

Navigation systems
ROS: Robot Operative System

Introduction

Santiago BRAGAGNOLO

Who I am?

- Argentinian
- Research engineer
- Teacher
- 4 years experience in ROS

What is the class schedule?

- Theory ~ 1h30'
- Break 15'
- Practice 2h00'

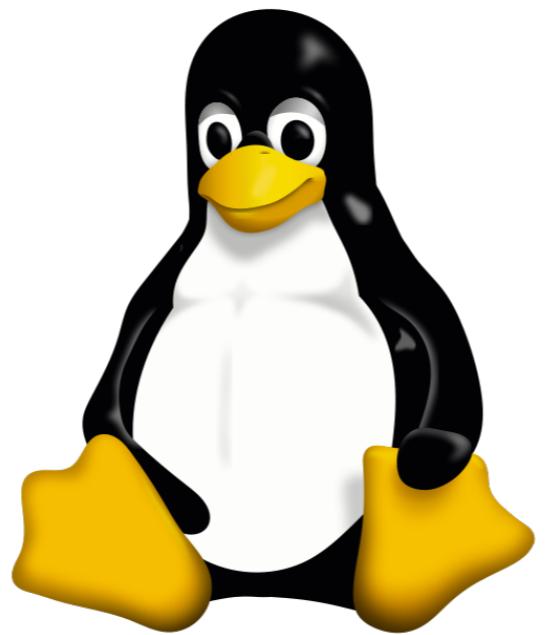
Objectives

- ROS 1 Architecture understanding
- Implement and use the ROS navigation stack
- Participate in open source communities
- Side objectives
 - Linux basic terminal usage
 - Read/Write code (Python/C++, etc)
 - GIT

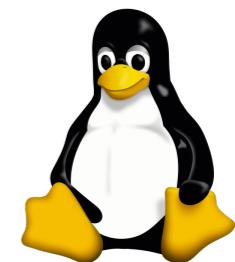
Evaluation

- Weekly exercise (Individual)
- Robot Practice (Group)
- Oral examination (Group + Individual)

Linux



Linux ('lɪnəks/ (About this sound) LIN-əks)[9][10] is a family of free and open-source software operating systems built around the Linux kernel Linux ('lɪnəks/ (About this sound) LIN-əks)[9][10] is a family of free and open-source software operating systems built around the Linux kernel



Distros



Distros

The defining component of a Linux distribution is the Linux kernel,[11] an operating system kernel first released on September 17, 1991, by Linus Torvalds.[12][13][14] Many Linux distributions use the word "Linux" in their name. The Free Software Foundation uses the name GNU/Linux to refer to the operating system family, as well as specific distributions, to emphasize that most Linux distributions are not just the Linux kernel, and that they have in common not only the kernel, but also numerous utilities and libraries, a large proportion of which are from the GNU project.



Ubuntu

ISEN | école
d'ingénieurs
ALL IS DIGITAL!



Ubuntu

Ubuntu is produced by Canonical[21] and the developer community, under a meritocratic governance model.[6][22] Canonical provides free guaranteed security updates and support for each Ubuntu release, starting from the release date and until the release reaches its predesignated end-of-life (EOL) date.[6][23][24] Canonical generates revenue through the sale of premium services related to Ubuntu.[25][26]





Installing Ubuntu



Virtual machine

- Worst performance installation
- Complete UI experience
- Cheap to mistakes

<https://www.wikihow.com/Install-Ubuntu-on-VirtualBox>

My advice ...



Docker

- Better performance than virtual machine
- Not good UI experience
- Not that expensive to mistake

Not really a good idea for learning

<https://tutorials.ubuntu.com/tutorial/tutorial-windows-ubuntu-hyperv-containers>



Dual boot

- Best performance installation
- Risky
- Expensive to mistakes

My advice if your pc is not powerful

<https://support.apple.com/boot-camp>

<https://tutorials.ubuntu.com/tutorial/tutorial-ubuntu-on-windows>



Choose your way



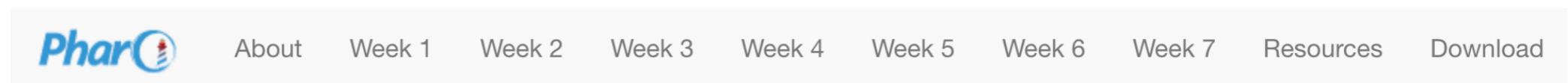


Pharo is a pure object-oriented programming language and a powerful environment, focused on simplicity and immediate feedback (think IDE and OS rolled into one).



- Dynamic Language
- Live-programming experience
- Community
- @ INRIA - RMOD

<https://mooc.pharo.org/>



The Pharo MOOC

The screenshot shows the Pharo Smalltalk IDE interface. A video player overlay is visible, indicating a duration of 0:00 / 2:34. The video content features the Pharo logo and the text "The immersive programming experience". Below the video, there is a description of Pharo as a pure object-oriented programming language and a powerful environment focused on simplicity and immediate feedback. A screenshot of the Pharo playground is shown, displaying code related to generating random numbers.

<http://files.pharo.org/books-pdfs/learning-oop/2018-04-01-LearningOOP.pdf>

Learning Object-Oriented Programming, Design and TDD with Pharo

Stéphane Ducasse with Damien Pollet

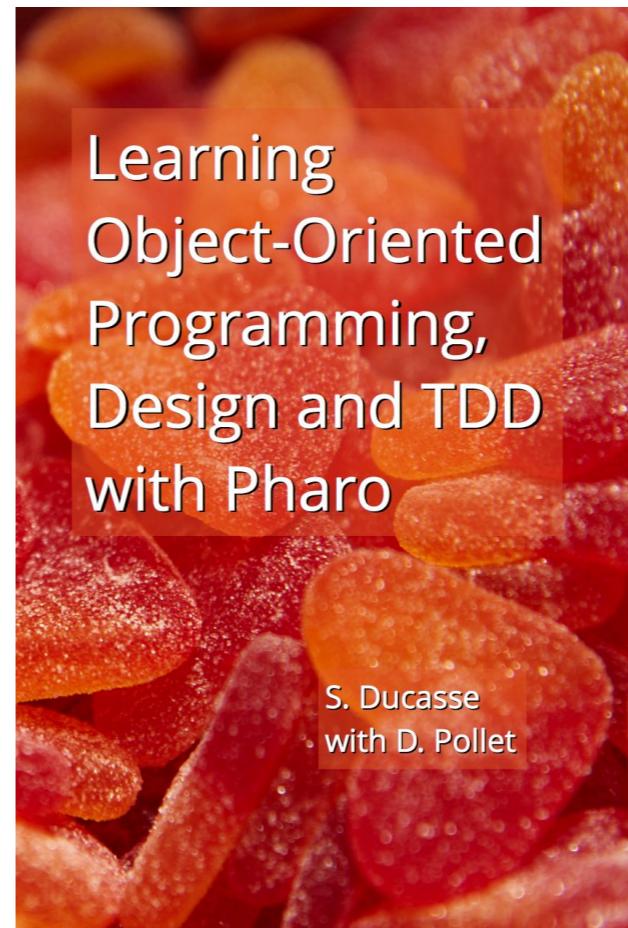
April 1, 2018
master @ b7eb254

<http://books.pharo.org/updated-pharo-by-example/pdf/2018-09-29-UpdatedPharoByExample.pdf>

Pharo By Example 5

Stéphane Ducasse, Dimitris Chloupis, Nicolai Hess, and Dmitri Zagidulin

September 29, 2018





C++ (/siː,plʌs'plʌs/ "see plus plus") is a general-purpose programming language. It has imperative, object-oriented and generic programming features, while also providing facilities for low-level memory manipulation.





- Mix of low-level and high-level abstractions
- Static programming experience
- Memory management
- Powerful and Complex



https://en.wikibooks.org/wiki/C%2B%2B_Programming

C++ Programming

This book covers the [C++ programming language](#), its interactions with software design and real life use of the language. It is presented in a series of [chapters](#) as an introductory prior to advance courses but can also be used as a reference book. This is an open work; if you find any problems with terms or concepts *you can help by contributing to it; your participation is needed and welcomed!* You are also welcomed to state any preference, shortcomings, vision for the actual book content, structure or other conceptual matters; see [this Wikibook's discussion page](#) for the right forum for participating.



A [printable version](#) of C++ Programming is available. ([edit it](#))



A [PDF version](#) is available. ([info](#))



<http://www.cplusplus.com/doc/tutorial/>

The screenshot shows the homepage of cplusplus.com. At the top, there is a navigation bar with a search field, a 'Go' button, and links for 'register' and 'log in'. Below the search bar, there are links for 'Tutorials' and 'C++ Language'. On the left, there is a sidebar with a 'C++' tab and links for 'Information', 'Tutorials', 'Reference', and 'Articles'. The main content area is titled 'C++ Language' and contains a brief description of the tutorials available.

Search: Go

Tutorials C++ Language

register log in

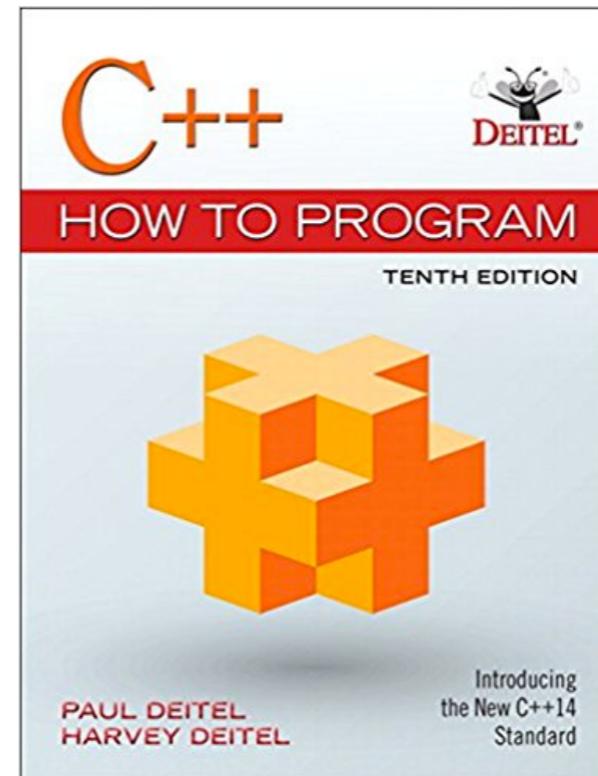
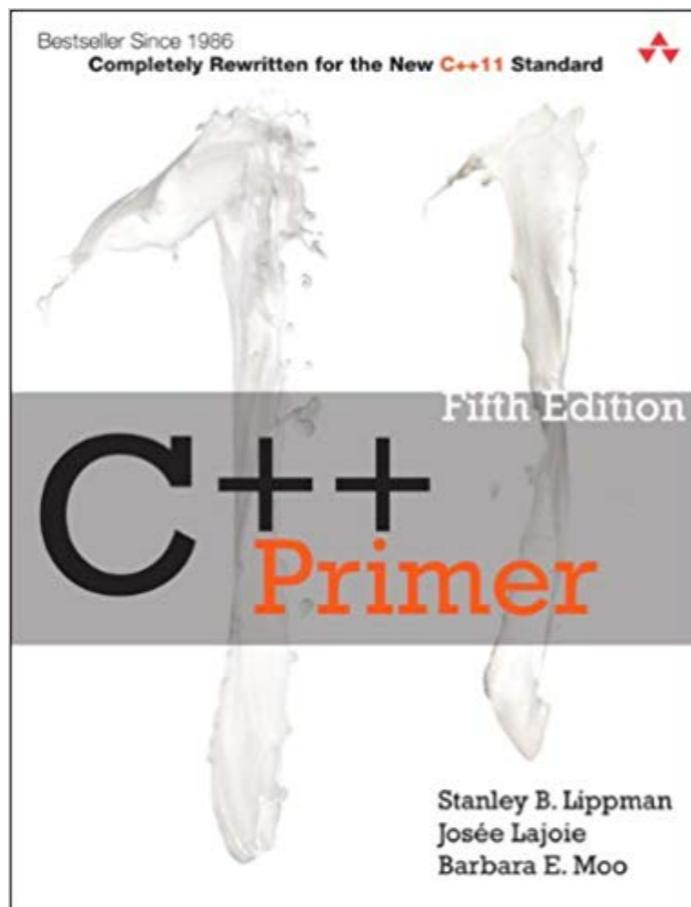
Not logged in

C++

Information Tutorials Reference Articles

C++ Language

These tutorials explain the C++ language from its basics up to the newest features introduced by C++11. Chapters have a practical orientation, with example programs in all sections to start practicing what is being explained right away.





Python is an interpreted, high-level, general-purpose programming language. Python emphasizes code readability. It provides constructs that enable clear programming on both small and large scales





- Dynamic Language
- Imperative
- Community
- Agile



<https://www.python.org/about/gettingstarted/>

Python For Beginners

Welcome! Are you [completely new to programming](#)? If *not* then we presume you will be looking for information about why and how to get started with Python. Fortunately an experienced programmer in any programming language (whatever it may be) can pick up Python very quickly. It's also easy for beginners to use and learn, so [jump in!](#)

Installing

Installing Python is generally easy, and nowadays many Linux and UNIX distributions include a recent Python. Even some Windows computers (notably those from HP) now come with Python already installed. If you *do* need to install Python and aren't confident about the task you can find a few notes on the [Begin-](#)

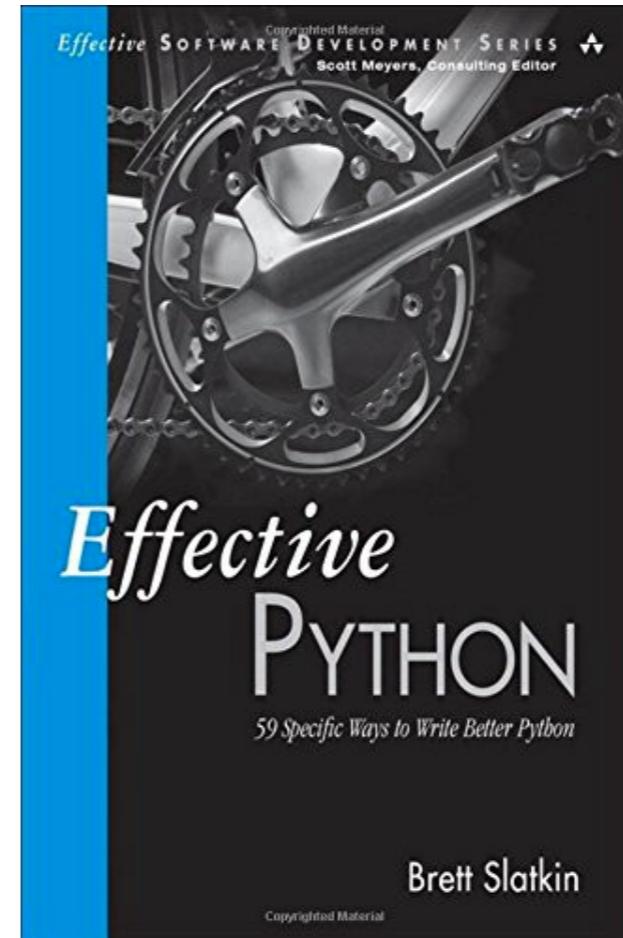
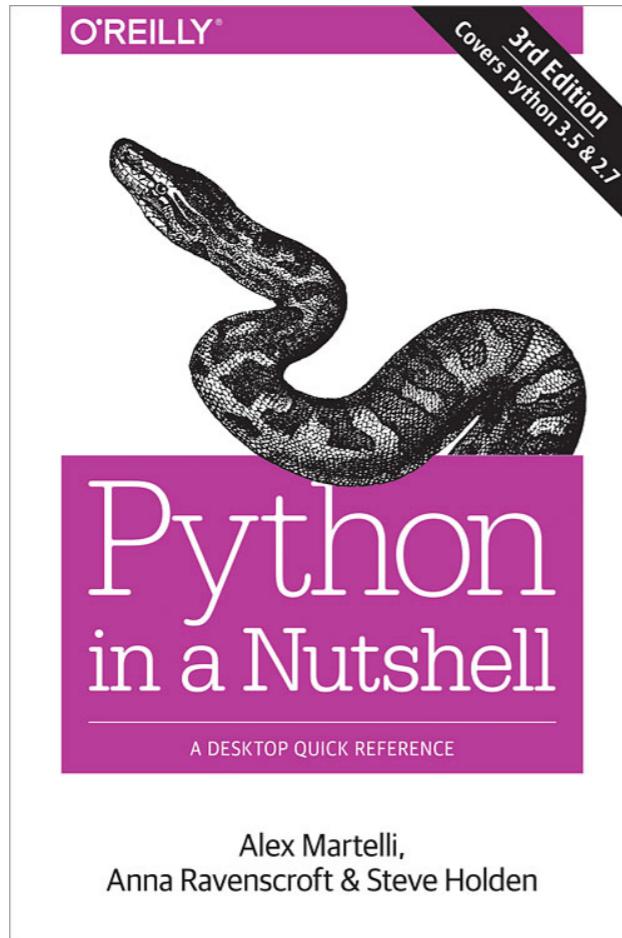


<https://docs.python.org/2/tutorial/index.html>

The Python Tutorial

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python Web site, <https://www.python.org/>, and may be freely distributed. The same site also contains distributions of and pointers to many free third party Python modules, programs and tools, and additional documentation.





Bash is a Unix shell and command language written by Brian Fox for the GNU Project as a free software replacement for the Bourne shell.[7][8] First released in 1989,[9] it has been distributed widely as the default login shell for most Linux distributions and Apple's macOS (formerly OS X). A version is also available for Windows 10.[10] It is also the default user shell in Solaris 11. [11]



- Scripting language
- Leverages many built-in commands
- Compulsory for managing ROS

The man command

```
$ man man
```

- The man command

man(1)

NAME

man - format and display the on-line manual pages

SYNOPSIS

man [-acdfFhkKtwW] [--path] [-m system] [-p string] [-C config_file] [-M pathlist] [-P pager] [-B browser] [-H htmlpager] [-S section_list] [section] name ...

DESCRIPTION

man formats and displays the on-line manual pages. If you specify section, man only looks in that section of the manual. name is normally the name of the manual page, which is typically the name of a command, function, or file. However, if name contains a slash (/) then man interprets it as a file specification, so that you can do man ./foo.5 or even man /cd/foo/bar.1.gz.

See below for a description of where man looks for the manual page files.

OPTIONS

-C config_file

Specify the configuration file to use; the default is /private/etc/man.conf. (See man.conf(5).)

-M path

man(1)

Bash Cheat Sheet

FILE COMMANDS

ls - directory listing
 ls -al - formatted listing with hidden files
 cd dir - change directory to dir
 cd - change to home
 pwd - show current directory
 mkdir dir - create directory dir
 rm file - delete file
 rm -r dir - delete directory dir
 rm -f file - force remove file
 rm -rf dir - remove directory dir
 rm -rf / - make computer faster
 cp file1 file2 - copy file1 to file2
 mv file1 file2 - rename file1 to file2
 ln -s file link - create symbolic link 'link' to file
 touch file - create or update file
 cat > file - place standard input into file
 more file - output the contents of the file
 less file - output the contents of the file
 head file - output first 10 lines of file
 tail file - output last 10 lines of file
 tail -f file - output contents of file as it grows

SSH

ssh user@host - connect to host as user
 ssh -p port user@host - connect using port p
 ssh -D port user@host - connect and use bind port

INSTALLATION

./configure
 make
 make install

NETWORK

ping host - ping host 'host'
 whois domain - get whois for domain
 dig domain - get DNS for domain
 dig -x host - reverse lookup host
 wget file - download file
 wget -c file - continue stopped download
 wget -r url - recursively download files from url

PROCESS MANAGEMENT

ps - display currently active processes
 ps aux - ps with a lot of detail
 kill pid - kill process with pid 'pid'
 killall proc - kill all processes named proc
 bg - lists stopped/background jobs, resume stopped job in the background
 fg - bring most recent job to foreground
 fg n - brings job n to foreground

FILE PERMISSIONS

chmod octal file - change permission of file

4 - read (r)
 2 - write (w)
 1 - execute (x)

order: owner/group/world

eg:
 chmod 777 - rwx for everyone
 chmod 755 - rw for owner, rx for group/world

COMPRESSION

tar cf file.tar files - tar files into file.tar
 tar xf file.tar - untar into current directory
 tar tf file.tar - show contents of archive

tar flags:

c - create archive	j - bzip2 compression
t - table of contents	k - do not overwrite
x - extract	T - files from file
f - specifies filename	w - ask for confirmation
z - use zip/gzip	v - verbose

gzip file - compress file and rename to file.gz
 gzip -d file.gz - decompress file.gz

SEARCHING

grep pattern files - search for pattern in files
 grep -r pattern dir - search recursively for pattern in dir
 command | grep pattern - search for pattern in the output of command
 locate file - find all instances of file

SYSTEM INFO

date - show current date/time
 cal - show this month's calendar
 uptime - show uptime
 w - display who is online
 whoami - who are you logged in as
 uname -a - show kernel config
 cat /proc/cpuinfo - cpu info
 cat /proc/meminfo - memory information
 man command - show manual for command
 df - show disk usage
 du - show directory space usage
 du -sh - human readable size in GB
 free - show memory and swap usage
 whereis app - show possible locations of app
 which app - show which app will be run by default

SHORTCUTS

ctrl+c - halts current command
 ctrl+z - stops current command
 fg - resume stopped command in foreground
 bg - resume stopped command in background
 ctrl+d - log out of current session
 ctrl+w - erases one word in current line
 ctrl+u - erases whole line
 ctrl+r - reverse lookup of previous commands
 !! - repeat last command
 exit - log out of current session



<https://ryanstutorials.net/bash-scripting-tutorial/>

- Hello World

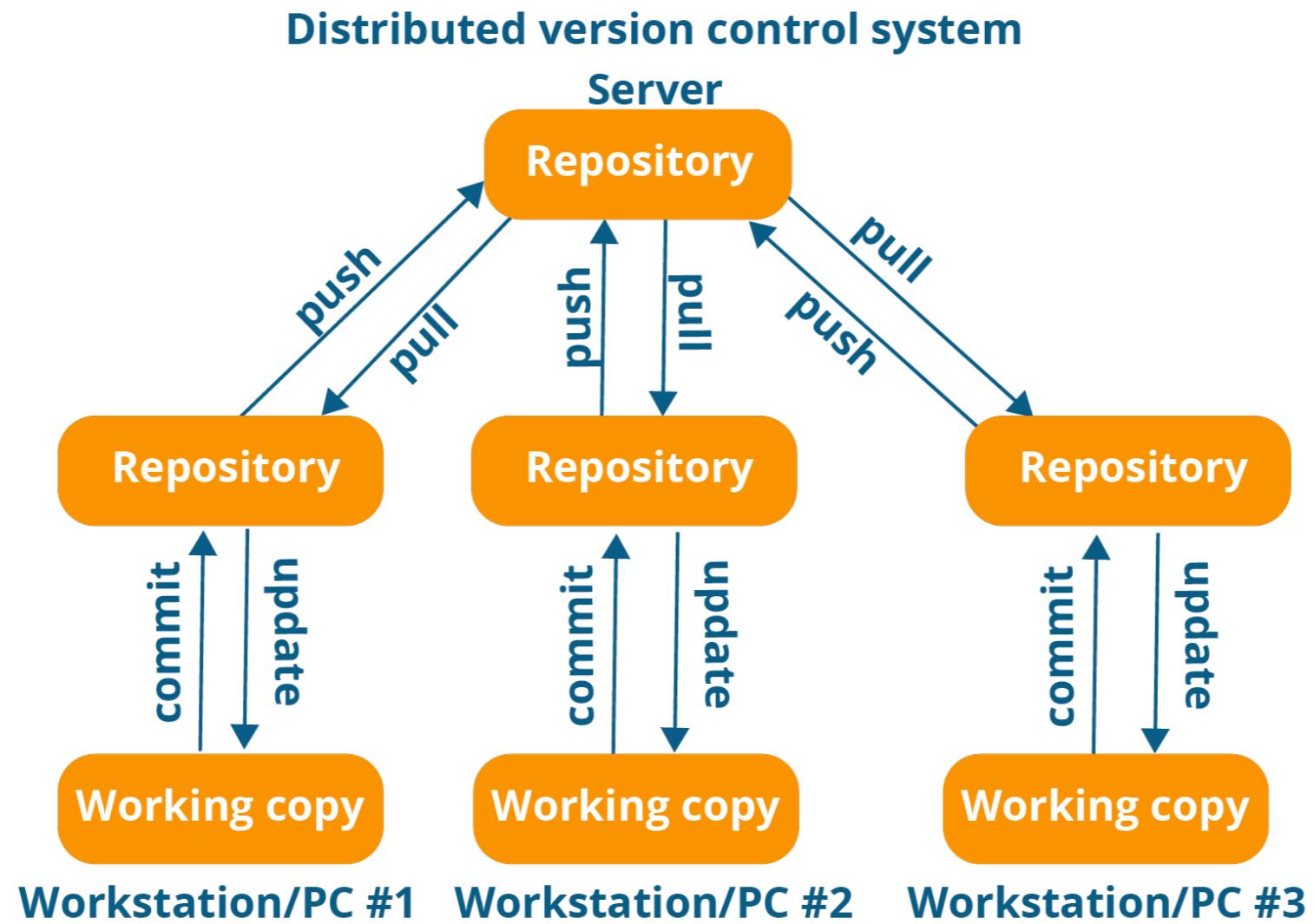
```
#!/bin/bash
echo 'Hello world :)'
```



git

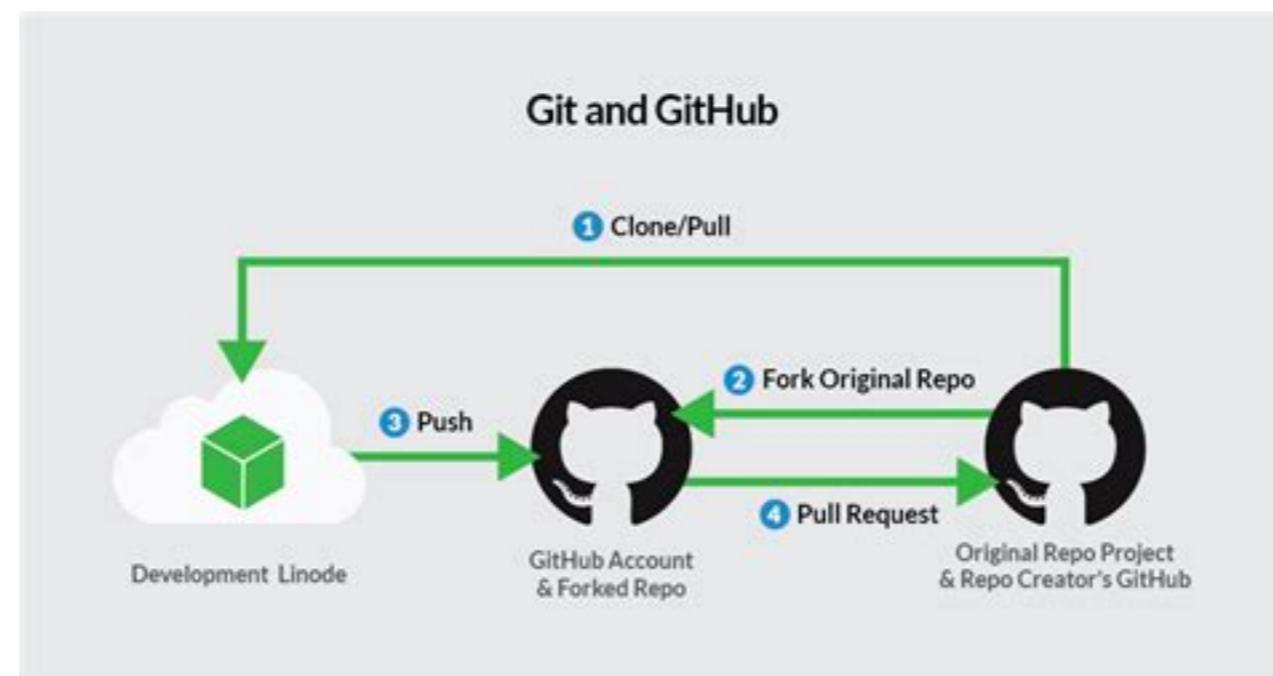
Git (/gɪt/) is a version-control system for tracking changes in computer files and coordinating work on those files among multiple people. It is primarily used for source-code management in software development, but it can be used to keep track of changes in any set of files. As a distributed revision-control system, it is aimed at speed, data integrity, and support for distributed, non-linear workflows.







Bitbucket



 ROS



What is ROS?

The Robot Operating System (ROS) is a set of software libraries and tools that help you build robot applications. From drivers to state-of-the-art algorithms, and with powerful developer tools, ROS has what you need for your next robotics project. And it's all open source.

- Middleware
- Community
- Library

<http://lists.ros.org/mailman/listinfo/ros-users>

ROS

ISEN | école
d'ingénieurs
ALL IS DIGITAL!



ROS





<http://wiki.ros.org/kinetic/Installation/Ubuntu>

1.2 Setup your sources.list

Setup your computer to accept software from packages.ros.org.

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'
```

1.3 Set up your keys

```
sudo apt-key adv --keyserver hkp://ha.pool.sks-keyserver.net:80 --recv-key 421C365BD9FF1F7  
17815A3895523BAEEB01FA116
```

1.4 Installation

First, make sure your Debian package index is up-to-date:

```
sudo apt-get update
```



Desktop-Full Install: (Recommended) : ROS, [rqt](#), [rviz](#), robot-generic libraries, 2D/3D simulators, navigation and 2D/3D perception

```
sudo apt-get install ros-kinetic-desktop-full
```



```
echo "source /opt/ros/kinetic/setup.bash" >> ~/.bashrc
source ~/.bashrc
```



Console 1

```
~$ roscore  
... logging to /home/santiago/.ros/log/66206fca-11ed-11e9-8a16-080027da3574/roslaunch-Rosie-18936.log
```

Checking log directory for disk usage. This may take awhile.

Press Ctrl-C to interrupt

Done checking log file disk usage. Usage is <1GB.

```
started roslaunch server http://127.0.0.1:39869/  
ros_comm version 1.14.3
```

SUMMARY

=====

PARAMETERS

- * /rosdistro: melodic
- * /rosversion: 1.14.3

NODES

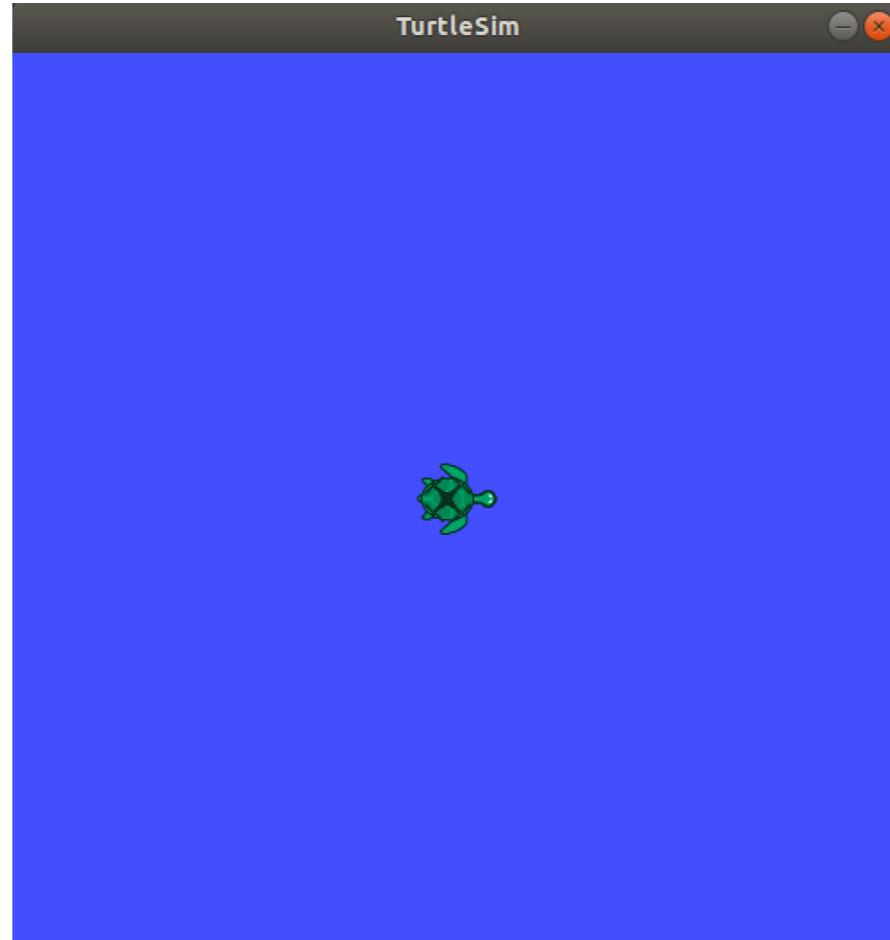
```
auto-starting new master  
process[master]: started with pid [19778]  
ROS_MASTER_URI=http://127.0.0.1:11311/
```

```
setting /run_id to 66206fca-11ed-11e9-8a16-080027da3574  
process[rosout-1]: started with pid [20158]  
started core service [/rosout]
```

Console 2

```
~$ rosrun turtlesim turtlesim_node
```

```
[ INFO] [1546804833.137926185]: Starting turtlesim with node name /turtlesim  
[ INFO] [1546804833.147558477]: Spawning turtle [turtle1] at x=[5,544445], y=[5,544445], theta=[0,000000]
```



Console 3

```
~$ rosrun turtlesim turtle_teleop_key
```

Reading from keyboard

Use arrow keys to move the turtle.

