not be equal to param2')
return True
```

Function - other examples

(continues on next page)

```
def __init__(self, msg, code):
    self.msg = msg
    self.code = code
```

8.1.4 Class

```
self.attr1 = param1
self.attr2 = param2
self.attr3 = param3 #: Doc comment *inline* with attribute
#: list of str: Doc comment *before* attribute, with type specified
self.attr4 = ["attr4"
```

(continues on next page)

```
@readwrite_property.setter
def __special_without_docstring__(self);
```

(continues on next page)

8.2 General

```
• flake8
— .flake8
```

Python samples

8.2.1 flake8

```
python -m flake8 test.py
```

.flake8

Flake8 configuration file is formated at ini File. and located at:

- Linux ~/.config/flake8
- Windows %userprofile%\.flake8

see my config .flake8

```
[flake8]
max-line-length = 200

ignore =
    #E501: Line too long
    E501

#E722 do not use bare 'except'
    E722

#W504 line break after binary operator (one has to disable one of the W503/W504」
→pair)
    W504

#W391 blank line at end of file
```

(continues on next page)

8.2. General 45

```
w391

exclude =
    .git,
    __pycache__,
    docs/source/conf.py,
    old,
    build,
    dist
```

8.3 Flake 8

• .flake8

Python samples

```
python -m flake8 test.py
```

8.3.1 .flake8

Flake8 configuration file is formated at ini File. and located at:

- Linux ~/.config/flake8
- Windows %userprofile%\.flake8

see my config .flake8

```
[flake8]
max-line-length = 200

ignore =
    #E501: Line too long
E501

#E722 do not use bare 'except'
E722

#W504 line break after binary operator (one has to disable one of the W503/W504_
→pair)
W504

#W391 blank line at end of file
W391

exclude =
    .git,
    __pycache__,
    docs/source/conf.py,
    old,
    build,
    dist
```

8.3. Flake 8 46

8.4 PIP

- Admin
- Package
- Create requirements.txt

8.4.1 Admin

```
# Show pip help
pip --help

# Show installed pip version
pip --version

# Update pip (Linux)
pip install --upgrade pip

# Update pip (Windows)
python -m pip install --upgrade pip
```

8.4.2 Package

```
# Search a package
pip search <packagename>

# See package version
pip show <packagename>

# See all installed packages
pip list

# Install
pip install <packagename>
pip install -I <packagename>==<package version>
pip install -I ipython==5.4.0

# Uninstall
pip uninstall <packagename>
```

8.4.3 Create requirements.txt

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

8.4. PIP 47

Licenses



9.1 All rights reserved

```
Copyright (c) 2019 - tschinz

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
```

9.2 MIT

```
Copyright (c) 2019 tschinz

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
```

(continues on next page)

```
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
```

9.3 WTFPL

```
DO WHAT THE FUCK YOU WANT TO PUBLIC LICENSE

Version 2, December 2004

Copyright (C) 2004 Sam Hocevar <sam@hocevar.net>

Everyone is permitted to copy and distribute verbatim or modified copies of this license document, and changing it is allowed as long as the name is changed.

DO WHAT THE FUCK YOU WANT TO PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION
```

9.3. WTFPL 49

ROS - Robot Operating System



10.1 Introduction

• Philosophy

ROS aka Robotic Operating System is not a OS itself but a framework and middleware.

- Software Framework for programming robots
- Prototype from Standfort AI Research Institute and created by Willow Garage in 2007
- Since 2013 maintained by the Open Source Robotics Foundation (OSRF)
- Consists of infrastrucutre, tools, capabilities and a ecosystem

Table 1: Source : ROS Tutorial #1 https://www.youtube.com/watch?v=9U6GDonGFHw&t=1s

| Advantages | Disadvantages |
|---------------------------------------------------------|-------------------------------------------------------|
| Provides lots of infrastructure, tools and capabilities | Approaching maturity, but still changing |
| Easy to try other people's work and shar your own | Security and scalability are not first-class concerns |
| Large community | OSes other than Ubuntu Linux are not well supported |
| Free, open source, BSD license | |
| Great for open-source and researchers | Not great for mission-critical tasks |



Fig. 1: ROS Equation

| Plumbing | Tools | Capabilities | Ecosystem | |
|-----------------------------|--------------------------|--------------|-----------------------|--|
| Process management | Simulation | Control | Package organization | |
| Inter-process communication | Visualization | Planning | Software distribution | |
| Device drivers | Graphical user interface | Perception | Documentation | |
| | Data logging | Mapping | Tutorials | |
| | | Manipulation | | |

10.1.1 Philosophy

- Peer to peer Individual programs communicate over defined API (ROS messages, services, etc.).
- Distributed Programs can be run on multiple computers and communicate over the network.
- **Multi-lingual** ROS modules can be written in any language for which a client library exists (C++, Python, MATLAB, Java, etc.).
- Thin The ROS conventions encourage contributors to create standalone libraries and then wrap those libraries so they can send and receive messages to and from other ROS modules.
- Free and open source The core of ROS is released under the permissive BSD license, which allows commercial and noncommercial use.

10.2 Basics

- · Coding Rules
- Standard Unit in ROS
- Master
- Publisher and Subscribers
- Catkin Overview
 - src/Folder
 - build/Folder
 - devel/Folder
 - install/Folder
- Messages

10.2.1 Coding Rules

The following rules apply when writing code with ROS.

Table 2: ROS Robot Programming by TurtleBot3 Developers, section 7.1.3

| Туре | Naming Rule | Example |
|------------------|--------------|----------------------------------------------------|
| Package | under_scored | first_ros_package |
| Topic, Service | under_scored | raw_image |
| File | under_scored | turtlebot3_fake.cpp |
| Namespace | under_scored | ros_awesome_package |
| Variable | under_scored | string table_name; |
| Type | camelCased | typedef int32_t PropertiesNumber; |
| Class | camelCased | class UrlTable |
| Structure | camelCased | struct UrlTableProperties |
| Enumeration Type | camelCased | enum ChoiceNumber |
| Function | camelCased | addTableEntry() |
| Method | camelCased | <pre>void setNumEntries(int32_t_num_entries)</pre> |
| Constant | ALL_CAPITALS | const uint8_t DAYS_IN_A_WEEK = T; |
| Marco | ALL_CAPITALS | #define PI_ROUNDED 3.0 |

10.2.2 Standard Unit in ROS

Table 3: Source : ROS Robot Programming by TurtleBot3 Developers, section 7.1.1

| Quantity | Unit |
|-------------|----------|
| Length | Meter |
| Mass | Kilogram |
| Time | Second |
| Current | Ampere |
| Angle | Radian |
| Frequency | Hertz |
| Force | Newton |
| Power | Watt |
| Voltage | Volt |
| Temperature | Celsius |

10.2.3 Master

ROS master is a Server tracking all network addresses of all nodes. In addition to network addresses it also tracks other information like parameters. All nodes must know the network address of the master on startup ROS_MASTER_URI.

A master can be started with the roscore command or a roslaunch will also start a master if it doesn't exists already.

roscore

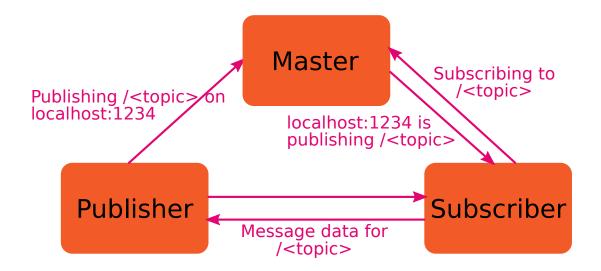


Fig. 2: ROS Master Publisher Slave

10.2.4 Publisher and Subscribers

With help of the master, publisher and subscriber establish a peer-to-peer connection. All nodes must know the network address of the master on startup ROS_MASTER_URI.

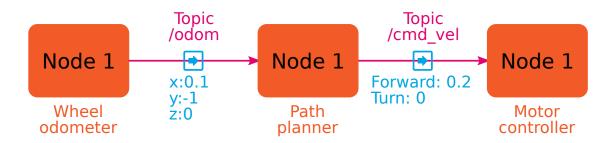


Fig. 3: ROS Publisher Slave

- Any node can publish a message to any topic
- Any node can subscribe to any topic
- Multiple nodes can publish to the same topic
- Multiple nodes can subscribe to the same topic
- A node can publish to multiple topics
- A node can subscribe to multiple topics

10.2.5 Catkin Overview

src/ Folder

Location for create and clone new packages

The command catkin_make searches only in the src/ folder for packages and builds them

It is a good practice to clone the ros packages into a different folder e.g. ~/git/<package_name> and create a symlink into you catkin workspace

```
ls -s ~/git/<package_name>/ ~/catkin_ws/src/
```

build/Folder

catkin_make create buixld files and intermediate cache CMake files inside the build/ folder.

devel/Folder

catkin_make builds each package, if successful, the target executable le is created. Executables are stored inside the devel/ folder. Current workspace packages can be access by the command line if the following command is used:

```
# for bash
source ~/<workspace_name>/devel/setup.bash
# for zsh
source ~/<workspace_name>/devel/setup.zsh
```

It is beneficial to add this the the ~/.bashrc or ~/.zshrc file.

In addition there is the catkin_tools program which simplifies the use.

See dedicated page: Catkin Tools

install/Folder

After building the executables in the devel/ folder, this executables can be install by:

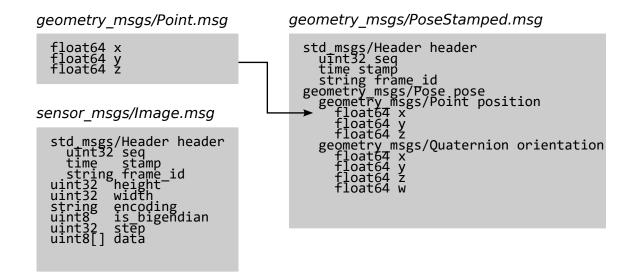
```
catkin_make install
```

See also:

• http://wiki.ros.org/catkin/workspaces#Catkin_Workspaces

10.2.6 Messages

- · Serialization format for structured data
- Defined in a .msq file
- Compiled to C++/Python classes before using them
- more info https://wiki.ros.org/Messages



10.3 Books summary

- Topics
- *SLAM* (*Simultaneous localization and modeling*)
- TF (Transform Frames)
- QR code reader
- 3D
- BAG recording
- Odometry and navigation
- Point Clouds
- OpenCV

10.3.1 Topics

Basic topics such as workspace description, packages and nodes creation can be found in most of the book mentioned in this summary. They are not part of this summary since it focuses on more advanced topics. Tutorials to understand those topics are available in books or on the ROS wiki.

This summary lists all the books we have related to ROS, and some more specific PDF documents. Storage of the referenced documents:

- books : ros/books/
 - Effective_Robotics_Programming_with_ROS_3E.pdf
 - Learning_ROS_for_Robotics_Programming_2E.pdf
 - Mastering_ROS_for_Robotics_Programming.pdf
 - Programming_Robots_with_ROS.pdf
 - $\hbox{\color{red} Programming_Robots_with_ROS-A_Practical_Introduction_to_the_Robot_Operating_System.pdf}$
 - Robot_Operating_System_for_Absolute_Beginners.pdf

- ROS_Robot_Programming.pdf
- ROS_Robotics_By_Example.pdf
- ROS_Robotics_By_Example_2E.pdf
- Teach_ROS_with_No_Hassle_2E.pdf
- other documents : ros/slides/
 - octomap.pdf
 - ros-ethz-1.pdf
 - ros-ethz-2.pdf
 - ros-ethz-3.pdf
 - ros-ethz-4.pdf
 - ros-ethz-5a.pdf
 - ros-ethz-5b.pdf
 - ros-ethz-5c.pdf
 - ros-misc.pdf
 - ros-tf.pdf
 - ros-tf-2.pdf

10.3.2 SLAM (Simultaneous localization and modeling)

• Mastering_ROS_for_Robotics_Programming.pdf page 146

10.3.3 TF (Transform Frames)

- Effective_Robotics_Programming_with_ROS_3E.pdf page 171
- Learning_ROS_for_Robotics_Programming_2E.pdf page 305

10.3.4 QR code reader

• TODO

10.3.5 3D

- Effective_Robotics_Programming_with_ROS_3E.pdf page 120
- Learning_ROS_for_Robotics_Programming_2E.pdf page 143
- Mastering_ROS_for_Robotics_Programming.pdf page 265

10.3.6 BAG recording

- Effective_Robotics_Programming_with_ROS_3E.pdf page 128
- Learning_ROS_for_Robotics_Programming_2E.pdf page 120

10.3.7 Odometry and navigation

- Effective_Robotics_Programming_with_ROS_3E.pdf page 179
- Learning_ROS_for_Robotics_Programming_2E.pdf page 303
- Mastering_ROS_for_Robotics_Programming.pdf page 140

10.3.8 Point Clouds

- Effective_Robotics_Programming_with_ROS_3E.pdf page 394
- Learning_ROS_for_Robotics_Programming_2E.pdf page 231
- Mastering_ROS_for_Robotics_Programming.pdf page 251

10.3.9 OpenCV

- Effective_Robotics_Programming_with_ROS_3E.pdf page 359
- Mastering_ROS_for_Robotics_Programming.pdf page 250

10.4 Catkin Tools

- Catkin build system
 - Installation Catkin Tools
- Cheat Sheet
 - Initialize Workspaces
 - Configuring Workspaces
 - Building Packages
 - Cleaning Build Products

10.4.1 Catkin build system

This Python package provides command line tools for working with the catkin meta-buildsystem and catkin workspaces. These tools are separate from the Catkin CMake macros used in Catkin source packages. It has to be installed separately.

• https://catkin-tools.readthedocs.io/

10.4. Catkin Tools 57

Installation Catkin Tools

10.4.2 Cheat Sheet

This is a non-exhaustive list of some common and useful invocations of the catkin command. All of the commands which do not explicitly specify a workspace path (with --workspace) are assumed to be run from within a directory contained by the target workspace. For thorough documentation, please see the chapters on each verb.

Initialize Workspaces

Initialize a workspace with a default layout (src/build/devel) in the *current* directory:

```
catkin init
catkin init --workspace .
catkin config --init
mkdir src && catkin build
```

... with a default layout in a different directory:

```
catkin init --workspace /tmp/path/to/my_catkin_ws
```

... which explicitly extends another workspace:

```
catkin config --init --extend /opt/ros/indigo
```

Initialize a workspace with a source space called other_src:

```
catkin config --init --source-space other_src
```

... or a workspace with **build**, **devel**, and **install space** ending with the suffix _alternate:

```
catkin config --init --space-suffix _alternate
```

Configuring Workspaces

View the current configuration:

```
catkin config
```

Setting and unsetting CMake options:

```
catkin config --cmake-args -DENABLE_CORBA=ON -DCORBA_IMPLEMENTATION=OMNIORB
```

10.4. Catkin Tools 58

```
catkin config --no-cmake-args
```

Toggle installing to the specified **install space**:

```
catkin config --install
```

Building Packages

Build all the packages:

catkin build

... one at a time, with additional debug output:

catkin build -p 1

... and force CMake to re-configure for each one:

catkin build --force-cmake

Build a specific package and its dependencies:

`catkin build <package_name>

... or ignore its dependencies:

catkin build <package_name> --no-deps

Build the package containing the current working directory:

catkin build --this

... but don't rebuild its dependencies:

catkin build --this --no-deps

Build packages with additional CMake args:

catkin build --cmake-args -DCMAKE_BUILD_TYPE=Debug

... and save them to be used for the next build:

catkin build --save-config --cmake-args -DCMAKE_BUILD_TYPE=Debug

Build all packages in a given directory:

```
catkin build $(catkin list -u /path/to/folder)
```

... or in the current folder:

```
catkin build $(catkin list -u .)
```

10.4. Catkin Tools 59

Cleaning Build Products

Blow away the build, devel, and install spaces (if they exist):

catkin clean

... or just the **build space**:

catkin clean --build

... or just clean a single package:

catkin clean PKGNAME

... or just delete the build directories for packages which have been disabled or removed:

catkin clean --orphans

10.5 Commandline Commands

- Commandline Variables
- Useful commands
 - ROS tools
 - * roscore
 - * rosversion
 - * rosparam
 - * rosnode
 - * rostopic
 - * roslaunch
 - * rosrun
 - * rosservice
 - * rosbag
 - * rosmsg
 - * Other Commands
 - Catkin
 - * Create Package
 - * Build
 - * Install
 - * Python modules
- Update services with RQT

10.5.1 Commandline Variables

```
cho $<variable_name>
                            # To display value
ROS_DISTRO
                             # Distro name e.g. melodic
ROS ETC DIR
ROS_LISP_PACKAGE_DIRECTORIES # common-lisp folder e.g. ~/catkin_ws/devel/share/
⇔common-lisp
ROS HOSTNAME
                            # ros hostname e.g. localhost
                            # ros master url e.g. http://localhost:11311
ROS_MASTER_URI
ROS_PACKAGE_PATH
                            # package path's e.g. ~/catkin_ws/src:/opt/ros/$(ROS_
→DISTRO)/share
ROS_PYTHON_VERSION
                           # python version 2 or 3 e.g. 2
ROS_ROOT
                            # ros installation e.g. /opt/ros/$(ROS_DISTRO)/share/
⇔ros
ROS_VERSION
                            # ros version 1 or 2 e.g. 1
```

10.5.2 Useful commands

ROS tools

roscore

Launch ROS master core

```
roscore
```

rosversion

```
rosversion -d # Print ROS distro name rosversion <package_name> # Print package vrosnode
```

${\tt rosparam}$

Nodes use the parameter server to store and retrieve parameters at runtime.

http://wiki.ros.org/rosparam

```
rosparam list  # list parameter names
rosparam get /<parameter_name> <value>  # set parameter
rosparam get /<parameter_name>  # get parameter
rosparam delete /<parameter_name>  # delete parameter
rosparam dump <file>  # dump parameter to file
rosparam load  # load parameter from file
```

rosnode

Work with nodes

rostopic

Work with topics

```
rostopic list
rostopic info /<topic_name>  # print information about active topic
rostopic echo /<topic_name>  # print messages to screen
rostopic pub /<topic_name> msg/MessageType "data:value" # pubish data to topic

rostopic type /<topic_name>  # print topic or field type
rostopic find <type>  # find topics by type
rostopic bw /<topic_name>  # display bandwidth used by topic
rostopic hz /<topic_name>  # display publishing rate of topic
```

roslaunch

To start a launch file which can contain multiple nodes.

```
roslaunch <ros_pkg_name> <launch_file_name> # Launch ros launch file
```

rosrun

To run a node

rosservice

Work with services

```
rosservice list # list active services
rosservice info <service_name> # print information about service
rosservice uri <service_name> # print service ROSRPC
```

rosbag

ROS offers the possibility to record the data published on topics into bag files:

1. create a directory to store the bag files:

```
~/ mkdir ros_bag_files && od ros_bag_files
```

2. run the record command:

```
rosbag record -O <bag_name>.bag <topic_name> <topic_name>
```

3. play the bag file:

```
rosbag play <bag_name>.bag
```

Many options are available for the *rosbag* command, see this page for more details.

Note: to play a bag with point clouds, it is required to have the following topics:

- /cloud
- /tf_static

The TF transformation is required, otherwise RViz can't display the point clouds.

```
rosbag record -O cloud.bag /cloud /tf_static
...
rosbag play cloud.bag
```

rosmsg

Display information about ros messages.

Other Commands

```
roscd <PKG_NAME>
                                    # move to the folder of the package
rosinstall <PKG_NAME>
                                    # install a ROS package
rosdep <PKG_NAME</pre>
                                    # install all the dependencies of a package
                                    # tool with many plug-ins available such as topic_
→publisher, service caller, ...
                                    # display the connections between nodes
rqt_graph
rviz
                                    # launch the graphical tool to visualize robots, _
\hookrightarrowpoint clouds, ...
                                    # create a PDFcalled ``frames.pdf`` with the TF_
view_frames
\hookrightarrowframes that are active
evince frames.pdf
                                    # show with evince the generated frames.pdf
```

Catkin

More info:

• http://wiki.ros.org/catkin/Tutorials

Create Package

- 1. new terminal
- 2. navigate to the source folder of the catkin workspace: .../catkin_ws/src
- 3. run:catkin_create_pkg <PACKAGE_NAME> <DEPENDENCIES>
- 4. update both CMakeLists.txt and package.xml note: run_depend has to be replaced by the exec_depend
- 5. write source code in the source folder of the package:
- 6. build the catkin workspace with the alias command: cm
- 7. launch the master as explained [here](ros-commands.md#roscore).
- 8. now launch the node as explained [here](#roslaunch) and [here](rosrun).

Build

```
cm
catkin_make  # build the whole workspace
catkin_make <PKG_NAME>  # build a single package
```

Install

```
catkin_make install  # installs all executables  # installs single executables
```

Python modules

Tips:

- put the script in a folder called scripts
- make sure to run chmod +x <script_name>.py so that the script is recognized as an executable by ROS

10.5.3 Update services with RQT

- 1. launch RQT from a new terminal: run rqt
- 2. Search for the plugin Service Caller
- 3. Choose the service that you want to update
- 4. Fill the expression field with an expected parameter of this service
- 5. Call the service and the response is displayed

10.6 Installation

- How to install ROS
 - Prerequisites
 - * NTP
 - * Sources
 - * Keys
 - ROS Base
 - ROS Additional Packages
 - * *RQT*
 - * Individual ROS packages
 - Setup ROS Environment
 - * Initialise rosdep
 - * Environment setup
 - * ROS Install
 - * Create catkin workspace
 - Shell Scripts
 - Additional Install
 - * Hitachi SDK
- Configuration
 - ROS Configuration
 - * .bashrc
 - * .zshrc
 - ROS Test

10.6.1 How to install ROS

This installation is based on Ubuntu 18.4 LTS and ROS Melodic Morenia.

Prerequisites

Some tools are not mandatory.

NTP

Only needed in a multi-pc system.

```
echo "Install Chrony and ntpdate"
sudo apt-get install -y chrony ntpdate
sudo ntpdate -q ntp.ubuntu.com
```

Sources

ROS Ubunbtu apt-get packages sources.

Ubuntu 18.04 LTS (Bionic Beaver)

Keys

- ROS Kinetic
- ROS Melodic

```
echo "Add ROS Package Key"
sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80' --recv-key

→C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654
```

ROS Base

```
echo "Install ROS Base Desktop Full"
sudo apt-get install ros-melodic-desktop-full
```

ROS Additional Packages

RQT

```
echo "Install ROS R-QT"
sudo apt-get install ros-melodic-rqt*
```

Individual ROS packages

Search & install individual ROS packages

```
echo "Install ROS R-QT"
apt-cache search ros-melodic
sudo apt-get install ros-melodic-[NAME_OF_PACKAGE]
```

Setup ROS Environment

Initialise rosdep

```
echo "[Initialize rosdep]"
sudo sh -c "rosdep init"
rosdep update
```

Environment setup

Differs depending if it's zsh or bash

```
echo "[Environment setup and getting rosinstall]"
if [ -n "$ZSH_VERSION" ]; then
    # assume Zsh
    source /opt/ros/$name_ros_version/setup.zsh
elif [ -n "$BASH_VERSION" ]; then
    # assume Bash
    source /opt/ros/$name_ros_version/setup.sh
fi
```

ROS Install

```
sudo apt install -y python-rosinstall python-rosinstall-generator python-wstool
```

Create catkin workspace

```
mkdir -p $HOME/$name_catkin_workspace/src
cd $HOME/$name_catkin_workspace/src
catkin_init_workspace
cd $HOME/$name_catkin_workspace
catkin_make
```

Shell Scripts

All the above can be done with help of the ros-melodic-install.bash

Additional Install

Hitachi SDK

```
cd ~/Downloads
eche "SINDENT Manually download http://hlds.co.jp/download/tofsdk/v2.3.0/

→HldsTofSdk.2.3.0ubuntu16_x64.zip into your Downloads/ folder"
eche ""
eche ""
eche "PRESS [ENTER] WHEN YOU'RE FINISHED AND TO CONTINUE THE INSTALLATION"
read
mkdir HldsTofSdk.2.3.0ubuntu16_x64
unzip HldsTofSdk.2.3.0ubuntu16_x64.zip -d ./HldsTofSdk.2.3.0ubuntu16_x64
sudo apt install HldsTofSdk.2.3.0ubuntu16_x64/libtof-dev_2.3.0-4ubuntu16_amd64.deb
```

10.6.2 Configuration

ROS Configuration

.bashrc

```
echo "[Set the ROS evironment in ~/.bashrc]"
echo "alias eb='vim ~/.bashrc'" >> ~/.bashrc
echo "alias sb='source ~/.bashrc'" >> ~/.bashrc
echo "alias gs='git status'" >> ~/.bashrc
echo "alias gp='git pull'" >> ~/.bashrc
echo "alias cw='cd ~/$name_catkin_workspace'" >> ~/.bashrc
echo "alias cs='cd ~/$name_catkin_workspace/src'" >> ~/.bashrc
echo "alias cm='cd ~/$name_catkin_workspace && catkin_make'" >> ~/.bashrc
echo "source /opt/ros/$name_ros_version/setup.bash" >> ~/.bashrc
echo "source ~/$name_catkin_workspace/devel/setup.bash" >> ~/.bashrc
echo "export ROS_MASTER_URI=http://localhost:11311" >> ~/.bashrc
echo "export ROS_HOSTNAME=localhost" >> ~/.bashrc
```

.zshrc

```
echo "[Set the ROS evironment in ~/.zshrc]"
echo "alias eb='vim ~/.zshrc'" >> ~/.zshrc
echo "alias sb='source ~/.zshrc'" >> ~/.zshrc
echo "alias gs='git status'" >> ~/.zshrc
echo "alias gp='git pull'" >> ~/.zshrc
echo "alias cw='cd ~/$name_catkin_workspace'" >> ~/.zshrc
echo "alias cs='cd ~/$name_catkin_workspace/src'" >> ~/.zshrc
echo "alias cm='cd ~/$name_catkin_workspace && catkin_make'" >> ~/.zshrc
echo "source /opt/ros/$name_ros_version/setup.zsh" >> ~/.zshrc
echo "source ~/$name_catkin_workspace/devel/setup.zsh" >> ~/.zshrc
echo "export ROS_MASTER_URI=http://localhost:11311" >> ~/.zshrc
echo "export ROS_HOSTNAME=localhost" >> ~/.zshrc
```

ROS Test

roscore

10.7 Launch

- Launcher
 - Launch file
 - * Arguments
 - * Including other launch files
 - Create a launcher in a new package
 - Include another launcher inside this launcher
 - Parameters in launcher
 - * Get the value of a parameter at run time
 - * Public vs Private parameters
- Rviz configuration

10.7.1 Launcher

- launch os a tool for launchine multiple nodes (as well as setting parameters)
- Are written in XM as *.launch files
- If not yet running, launch atuomatically stars a roscore

Browse to the folder and start a launch file with

```
roslaunch <file_name>.launch
```

Start a launch file from a package with

```
roslaunch <package_name> <file_name>.launch
```

Launch file

Listing 1: talker_listerner.launch

```
<launch>
<node name="listener" pkg="roscpp_tutorials" type="listener" output="screen"/>
<node name="talker" pkg="roscpp_tutorials" type="talker" output="screen"/>
</launch>
```

launch: Root element of the launch file

- node: Each <node> tag specifies a node to be launched
- name: Name of the node (free to choose)
- pkg: Package containing the node
- type: Type of the node, there must be a corresponding executable with the same name

10.7. Launch 69

• output: Specifies where to output log messages (screen: console, log: log file)

More Info

- http://wiki.ros.org/roslaunch/XML
- http://wiki.ros.org/roslaunch/Tutorials/Roslaunch%20tips%20for%20larger%20projects

Arguments

• Create re-usable launch files with <arg> tag, which works like a parameter (default optional)

```
<arg name="arg_name" default="default_value"/>
```

· Use arguments in launch file with

```
$(arg_arg_name)
```

• When launching, arguments can be set with

```
roslaunch launchf_file.launch arg_name:value
```

Example:

Listing 2: range_world.launch

```
<?xml version="1.0"?>
 <launch>
   <arg name="use_sim_time" default="true"/>
   <arg name="world" default="gazebo_ros_range"/>
   <arg name="debug" default="false"/>
   <arg name="physics" default="ode"/>
   <group if="$(arg use_sim_time)">
     <param name="/use_sim_time" value="true" />
    </group>
   <include file="$(find gazebo_ros) /launch/empty_world.launch">
   <arg name="world_name" value="$(find gazebo_plugins) / test/test_worlds/$(arg...)</pre>
→world).world"/>
    <arg name="debug" value="$(arg debug)"/>
    <arg name="physics" value="$(arg physics)"/>
  </include>
</launch>
```

More info http://wiki.ros.org/roslaunch/XML/arg

Including other launch files

• Include other launch files with <include> tag to organize large projects

```
<include file="package_name" />
```

• Find the system path to other packages with

```
$(find package_name)
```

· Pass arguments to the included file

```
<arg name="arg_name" value="value"/>
```

10.7. Launch 70

Listing 3: range_world.launch

```
<?xml version="1.0"?>
<launch>
 <arg name="use_sim_time" default="true"/>
 <arg name="world" default="gazebo_ros_range"/>
 <arg name="debug" default="false"/>
 <arg name="physics" default="ode"/>
   <group if="$(arg use_sim_time)">
   <param name="/use_sim_time" value="true" />
 </group>
   <include file="$(find gazebo_ros) /launch/empty_world.launch">
   <arg name="world_name" value="$(find gazebo_plugins)/test/test_worlds/</pre>

$ (arg world).world"/>

   <arg name="debug" value="$(arg debug)"/>
   <arg name="physics" value="$(arg physics)"/>
 </include>
</launch>
```

More info: http://wiki.ros.org/roslaunch/XML/include

Create a launcher in a new package

```
1. move to the folder of the package
```

```
2. run: mkdir launch && cd launch
```

3. run:gedit <LAUNCHER_NAME>.launch

4. fill the launcher file, for example:

Include another launcher inside this launcher

Add the include directive:

```
<launch>
    <include file="$(find <PKG_NAME>)/launch/<LAUNCHER_NAME>.launch" />
    </launch>
```

This is very useful to combine launcher together, or complete a first launcher:

- the first launcher is responsible to launch a driver
- the second launcher that includes the first one launches also a graphical tool on top of that

The advantage being that it is not necessary to copy paste all the code of the first launcher into the second one to use them together.

10.7. Launch 71

Parameters in launcher

Parameters can be set in the launcher and get by the node at run time. This is a convenient way to avoid rebuilding the code each time it is necessary to change the value of a variable, for example a path to a file.

The syntax is the following one:

```
<param name="<PARAM_NAME>" type="<TYPE>" value="<VALUE>" />
```

Get the value of a parameter at run time

It can be used in the node at run time with this C++ code:

```
ros::NodeHandle nh;
std::string iniPath;
nh.getParam("ini_path", iniPath);
```

The node handler gets the parameter called *ini_path* in the launcher and will store it in the variable *iniPath*. If the parameter is public, therefore accessible by all the nodes, this is sufficient to get its value. If the parameter is private to a node, then the node handler needs to know the name of the node:

```
ros::NodeHandle nh;
std::string iniName;
nh.getParam("tof_driver_1/ini_name", iniName);
```

To get the name of the node at run time, it is possible to use this line:

```
std::string nodeName = ros::this_node::getName();
```

Public vs Private parameters

Depending of where the parameter is declared in the launcher, the parameter will be either private to a node, or accessible by all the nodes. If the parameter is declared outside of a <node></node> tag, it is public and accessible to all the nodes. At the opposite, if the parameter is declared inside a <node></node> tag, it will only be accessible by the node, with the specific method described above.

In this example:

- The parameter *ini_path* is public and accessible by all the nodes only with its name.
- The parameter *ini_name* is private to each node and is accessible with the name of the node and its name, concatenated together. This allows to declare two time the same parameter with different value, as long as they are declared inside different nodes.

(continues on next page)

10.7. Launch 72

(continued from previous page)

10.7.2 Rviz configuration

After setting up the display configuration in Rviz, you can save it with the tab File -> Save config as -> ...

Then you can call it directly in the launch file by adding:

```
<node pkg="rviz" type="rviz" name="rviz"
args="-d <PATH_TO_FILE>/<CONFIG_NAME>.rviz"/>
```

This will open Rviz with the saved configuration when the *launch* file is launched.

10.8 Lidar Driver

• Install the SDK

10.8.1 Install the SDK

run in a new terminal:

```
sudo dpkg -i libtof-dev_<version_number>ubuntu16_amd64.deb
```

10.9 Packages

- Package Structure
- Package Files
 - file package.xml
 - file CMakeLists.txt
- Eclipse integration
- C++ Client Library
 - Example
 - Node Handle
 - Logging ROS_INFO
 - * Severity Levels

10.8. Lidar Driver 73

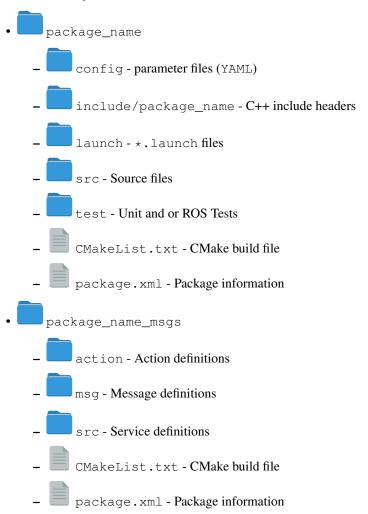
- Subscriber
- Publisher
- OOP
- Parameter Server
 - * C++API

10.9.1 Package Structure

ROS software is organized into packages, which can contain source code, launch files, configuration files, message definitions, data, and documentation. A package can depend on other packages called *dependencies*.

```
catkin_create_pkg <package_name> {dependencies}
```

A package need two things, its source code and the message definition. It is encouraging to place message definition into a separate folder.



More info

• http://wiki.ros.org/Packages

10.9.2 Package Files



package.xml

- The package . xml file defines the properties of the package
 - Package name
 - Version number
 - Authors
 - Dependencies on other packages
 - **–** ...

Listing 4: package.xml

More info

• http://wiki.ros.org/catkin/package.xml



CMakeLists.txt

The CMakeLists.txt is the input to the CMake build system

- 1. Required CMake Version (cmake_minimum_required)
- 2. Package Name (project())
- 3. Find other CMake/Catkin packages needed for build (find_package())
- 4. Message/Service/Action Generators (add_message_files(), add_service_files(),
 add_action_files())
- 5. Invoke message/service/action generation (generate_messages())
- Specify package build info export (catkin_package())
- 7. Libraries/Executables to build (add_library()/add_executable()/ target_link_libraries())
- 8. Tests to build (catkin_add_gtest())
- 9. Install rules (install())

Listing 5: CMakeLists.txt

More info

• http://wiki.ros.org/catkin/CMakeLists.txt

10.9.3 Eclipse integration

• Build the Eclipse project files with additional build flags

```
catkin build package_name --cmake-args -G"Eclipse CDT4 - Unix Makefiles" -D__ 

--cplusplus=201103L D__GXX_EXPERIMENTAL_CXX0X__=1
```

- To use flags by default in your catkin environment, use the *catkin config* command.
- The Eclipse project files will be generated in ~/catkin_ws/build

10.9.4 C++ Client Library

- http://wiki.ros.org/roscpp
- http://wiki.ros.org/roscpp/Overview

Example

(continues on next page)

(continued from previous page)

```
ROS_INFO_STREAM("Hello World " << count); // ROS_info() logs messages from_

ofs

ros::spinOnce(); // processes incommind msg via_

ocallbacks

loopRate.sleep();

count++;

return 0;
```

Node Handle

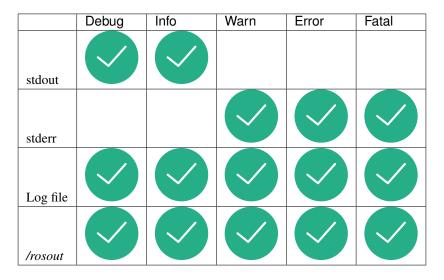
http://wiki.ros.org/roscpp/Overview/NodeHandles

Logging ROS_INFO

- http://wiki.ros.org/rosconsole
- http://wiki.ros.org/roscpp/Overview/Logging

Send text to log files and console. Instead of std::cout, use e.g. ROS_INFO.

Severity Levels



Formatting Style

```
ROS_INFO("Result: %d", result); // printf style
ROS_INFO_STREAM("Result: " << result); // stream style
```

Launchfile

To see the output in the console set configuration to screen in the launch file.

```
<launch>
    <node name="listener" more="stuff" output="screen"/>
</launch>
```

Subscriber

http://wiki.ros.org/roscpp/Overview/Publishers%20and%20Subscribers

Start listening to a topic by calling the method subscribe() of the node handle

```
ros::Subscriber subscriber = nodeHandle.subscribe(topic, queue_size, callback_

→function);
```

Example

Listing 6: listener.cpp

```
#include "ros/ros.h"
#include "std_msgs/String.h"

// callback function when a message is received
void chatterCallback(const std_msgs::String& msg) {
    ROS_INFO("I heard: [%s]", msg.data.c_str());
}

int main(int argc, char **argv) {
    ros::init(argc, argv, "listener");
    ros::NodeHandle nodeHandle;
    // Subscript to topic with a queue size of 10 (1-10 is recommended)
    ros::Subscriber subscriber = nodeHandle.subscribe("chatter",10,
    chatterCallback);
    ros::spin(); // stay's here forever
    return 0;
}
```

Publisher

http://wiki.ros.org/roscpp/Overview/Publishers%20and%20Subscribers

Create a publisher with help of the node handle

```
ros::Publisher publisher = nodeHandle.advertise<message_type>(topic, queue_size);
```

Example

```
:caption: talker.cpp
#include <ros/ros.h>
#include <std_msgs/String.h>

int main(int argc, char **argv) {
    ros::init(argc, argv, "talker");
    ros::NodeHandle nh;
    // Node handle queue size of 1
```

(continues on next page)

(continued from previous page)

```
ros::Publisher chatterPublisher = nh.advertise<std_msgs::String>("chatter", 1);
ros::Rate loopRate(10);

unsigned int count = 0;
while (ros::ok()) {
    std_msgs::String message;
    // Cretae message content
    message.data = "hello world " + std::to_string(count);
    ROS_INFO_STREAM(message.data);
    chatterPublisher.publish(message);
    ros::spinOnce();
    loopRate.sleep();
    count++;
}
return 0;
}
```

OOP

http://wiki.ros.org/roscpp_tutorials/Tutorials/UsingClassMethodsAsCallbacks

Example

```
:caption: my_package_node.cpp
#include <ros/ros.h>
#include "my_package/MyPackage.hpp"
int main(int argc, char** argv) (
    ros::init(argc, argv, "my_package");
    ros::NodeHandle nodeHandle("~");
    // Call
    my_package::MyPackage myPackage(nodeHandle);

    ros::spin();
    return 0;
}
```

| class MyPackage | class Algorithm |
|-------------------------------------|----------------------------------------------------------------|
| Main node class providing ROS | Class implementing the algorithmic part of the node |
| interface (subscribers, parameters, | Note: The algorithmic part of the code could be separated in a |
| timers etc.) | (ROS-independent) library |

Parameter Server

http://wiki.ros.org/roscpp/Overview/Parameter%20Server

Example Parameter File

```
:caption: config.yaml

camera:
    left:
        name: left_camera
        exposure: 1
    right:
        name: right_camera
        exposure: 1.1
```

Example Launch file

C++ API

```
ros::NodeHandle nodeHandle("~");
std::string topic;
if (!nodeHandle.getParam("topic", topic)) {
    ROS_ERROR("Could not find topic parameter!");
}
```

Get a parameter in C++ with

```
nodeHandle.getParam(parameter_name, variable)
```

- Method returns true if parameter was found, false otherwise
- Global and relative parameter access:
 - Global parameter name with preceding /

```
nodeHandle.getParam("/package/camera/left/exposure", variable)
```

Relative parameter name (relative to the node handle)

```
nodeHandle.getParam("camera/left/exposure", variable)
```

• For parameters, typically use the private node handle

```
ros::NodeHandle("~")
```

10.10 External Packages and Nodes

```
Terminology
Overview
3D Mapping

SLAM
* Octomap_server: +
* Hector slam: +
* REMODE: ~
LOAM
* RTABMAP: +
* Spin Hokuyo: +
* Lego-LOAM: ~
- Velodyne loam: ~
```

- Bad solution -
- Modbus
- · Object Tracking
 - Multiple objects lidar tracking : ~
- Object Detection
- QR code readers

10.10.1 Terminology

- +: interesting topics and hardware abstraction
- ~: interesting, but quite a lot of work to do for hardware compatibility or mapping
- -: bad solution

10.10.2 Overview

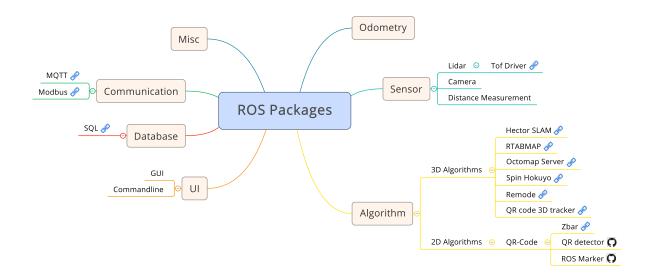


Fig. 4: ROS Packages Overview

10.10.3 3D Mapping

SLAM

Octomap server: +

3D occupancy grid mapping, independent from sensor, looks like it does not need odometry

- https://youtu.be/yp0f8-AKvDU
- https://wiki.ros.org/octomap_mapping
- https://wiki.ros.org/octomap_server
- http://octomap.github.io/

Plus:

- · maintained
- compatible with melodic
- · documentation available as well as many
- · no odometry
- independent from hardware (only require the right input topics)

Minus:

• ...

Inputs required:

• sensor_msgs/PointCloud2

Hector slam: +

- https://github.com/tu-darmstadt-ros-pkg/hector_slam
- http://wiki.ros.org/hector_slam

Not sure whether we're interested in hector slam itself, or on the

Plus:

- · maintained
- not directly compatible with melodic, but easy to build it from source for melodic
- · odometry not needed

Minus:

- mostly created for 2D mapping and robot navigation
- not much documentation

Inputs required:

• ...

REMODE: ~

https://www.ros.org/news/2016/02/open-source-release-remode-probabilistic-monocular-dense-reconstruction-in-real-time.

modeling of many 3D objects, like rooms, persons, ...

Plus:

- · noise reduction
- nice rendering

Minus:

- not much documentation and precisions about hardware/drivers/topics
- maybe "too much" for our needs?
- looks like it is not maintained anymore: latest commit was 4 years ago

Inputs:

• ...

LOAM

RTABMAP:+

• http://wiki.ros.org/rtabmap_ros

Plus:

- · maintained
- compatible with melodic
- real time mapping
- publishes:
 - 3D point clouds
 - 2D occupancy maps
- · tutorials and documentation available

Minus:

• oriented towards robot navigation, although "top-down" modeling seems to be possible

Inputs required:

- odometry (not mandatory in all cases)
- scan 2D or 3D

Spin Hokuyo: +

- https://github.com/RobustFieldAutonomyLab/spin_hokuyo
- http://wiki.ros.org/spin_hokuyo

It creates a point cloud with a 2D LiDaR and a servomotor. The interesting node compiles small point clouds to make one big point cloud. Could be very useful to make our digital model.

Plus:

- has a node that compiles point clouds and publish them on a topic
- · great rendering

Minus:

- · designed for another sensor, but the node that compiles point clouds does not care about that
- · need some odometry work

Inputs required:

- · laser scan
- odometry

Lego-LOAM: ~

• https://github.com/RobustFieldAutonomyLab/LeGO-LOAM

Plus:

· good rendering

Minus:

- designed for robot navigation, not for "top-down mapping"
- designed for another sensor (velodyne)

Inputs:

• ...

Velodyne loam: ~

• http://wiki.ros.org/loam_velodyne

Plus:

- good rendering
- builds 3D maps

Minus:

- · for velodyne sensor
- · robot navigation

Inputs:

• ...

Bad solution -

- https://github.com/koide3/hdl_graph_slam : not what we need. creates maps with corridors and doors, but not "top-down" mapping
- http://wiki.ros.org/robot_pose_ekf : not what we need
- http://wiki.ros.org/ethzasl_icp_mapper: doc not up to date, slowly not maintained anymore, ...
- $\bullet \ https://github.com/ethz-asl/libpointmatcher/blob/master/doc/index.md\\$

10.10.4 Modbus

• http://wiki.ros.org/modbus

10.10.5 Object Tracking

Multiple objects lidar tracking: ~

• https://github.com/praveen-palanisamy/multiple-object-tracking-lidar

Plus:

- tracks objects in real time
- · hardware independent

Minus:

• 2D maps, most likely used for robot navigation

Inputs:

• ...

10.10.6 Object Detection

- https://www.acin.tuwien.ac.at/vision-for-robotics/software-tools/v4r-library/
- https://rgit.acin.tuwien.ac.at/v4r/v4r_ros_wrappers
- http://wiki.ros.org/object_recognition
- https://www.osrfoundation.org/ros2-object-detection-demo/
- http://wiki.ros.org/find_object_2d

10.10.7 QR code readers

- http://wiki.ros.org/zbar_ros
- https://github.com/mdrwiega/qr_detector
- http://wiki.ros.org/visp_auto_tracker

10.11 RViz

- Overview
- Run
- Built-In Display Types

10.11.1 Overview

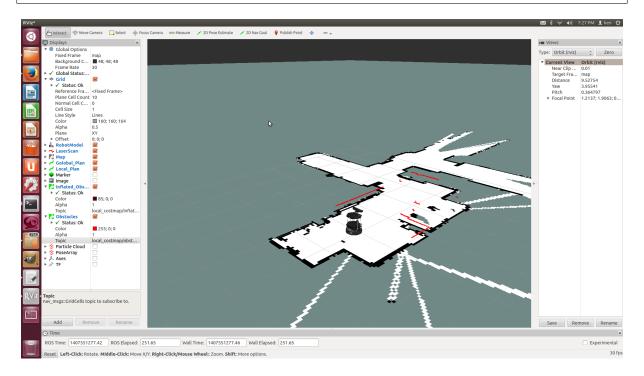
http://wiki.ros.org/rviz

- 3D visualization tool for ROS
- Subscribes to topics and visualizes the message contents
- Different camera views (orthographic, top-down, etc.)
- Interactive tools to publish user information
- Save and load setup as RViz configuration
- Extensible with plugins

10.11. RViz 85

10.11.2 Run

rosrun rviz rviz



Save configuration with $\mathtt{ctrl+s}$

10.11. RViz 86

10.11.3 Built-In Display Types

| Name | Description | Messages Used |
|----------|---------------------------------------------------------------------|-----------------------------|
| Axes | Displays a set of Axes | |
| Effort | Shows the effort being put into each revolute joint of a robot. | sensor_msgs/JointStates |
| Camera | Creates a new rendering window from the perspective of a camera, | sensor_msgs/Image, sen- |
| | and overlays the image on top of it. | sor_msgs/CameraInfo |
| Grid | Displays a 2D or 3D grid along a plane | |
| Grid | Draws cells from a grid, usually obstacles from a costmap from the | nav_msgs/GridCells |
| Cells | navigation stack. | |
| Image | Creates a new rendering window with an Image. Unlike the Cam- | sensor_msgs/Image |
| | era display, this display does not use a CameraInfo. Version: Dia- | |
| | mondback+ | |
| Inter- | Displays 3D objects from one or multiple Interactive Marker | visualiza- |
| active- | servers and allows mouse interaction with them. Version: Elec- | tion_msgs/InteractiveMarker |
| Marker | tric+ | |
| Laser | Shows data from a laser scan, with different options for rendering | sensor_msgs/LaserScan |
| Scan | modes, accumulation, etc. | |
| Map | Displays a map on the ground plane. | nav_msgs/OccupancyGrid |
| Markers | Allows programmers to display arbitrary primitive shapes through | visualization_msgs/Marker, |
| | a topic | visualiza- |
| | | tion_msgs/MarkerArray |
| Path | Shows a path from the navigation stack. | nav_msgs/Path |
| Point | Draws a point as a small sphere. | geome- |
| | | try_msgs/PointStamped |
| Pose | Draws a pose as either an arrow or axes. | geome- |
| | | try_msgs/PoseStamped |
| Pose | Draws a "cloud" of arrows, one for each pose in a pose array | geometry_msgs/PoseArray |
| Array | | |
| Point | Shows data from a point cloud, with different options for rendering | sensor_msgs/PointCloud, |
| Cloud(2) | modes, accumulation, etc. | sensor_msgs/PointCloud2 |
| Polygon | Draws the outline of a polygon as lines. | geometry_msgs/Polygon |
| Odome- | Accumulates odometry poses from over time. | nav_msgs/Odometry |
| try | | |
| Range | Displays cones representing range measurements from sonar or IR | sensor_msgs/Range |
| D 1 : | range sensors. Version: Electric+ | |
| Robot- | Shows a visual representation of a robot in the correct pose (as | |
| Model | defined by the current TF transforms). | |
| TF | Displays the ros wiki tf transform hierarchy. | |
| Wrench | Draws a wrench as arrow (force) and arrow + circle (torque) | geome- |
| 0 1 | | try_msgs/WrenchStamped |
| Oculus | Renders the RViz scene to an Oculus headset | |

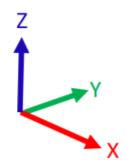
10.12 Transform Frames

A frame in the ROS language is a specific coordinate system in the space. ROS abstracts elements of a robot as coordinates frames. Each physical part of a robot that has a particular meaning will most likely have its own frame .

a sensor : laser_frame an arm : left_arm_frame

It is up to the programmer to create frames where it is necessary, but some frames are already defined by ROS (see below).

Each frame has it own origin and coordinate system:



Memory trick: RGB -> XYZ

Fig. 5: coordinate frame axis

To keep trace of the frames in the whole coordinate system, they must all refer to a main frame. Knowing the position of the main frame and the relative positions of all the other frames, ROS is able to know the exact position of each frame all continuously.

The TF2 package tracks the coordinate frames. There are several predefined frames:

- world: kind of the parent of all the frames, does not move, there is only one single world
- *map*: child of *world*, can be freely fixed in the world frame, does not move compared to the *world*, but it can be several *map* frames in a *world* (usually one *map* per robot)
- *odom*: child of *map*, fixed at the start point of the robot in the *map* frame, does not move compared to *world* and *map*
- base_link: kind of the reference frame of a robot, it is moving in odom, therefore moving in map and world
- •

The TF tree shows the relations between the frames:

One can create coordinate frames for each part of the robot that needs to be tracked, for example:

- scanner_frame: position of the scanner on a robot, somehow linked to the base_link
- wheels_frame: position of the wheels on a robot, somehow linked to the base_link

The links between the *base_link* and the other frames can be direct, or they can be relative to it via other frames.

They are linked together by TF (transform frames). TF can be either static, which means that the relation between two frames will never change (for example two sensors being fixed 1 meter away), or dynamic when the relation evolves in the time (for example the arm of a robot compared to its head).

Let us use the example our two LIDAR sensor: they are oriented in the same way, they are on the same table, the only difference being there is 2.15 meter between them. For this example, they will never move nor rotate. We can use the node $static_transform_publisher$ to inform other nodes that will use their data of their relative position. We will also fix them in the world, map and $base_link$ frame.

Since the *base_link* frame will not move neither, it will also be fixed to the *map* by a static transform. The static transformations are called as a node from a launcher:

(continues on next page)

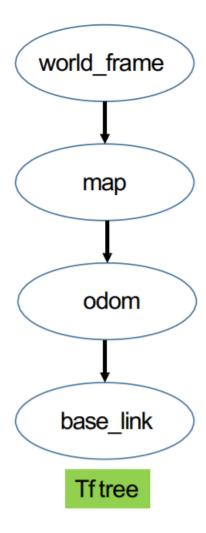


Fig. 6: tf tree

(continued from previous page)

Which will produce the following TF tree:

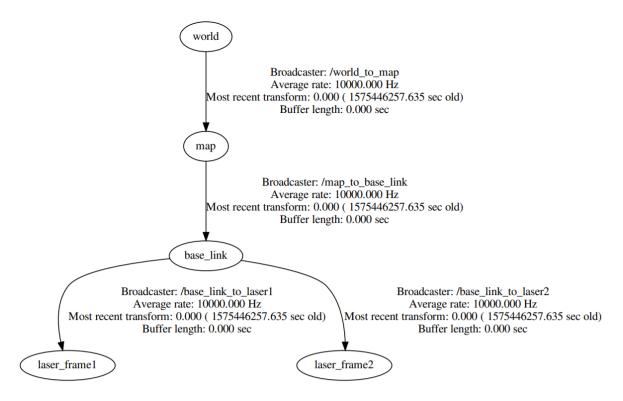


Fig. 7: lidar tf tree

The arguments are:

- translations in X, Y, Z
- rotations around X, Y, Z
- parent frame_id
- child frame_id

Each topic has a reference frame. This means that each message being published on a topic kind of contains the position "from where it comes". This is the *frame_id* parameter. The node that will published the data of the LIDAR shall publish them with the right *frame_id*, otherwise the TF tree will not be able to link all the TF together.

Documentation about frames and transformations can be found there:

• tf2

11

LaTeX



11.1 Introduction

- Some LaTeX helppages
- Generate PDF files

11.1.1 Some LaTeX helppages

- HEI SPL Latex Templates
- Cheatsheet A Guide to Latex
- Tex Stackexchange Forum

11.1.2 Generate PDF files

Latex is best suited to insert images as pdf. In order to convert images or svg into pdf use inkscape Convert \star .svg images with inkscape to \star .pdf and \star .pdf_tex

 $\verb"inkscape -D -z --file="image.svg --export-pdf="image.pdf --export-latex" \\$

11.2 Installation LaTeX

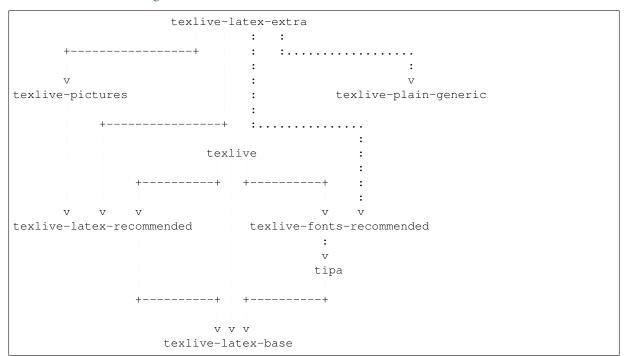
- Base Install
 - Linux
 - Windows
- Manual Package install
 - Manual Package Linux
 - Manual Package Windows

11.2.1 Base Install

Linux

| Package | Archives | Disk Space |
|---------------------------|----------|------------|
| texlive-latex-base | 59 MB | 216 MB |
| texlive-latex-recommended | 74 MB | 248 MB |
| texlive-pictures | 83 MB | 277 MB |
| texlive-fonts-recommended | 83 MB | 281 MB |
| texlive | 98 MB | 314 MB |
| texlive-plain-generic | 82 MB | 261 MB |
| texlive-latex-extra | 144 MB | 452 MB |
| texlive-full | 2804 MB | 5358 MB |

see also Tex Stack Exchange



sudo apt-get install texlive-latex-extra

Windows

- Install MikTeX https://miktex.org/download
- · MikTeX Packages
 - minted

```
pip install pygments
```

add Python Scripts to PATH Environment Variable. %USERPROFILE%\AppData\Local\
Continuum\anaconda3\Scripts\

- · Install TeXstudio
 - https://texstudio.org
 - Options => Configure TeXstudio => Commands => add Interpreter Flag -shell-escape
 - enable line numbers
 - enable white spaces
- Install Inkscape
 - https://inkscape.org/release/

11.2.2 Manual Package install

For manual installing *.sty Packages and *.cls Class files.

Warning: For every package create a separate folder

Manual Package Linux

• Find TEXMFHOME directory

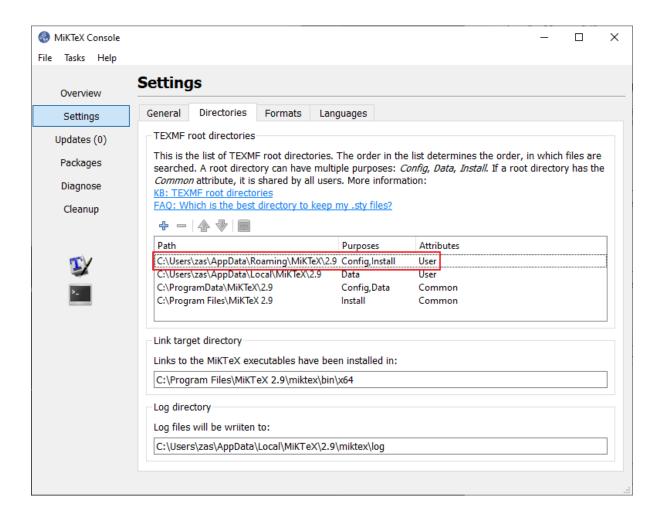
```
kpsewhich -var-value TEXMFHOME
```

- Navigate to \$(TEXMFHOME)/tex/latex
- Copy paste your *.sty and *.cls
- Update Package index

texhash

Manual Package Windows

- Open MikTeX Console and go to Settings -> Directories
- The Config, Install and User folder is the location of your Packages: %USERPROFILES%/ AppData/Roaming/MikTeX/2.9/
- Inside you have to navigate to tex/latex/ folder
- %USERPROFILES%/AppData/Roaming/MikTeX/2.9/tex/latex/
- Copy paste your *.sty and *.cls
- Update Package index



texhash

ReStructuredText

12.1 Introduction

• Some RST Syntax helppages

12.1.1 Some RST Syntax helppages

- rst-cheatsheet.pdf
- Thomas Cokelaer RST Sphinx Syntax
- Docutil Quickref
- Raslina RST Cheatsheet

12.2 RST and Sphinx Cheatsheet

In this page you will get a quick overview about the most used syntax.

- Table of content
- Titles
- Markup
- Links
 - External Links
 - * Internet
 - * Other Repo's
 - * Other Sphinx Pages
 - Internal Links
 - * Link to Titles
 - * Internal References

- * File Links
- Images
 - Image Placement
 - Inline Images
- Lists
- Tables
- Code
- Infoboxes
- Special Formatting
- Math
- Exclude
- GraphViz
- Wavedrom
 - Timing Diagrams
 - Register
- PlantUML

12.2.1 Table of content

To include a table of content of all title in a page use

```
.. contents:::local:
```

12.2.2 Titles

The lines have to be as long or longer than the text.

12.2.3 Markup

| *emphasis* | emphasis |
|---------------------|------------------------------------------------------------------------------------|
| **strong emphasis** | strong emphasis |
| `interpreted text` | The rendering and meaning of interpreted text is domain- or application-dependent. |
| ``inline literal`` | inline literal |
| :markup: | markup |
| > quote markup | > quote markup |

12.2.4 Links

External Links

Internet

```
`python <http://www.python.org/>`_
`<http://www.python.org/>`_
http://www.python.org/
```

python

http://www.python.org/

http://www.python.org/

Other Repo's

The plugin 'sphinx.ext.extlinks allows creating shortcuts

```
:config_repo: jupyter config <tree/master/config/jupyter>
:zawiki_repo: zawiki link <>>
```

jupyter config zawiki link

Other Sphinx Pages

- absolute link from root About
- · relative link from document location About

```
* absolute link from root
   :doc: /about/index

* relative link from document location
   :doc: ../../about/index
```

In order to link to another subheader in another document you need to use *Internal References*.

In the page to be jumped to add .. _ref_name:, and then you can:

```
:ref:`ref_name`
:ref:`link title<ref_name>`
```

Like so:

- How to use Sphinx Documentation
- Sphinx Doc Link

Internal Links

Link to Titles

Link to titles directly is done with the extension sphinx.ext. autosectionlabel.

Important: You need to add the folder_name and subfolder(s)`_name name as well as file_name without .rst extension in order to reference a section title. This avoids the duplicated label warning.

Back to top

Images

Internal References

In any place of the document a reference point can be inserted and later refered to.

```
see :ref: ref-point'
```

see Internal References

File Links

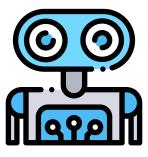
To link to a file within the Sphinx file structure use the Role : download:

 $../../{\tt coding/ros/books/Mastering_ROS_for_Robotics_Programming.}$ pdf

Mastering_ROS_for_Robotics_Programming

12.2.5 **Images**

.. figure:: /img/logo.*



Important: Images should be either in png or svg format

Important: For \star . svg files the file ending needs to be changed from svg to \star . That way for html svg is used and pdf or pn for the latex or pdf output.

Image Placement

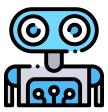
```
figure: /img/logo.*
   :align: left
   :width: 100px

   figure: /img/logo.*
   :align: center
   :width: 100px

   figure: /img/logo.*
   :align: right
   :width: 100px

   figure: /img/logo.*
   :align: center
   :width: 100px

   figure: /img/logo.*
   :align: center
   :width: 100px
   :height: 100px
   :scale: 50 %
   :alt: this is the knowhow logo
Caption of figure
```



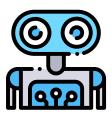




Fig. 1: Caption of figure

Inline Images

For inline images to work, a substitution needs to be made



```
After that the image | folder | can be integrated inline.
```

الكلحيان

After that the image can be integrated inline.

12.2.6 Lists

- item 1
- item 1.1
- item 1.2
- item 2
- item 2.1
- * item 2.1.1
- 1. auto enumerated list item 1
- 2. auto enumerated list item 1
- 3. auto enumerated list item 1
- 4. auto enumerated list item 1
- 3. enumerated list start with item 3
- 4. auto enumerated list item 4
- 5. auto enumerated list item 5
- 6. auto enumerated list item 6

12.2.7 **Tables**

| Header 1 | Header 2 | Header 3 |
|------------|-------------------------|-----------------------------|
| body row 1 | column 2 | column 3 |
| body row 2 | Cells may span columns. | |
| body row 3 | Cells may span rows. | • Cells |
| | | • contain |
| body row 4 | | blocks. |

| Inputs | | Output |
|--------|-------|--------|
| Α | В | A or B |
| False | False | False |
| True | False | True |
| False | True | True |
| True | True | True |

```
list-table:
:widths: 1 1 2
* - Type
 - Literal
 - Description
* - Boolean
 - true, false
* - Int
 - 3, 0x32
  - 32 bits integer
* - Float
 -3.14f
  - 32 bits floating point
* - Double
 - 3.14
  - 64 bits floating point
* - String
- "Hello world"
 - UTF-16 string
```

| Туре | Literal | Description |
|---------|---------------|------------------------|
| Boolean | true, false | |
| Int | 3, 0x32 | 32 bits integer |
| Float | 3.14f | 32 bits floating point |
| Double | 3.14 | 64 bits floating point |
| String | "Hello world" | UTF-16 string |

Table 1: Table caption

| Inputs | | Output |
|--------|-------|--------|
| Α | В | A or B |
| False | False | False |

12.2.8 Code

see also: https://build-me-the-docs-please.readthedocs.io/en/latest/Using_Sphinx/ShowingCodeExamplesInSphinx.html

```
import antigravity

def main():
    antigravity.fly()
    if __name__ == '__main__':
        main()
```

```
import antigravity

def main():
    antigravity.fly()
if __name__ == '__main__':
    main()
```

```
code-block: python
:linenos:
:caption: Code Blocks can have captions.

import antigravity

def main():
    antigravity.fly()
if __name__=='__main__':
    main()
```

Listing 1: Code Blocks can have captions.

```
import antigravity

def main():
    antigravity.fly()
    if __name__=='__main__':
        main()
```

```
code-block:: python
:linenos:
:lineno-start: 10

import antigravity

def main():
    antigravity.fly()
if __name__ == '__main__':
    main()
```

```
import antigravity

import antigravity

def main():
    antigravity.fly()

if __name__=='__main__':
    main()
```

12.2.9 Infoboxes

```
This is a Note Box
```

Note: This is a Note Box

```
This is a Warning Box
```

Warning: This is a Warning Box

```
This is a Important Box
```

Important: This is a Important Box

```
This is a See Also Box
```

See also:

This is a See Also Box

12.2.10 Special Formatting

```
versionadded:: 2.5
The *spam* parameter.

versionchanged:: 2.5
Feature description

deprecated:: 3.1
Use :func: spam instead.
```

New in version 2.5: The *spam* parameter.

Changed in version 2.5: Feature description

Deprecated since version 3.1: Use spam() instead.

12.2.11 Math

```
Inline math: 'a^2 + b^2 = c^2'.
```

Inline math $a^2 + b^2 = c^2$.

```
f(x) &= x^2\\
g(x) &= \frac{1}{x}\\
F(x) &= \int^a_b \frac{1}{3}x^3
```

$$f(x) = x^{2}$$

$$g(x) = \frac{1}{x}$$

$$F(x) = \int_{b}^{a} \frac{1}{3}x^{3}$$

12.2.12 Exclude

In order to exclude some parts for a certain output use the . . only:: output directive.

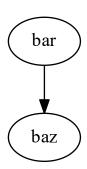
```
only:: html
only:: draft
only:: latex
only:: html or draft or latex
only:: html and draft
```

Important: This is needed for the all the *Wavedrom* code

12.2.13 GraphViz

Get more samples herer: https://graphviz.gitlab.io/gallery/

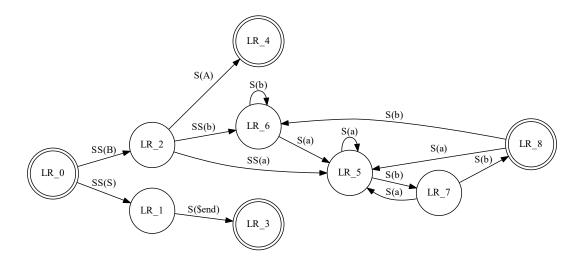
```
digraph foo {
    "bar" -> "baz";
}
```



```
digraph finite_state_machine {
    rankdir=LR;
    size="8,5"
    node [shape = doublecircle]; LR_0 LR_3 LR_4 LR_8;
    node [shape = circle];
    LR_0 -> LR_2 [ label = "SS(B)" ];
    LR_0 -> LR_1 [ label = "SS(S)" ];
    LR_1 -> LR_3 [ label = "S($end)" ];
    LR_2 -> LR_6 [ label = "SS(b)" ];
```

(continues on next page)

```
LR_2 -> LR_5 [ label = "SS(a)" ];
LR_2 -> LR_4 [ label = "S(A)" ];
LR_5 -> LR_7 [ label = "S(b)" ];
LR_6 -> LR_5 [ label = "S(a)" ];
LR_6 -> LR_6 [ label = "S(b)" ];
LR_7 -> LR_8 [ label = "S(a)" ];
LR_7 -> LR_8 [ label = "S(b)" ];
LR_7 -> LR_5 [ label = "S(a)" ];
LR_8 -> LR_6 [ label = "S(b)" ];
LR_8 -> LR_6 [ label = "S(b)" ];
```



12.2.14 Wavedrom

For more information see:

- · Wavedrom JSON Wiki
- Wavedrom Tutorial

Timing Diagrams

This documentation makes use of the sphinxcontrib-wavedrom plugin, So you can specify a timing diagram, or a register description with the WaveJSON syntax like so:

(continues on next page)

```
{ "name": 'clk4', "wave": 'xlh.L.Hx' ),
]}
```

and you get:

Note: if you want the Wavedrom diagram to be present in the pdf export, you need to use the "non relaxed" JSON dialect. long story short, no javascript code and use " arround key value (Eg. "name").

Register

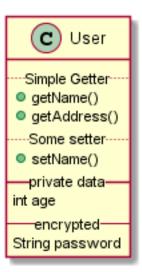
you can describe register mapping with the same syntax:

12.2.15 PlantUML

This documentation makes use of the sphinxcontrib.plantuml plugin, for more information see the sphinxcontrib.plantuml plugin and the PlantUML Webpage. For a small Cheatsheet for PlantUML see https://ogom.github.io/draw_uml/plantuml/

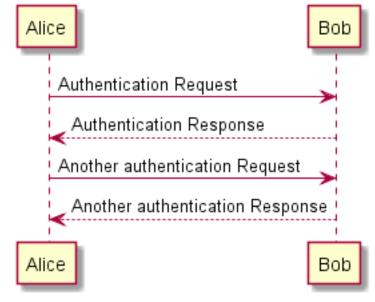
```
uml
class Foo1 {
 You can use
  several lines
 as you want
 and group
  --
 things together.
 You can have as many groups
 as you want
 End of class
class User {
 .. Simple Getter ..
  + getName()
 + getAddress()
  .. Some setter ..
  + setName()
   _ private data __
 int age
  -- encrypted --
  String password
```





```
Alice -> Bob: Authentication Request
Bob --> Alice: Authentication Response

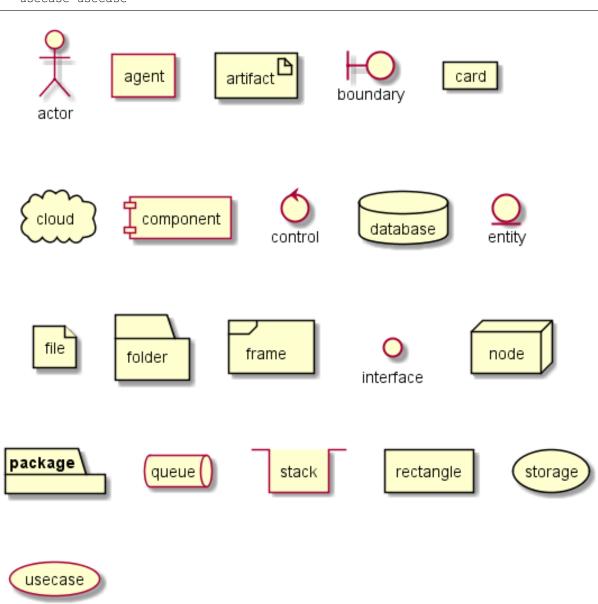
Alice -> Bob: Another authentication Request
Alice <-- Bob: Another authentication Response
```



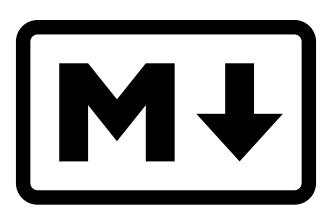
```
actor actor
agent agent
artifact artifact
boundary boundary
card card
cloud cloud
component component
control control
database database
entity entity
file file
folder folder
```

(continues on next page)

frame frame
interface interface
node node
package package
queue queue
stack stack
rectangle rectangle
storage storage
usecase usecase



Markdown



13.1 Github Markdown

GFM is a variant of markdown developed by Github.

- $\bullet \ https://help.github.com/articles/github-flavored-markdown$
- https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet This is intended as a quick reference and showcase. For more complete info, see John Gruber's original spec and the Github-flavored Markdown info page.

13.1.1 Table of Contents

Headers Emphasis Lists Links Images Code and Syntax Highlighting Tables Blockquotes Inline HTML Horizontal Rule Line Breaks YouTube Videos

13.1.2 Headers

```
# H1
## H2
### H3
#### H4
##### H5
##### H6

Alternatively, for H1 and H2, an underline-ish style:
```

(continues on next page)

```
Alt-H1
-----
Alt-H2
-----
```

13.2 H1

13.2.1 H2

H3

H4

H5

H6

Alternatively, for H1 and H2, an underline-ish style:

13.2.2 Emphasis

```
Emphasis, aka italics, with *asterisks* or _underscores_.

Strong emphasis, aka bold, with **asterisks** or _underscores_.

Combined emphasis with **asterisks and _underscores_**.

Strikethrough uses two tildes. ~~Scratch this.~~
```

Emphasis, aka italics, with asterisks or underscores.

Strong emphasis, aka bold, with asterisks or underscores.

Combined emphasis with asterisks and underscores.

Strikethrough uses two tildes. ~~Scratch this.~~

13.2.3 Lists

(continues on next page)

```
Note that this line is separate, but within the same paragraph.

(This is contrary to the typical GFM line break behaviour, where trailing 

⇒spaces are not required.)

* Unordered list can use asterisks

- Or minuses

+ Or pluses
```

- 1. First ordered list item
- 2. Another item
- Unordered sub-list.
- 1. Actual numbers don't matter, just that it's a number
- 2. Ordered sub-list
- 3. And another item.

You can have properly indented paragraphs within list items. Notice the blank line above, and the leading spaces (at least one, but we'll use three here to also align the raw Markdown).

To have a line break without a paragraph, you will need to use two trailing spaces. Note that this line is separate, but within the same paragraph. (This is contrary to the typical GFM line break behaviour, where trailing spaces are not required.)

- Unordered list can use asterisks
- · Or minuses
- Or pluses

13.2.4 Links

There are two ways to create links.

```
[I'm an inline-style link] (https://www.google.com)

[I'm an inline-style link with title] (https://www.google.com "Google's Homepage")

[I'm a reference-style link] [Arbitrary case-insensitive reference text]

[I'm a relative reference to a repository file] (../blob/master/LICENSE)

[You can use numbers for reference-style link definitions] [1]

Or leave it empty and use the [link text itself].

URLs and URLs in angle brackets will automatically get turned into links.

http://www.example.com or <http://www.example.com> and sometimes
example.com (but not on Github, for example).

Some text to show that the reference links can follow later.

[arbitrary case-insensitive reference text]: https://www.mozilla.org
[1]: http://slashdot.org
[link text itself]: http://www.reddit.com
```

I'm an inline-style link

I'm an inline-style link with title

I'm a reference-style link

```
[I'm a relative reference to a repository file]("../../README.md")
```

You can use numbers for reference-style link definitions

Or leave it empty and use the link text itself.

URLs and URLs in angle brackets will automatically get turned into links. http://www.example.com or http://www.example.com and sometimes example.com (but not on Github, for example).

Some text to show that the reference links can follow later.

13.2.5 Images

```
Here's our logo (hover to see the title text):

Inline-style:
![alt text] (https://github.com/adam-p/markdown-here/raw/master/src/common/images/
icon48.png "Logo Title Text 1")

Reference-style:
![alt text][logo]

[logo]: https://github.com/adam-p/markdown-here/raw/master/src/common/images/
icon48.png "Logo Title Text 2"
```

Here's our logo (hover to see the title text):

Inline-style: 💆

Reference-style: 5

13.2.6 Code and Syntax Highlighting

Code blocks are part of the Markdown spec, but syntax highlighting isn't. However, many renderers – like Github's and *Markdown Here* – support syntax highlighting. Which languages are supported and how those language names should be written will vary from renderer to renderer. *Markdown Here* supports highlighting for dozens of languages (and not-really-languages, like diffs and HTTP headers); to see the complete list, and how to write the language names, see the highlight.js demo page.

```
Inline `code` has `back-ticks around` it.
```

Inline code has back-ticks around it.

Blocks of code are either fenced by lines with three back-ticks ```, or are indented with four spaces. I recommend only using the fenced code blocks – they're easier and only they support syntax highlighting.

```
var s = "JavaScript syntax highlighting";
alert(s);
```

```
s = "Python syntax highlighting"
print s
```

```
No language indicated, so no syntax highlighting in Markdown Here (varies on 

→ Github).

But let's throw in a <b>tag</b>.
```

13.2.7 Tables

Tables aren't part of the core Markdown spec, but they are part of GFM and *Markdown Here* supports them. They are an easy way of adding tables to your email – a task that would otherwise require copy-pasting from another application.

Colons can be used to align columns.

There must be at least 3 dashes separating each header cell. The outer pipes (I) are optional, and you don't need to make the raw Markdown line up prettily. You can also use inline Markdown.

```
Markdown | Less | Pretty — | — | — Still | renders | nicely 1 | 2 | 3
```

13.2.8 Blockquotes

```
> Blockquotes are very handy in email to emulate reply text.
> This line is part of the same quote.

Quote break.

> This is a very long line that will still be quoted properly when it wraps. Oh_
→boy let's keep writing to make sure this is long enough to actually wrap for_
←veryone. Oh, you can *put* **Markdown** into a blockquote.
```

Blockquotes are very handy in email to emulate reply text. This line is part of the same quote.

Quote break.

This is a very long line that will still be quoted properly when it wraps. Oh boy let's keep writing to make sure this is long enough to actually wrap for everyone. Oh, you can *put* **Markdown** into a blockquote.

13.2.9 Inline HTML

You can also use raw HTML in your Markdown, and it'll mostly work pretty well.

13.2.10 Horizontal Rule

```
Three or more...

---

Hyphens

***

Asterisks

____
Underscores
```

Three or more...

Hyphens

Asterisks

Underscores

13.2.11 Line Breaks

My basic recommendation for learning how line breaks work is to experiment and discover – hit <Enter> once (i.e., insert one newline), then hit it twice (i.e., insert two newlines), see what happens. You'll soon learn to get what you want. "Markdown Toggle" is your friend.

Here are some things to try out:

Here's a line for us to start with.

This line is separated from the one above by two newlines, so it will be a separate paragraph.

This line is also begins a separate paragraph, but... This line is only separated by a single newline, so it's a separate line in the *same paragraph*.

(Technical note: Markdown Here uses GFM line breaks, so there's no need to use MD's two-space line breaks.)

13.2.12 YouTube Videos

They can't be added directly but you can add an image with a link to the video like this:

Or, in pure Markdown, but losing the image sizing and border:

```
[![IMAGE ALT TEXT HERE](http://img.youtube.com/vi/YOUTUBE_VIDEO_ID_HERE/0.

→jpg)](http://www.youtube.com/watch?v=YOUTUBE_VIDEO_ID_HERE)
```

Referencing a bug by #bugID in your git commit links it to the slip. For example #1.

Multimedia



15 Security



16

Past Time



16.1 Programmer Jokes

- Christmas and Halloween
- 10 Kind of People
- Error Free Programs
- Boolean Answer
- Programmer Checks
- Debugging
- HTML Tags
- Teacher Punishement
- Accelerate a computer

16.1.1 Christmas and Halloween

Question Why do programmers always mix up Halloween and Christmas?

Answer

- 31 Dec Christmas
- 25 Okt Halloween

16.1.2 10 Kind of People

Question

There are 10 types of people in this world. Those who understand binary and those who don't.

Answer0b10 = 2

16.1.3 Error Free Programs

Question There are two ways to write error-free programs; only the third one works.

Answer There is no error free program, therefore the anwer is also wrong

16.1.4 Boolean Answer

Question The best thing about a Boolean is even if you are wrong, you are only off by a bit.

Answer Boolean = 0 or 1 only of by 1bit

16.1.5 Programmer Checks

Question A good programmer is someone who always looks both ways before crossing a one-way street.

Answer Programmers can't make assumptions, they have to check everything

16.1.6 Debugging

Question Debugging: Removing the needles from the haystack.

Answer Debugging is removing bugs form a program. Bugs are hard to find like needles

16.1.7 HTML Tags

Question



Fig. 1: <DIV>Q: How to you annoy a web developer? </SPAN`>

Answer HTML Tags are wrong DIV and SPAN means the same. Above code is wrong.

16.1.8 Teacher Punishement

Question

```
# include <5 (dio.m)
int main(void)

{
  int count;
  for (count = 1; count <= 500; count ++)
    printf ("I will not throw paper dirplanes in class.");
  return 0;
}

***MOD 10-3**
```

Fig. 2: teacher_punishement

Answer In this joke, his teacher probably gave him the punishment "Write 'I will not throw paper airplanes in class.' on the board 500 times."

```
#include <stdio.h>
int main(void)
{
  int count;
  for (count = 1; count <= 500; count++)
    printf("I will not throw paper airplanes in class.");
  return 0;
}</pre>
```

16.1.9 Accelerate a computer

Question The best method for accelerating a computer is the one that boosts it by 9.8 m/s2

Answer Let it drop. Earth gravity accelerates it by 9.8m/s2

SQL Naming Question

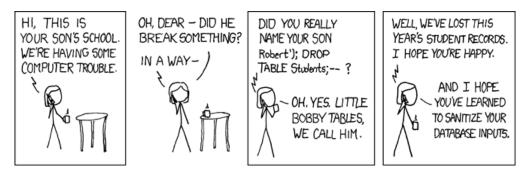
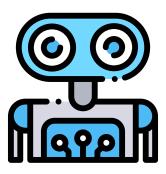


Fig. 3: sql_name

Answer This joke has to do with SQL, which are commands used to control databases as well as a common hack used against insecure sites, called SQL Injection.



17.1 About

17.1.1 Authors

• tschinz - Github Profile

17.1.2 Find me at

- Github
- Flickr
- Twitter @tschinz

17.2 Credits

On this website information, images and documents are used. Hereafter these credits are all listed. Icons made by Freepik from Flaticon

17.3 How to use Sphinx Documentation

- Sphinx Requirements
- How to create a new Sphinxdoc
- How to Build Sphinxdoc locally
 - Without pipenv
 - With pipenv
 - Continuous Build
- Commit to Repository
- Continuous Integration(CI)

17.3.1 Sphinx Requirements

- make
 - Windows GnuWin32
 - Linux

```
sudo apt-get install build-essential
```

- Python 3
 - Python
 - Anaconda
- Python Modules (can be installed with pipenv)

```
pip install sphinx
pip install sphinx-rtd-theme
pip install sphinxcontrib-wavedrom
pip install sphinxcontrib-plantuml
pip install recommonmark
```

- Latex Tools (only for latex build)
 - Windows
 - * MikTex
 - * TexStudio
 - Linux

```
sudo apt install texlive-fonts-recommended texlive-latex-recommended 

→texlive-latex-extra
```

- Inkscape (for .svg to .pdf and to .png conversion)
 - Windows Inkscape
 - Linux

```
sudo apt-get install inkscape
```

17.3.2 How to create a new Sphinxdoc

```
sphinx-quickstart
```

17.3.3 How to Build Sphinxdoc locally

Without pipenv

- Install requirements see: Sphinx Requirements
- cd to the git folder
- Generate the desired output

```
make  # list all the available output format
make help  # list all the available output format

make html  # for html
make latex  # for latex
make latexpdf # for latex (will require latexpdf installed)

make clean  # cleans all generated file, TODO before committing
make clean-images # cleans all autogerated png and pdf files
```

With pipenv

- Install requirements Sphinx Requirements
- Create a virtual environment with pipenv (will use the Pipfile for installing the necessary packages)

```
pipenv install
```

• then you can build the documentation

```
pipenv run make html
```

• if you want run make multiple times, prepone pipenv run on each command can be annoying, you can spawn a subshell with

```
pipenv shell
```

• and then you can use make the usual way

```
make  # list all the available output format
make help  # list all the available output format

make html  # for html
make latex  # for latex
make latexpdf # for latex (will require latexpdf installed)

make clean  # cleans all generated file, TODO before committing
make clean-images # cleans all autogerated png and pdf files
```

all the outputs will be in _build folder

• html: _build/html

• pdf & tex: _build/latex

Continuous Build

During developement or creation of a page, the script build-loop.bash will rebuild the webpage every X seconds. In this way a constant preview of the page can be shown.

17.3.4 Commit to Repository

Before performing a commit the following steps are required:

• Verify the html documentation local How to Build Sphinxdoc locally

make html

- Solve all build Warnings and Errors display during build in the commandline
- Generate pdf

make latexpdf

• Clean the repo from generated files

make clean

• Commit and push the changes SPL Knowhow CI

17.3.5 Continuous Integration(CI)

The .travis.yml will run on each master commit and create a _build/ folder which will be pushed onto the branch gh-pages and consequently be used by github to displayed static html pages.

17.4 HACK this documentation

- New Documentation Section
- Example Section
 - Section Images
 - Write the contents

If you want to add your page to this documentation you need to add your source file in the appropriate section. Every main section has its own folder structure and its own img/ folder containing all images for this section.

This documentation uses a recursive index tree: every folder have a special index.rst file that tell sphinx witch file, and in what order put it in the documentation tree.

If you don't have enough knowledge about ReStructuredText then you can also use the pandoc translator or use the internal *Cheatsheet*

17.4.1 New Documentation Section

If you want to add a new section, you need to specify in the main index.rst, the section/index.rst file of the new section.

```
: toctree::
    :hidden:
    :glob:
    :maxdepth: 2
    :titlesonly:
    :caption: Content

linux/index
    mac/index
    windows/index
    tools/index
    coding/index
    writing/index
    multimedia/index
    security/index
    about/index
```

The section name should be the same as the folder name, but for Sphinx this is not required. Sphinx will take the name of the section from the title of the section/index.rst file.

17.4.2 Example Section

I want to document the new topic in SPL Knowhow repo, and want to create a section for it; let's call it Section So I need to create a folder named section/ (name is not important), and in it create a section/index.rst file like:

Note: The .. toctree:: directive accept some parameters, in this case :glob: makes so you can use the * to include all the remaining files.

Note: The file path is relative to the index file, if you want to specify the absolute path, you need to prepend /

Now I can add additional ReST files like section/intro.rst and other files like section/section_part_1.rst, ssection/ection_part_2.rst, etc.

Section Images

Add an image folder in the section folder section/img, in case of additional documents ass a section/docs folder too.

Write the contents

That's it, now you can add all you want in the new section section and all pages will show up in the documentation automatically.

17.5 License

Copyright (c) 2020, tschinz All rights reserved.

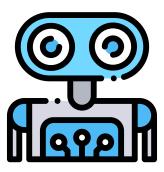
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name zawiki nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

17.5. License 128

Welcome to Zawiki Knowledge Database



This Repo is a collection of markdown and ReStructuredText pages. Here you can find various informations about topics I've always forget. This pages let me help to remember less but know more.

18.1 Site purpose and structure

18.1.1 Getting started

Want to try it for yourself? Then jump to the *getting started* page and have fun, but first you need to learn *ReStructured Text* !!!

You can view the content as a:

- Webpage
- PDF
- Repo

18.1.2 Known Issues / TODOs

- Github CI not working for PDF creation
- Missing pages from original Zawiki
- missing links to config repo