

# Hello Python!

A crash course introduction to  
the computer, programming,  
and the Python programming  
language

ICTP – ESP – Graziano Giuliani <ggiulian@ictp.it>

27/05/23

PWF Cameroon  
School on Climate Science  
5-9 June 2023  
University of Dschang



The Abdus Salam  
International Centre  
for Theoretical Physics

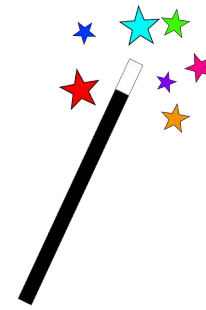


# The Computer

- A machine is a physical system using power to apply forces and control movement to perform an action.
- A digital bit is the minimal amount of information representing a logical state of a system with two possible values [0-1]
- A computer is an electronic digital machine carrying out logical operations on binary bits of information
- Switching a bit is an action and work must be done on the system to change it



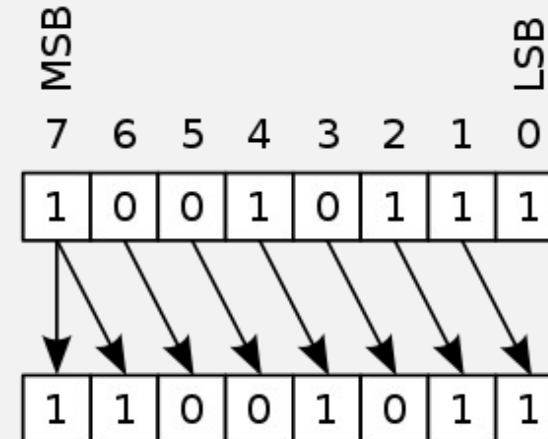
# Computer Program



- A computer program is a sequence of instructions modifying a string of bits representing the status of a system. The initial status is the input of the program, the final state is the output of the program
- The instructions are stored in bit patterns themselves, following conventions defined by the designer of a piece at the hearth of the computer called the **Processor**.
- The program and the input data are sequences of bits registered in magnetic **Memory** on the computer. The output of the program is registered in the memory, either updating the initial system status or creating a new status.

Where is the magic?

- The information is CODED : a → 01100001
- There are codes for
  - **Characters**
  - **Exact Integer numbers**
  - **Approximate representation of Real numbers**
- Special processing units in the processor can perform
  - **Character string manipulation**
  - **Arithmetic operation**
  - **Mathematical complex evaluation**



# Write a program

Hardware : the metal box in which the “magic” happens

Software : the instructions describing how to go from input to output state

Operating System : The software part directly controlling the Hardware circuits

Application : The program you use

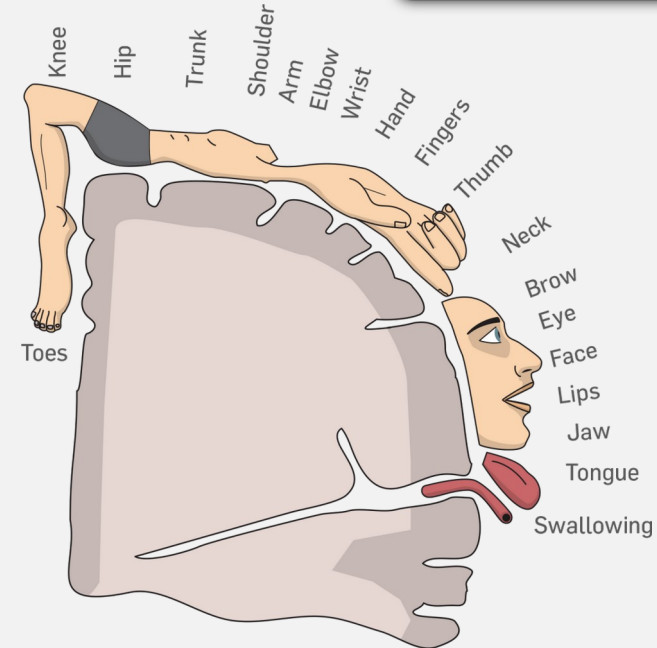
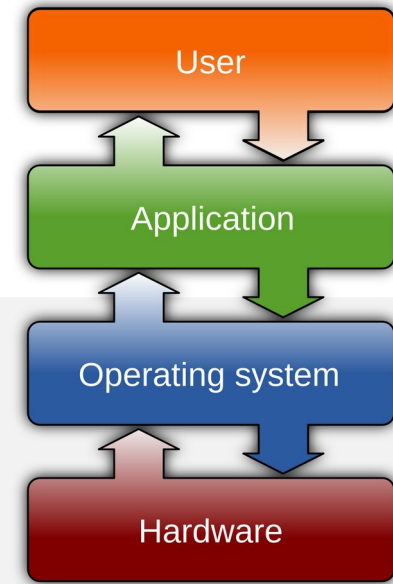
User : The human, or another program

The at the User level, we need an interface a human can use:

## Eyes, Hands

Eyes : written language → monitor, paper

Hands : tools to write → keyboard, mouse



We need a Language to write a Program which translates into Program

# Programming language

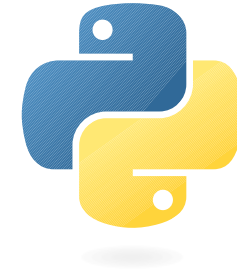
*A programming language is a system of notation for writing computer programs. Most programming languages are text-based formal languages. [...]*

*The description of a programming language is usually split into the two components of syntax (form) and semantics (meaning), which are usually defined by a formal language. Some languages are defined by a specification document (for example, the C programming language is specified by an ISO Standard) while other languages (such as Perl) have a dominant implementation that is treated as a reference. Some languages have both, with the basic language defined by a standard and extensions taken from the dominant implementation being common.*

*Programming language theory is the [field] of computer science that studies the design, implementation, analysis, characterization, and classification of programming languages.*

[https://en.wikipedia.org/wiki/Programming\\_language](https://en.wikipedia.org/wiki/Programming_language)

# Python Language



Python is an interpreted programming language with a commitment on code beauty

```
Python 3.11.0 (main, Oct 25 2022, 14:34:08) [GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import this
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
>>>
```

# The interpreter

The Python interpreter is a computer program that is able to translate syntactically correct python language instructions semantically describing a program execution into the sequence of byte code bit sequences that can be executed by the computer processor.

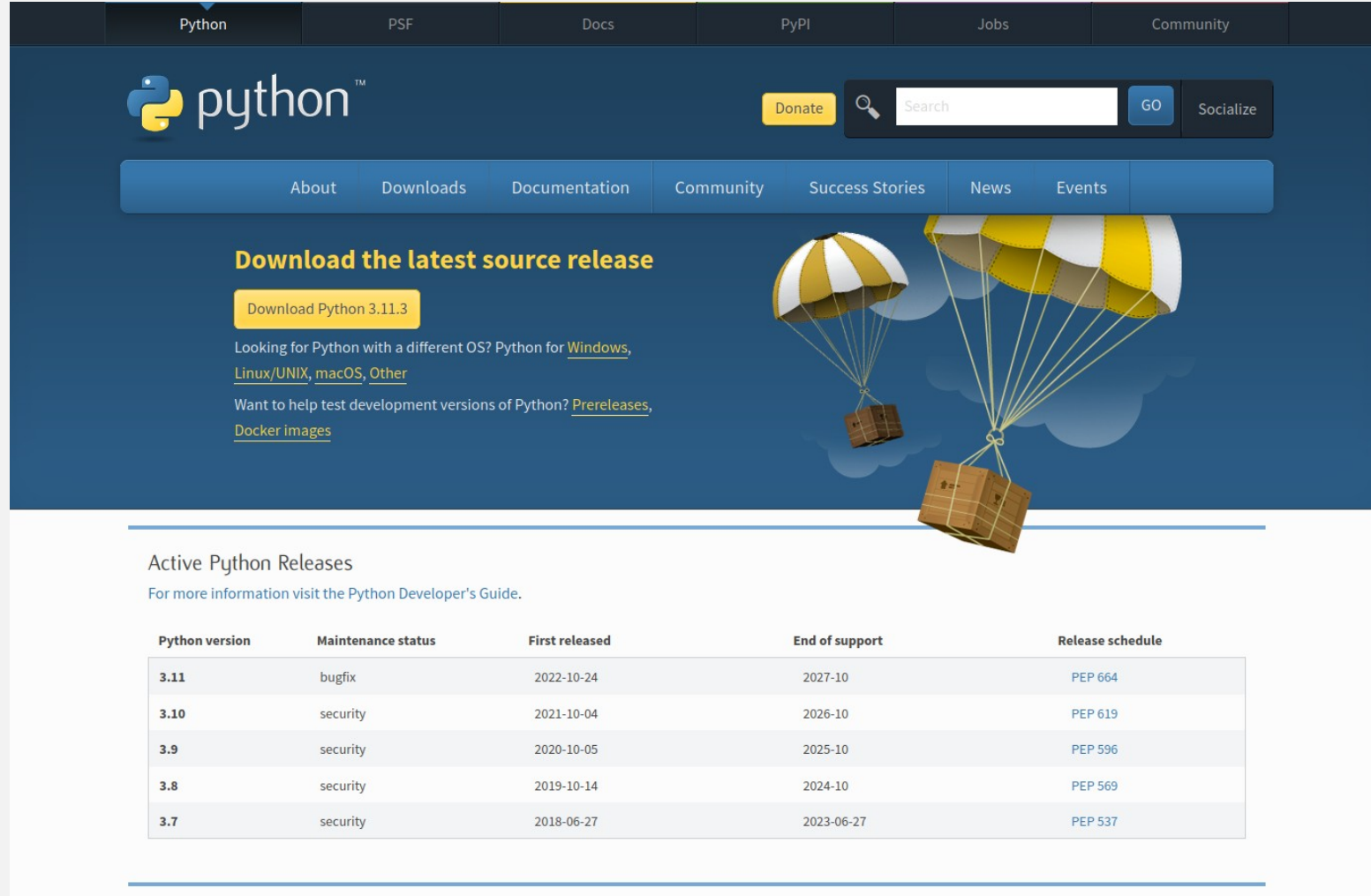
- Input phase : User types in the program, line by line or in a text file
- Interpretation phase : Python program reads the instruction, validates syntax
- Execution phase : The translated byte code is executed by the computer
- Completion phase : The program completes and the interpreter gives control to the OS

REPL : Read Evaluate Print Loop

```
Python 3.10.8 (main, Oct 25 2022, 15:21:47) [GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> a = 10
>>> b = 20
>>> a+b
30
>>> █
```

# Install the interpreter

<https://www.python.org/downloads>



The screenshot shows the Python.org website's download page. At the top, there is a navigation bar with links for Python, PSF, Docs, PyPI, Jobs, and Community. Below this is the Python logo, a search bar, and a 'Donate' button. A secondary navigation bar contains links for About, Downloads, Documentation, Community, Success Stories, News, and Events. The main content area features a heading 'Download the latest source release' and a prominent yellow button labeled 'Download Python 3.11.3'. Below the button, there are links for other operating systems: 'Looking for Python with a different OS? Python for [Windows](#), [Linux/UNIX](#), [macOS](#), [Other](#)'. Further down, there are links for 'Want to help test development versions of Python? [Prereleases](#), [Docker images](#)'. An illustration of two parachutes carrying crates is positioned to the right of the text. Below the main content, there is a section titled 'Active Python Releases' with a link to the 'Python Developer's Guide'. A table lists the active Python versions, their maintenance status, release dates, end of support dates, and release schedules.

## Download the latest source release

[Download Python 3.11.3](#)

Looking for Python with a different OS? Python for [Windows](#), [Linux/UNIX](#), [macOS](#), [Other](#)

Want to help test development versions of Python? [Prereleases](#), [Docker images](#)

### Active Python Releases

For more information visit the [Python Developer's Guide](#).

Python version	Maintenance status	First released	End of support	Release schedule
3.11	bugfix	2022-10-24	2027-10	PEP 664
3.10	security	2021-10-04	2026-10	PEP 619
3.9	security	2020-10-05	2025-10	PEP 596
3.8	security	2019-10-14	2024-10	PEP 569
3.7	security	2018-06-27	2023-06-27	PEP 537



# Install Science Python

<https://www.anaconda.com/download>

The screenshot shows the Anaconda website's download page. At the top left is the Anaconda logo. The navigation menu includes 'Enterprise', 'Pricing', 'Solutions', 'Resources', and 'About'. A 'Contact Sales' button is in the top right. The main heading is 'Anaconda Distribution' followed by 'Free Download'. Below this is the text 'Everything you need to get started in data science on your workstation.' A list of four features is provided, each with a green checkmark. At the bottom of the list are two buttons: 'Start Coding Now' and 'Download'. Below the buttons is the text 'Get Additional Installers' with icons for Windows, macOS, and Linux. At the bottom of the page, there are three icons: a purple flower-like icon, a purple group of people icon, and a purple shield icon. A chat bubble is visible in the bottom right corner with the text 'Hey! Welcome to Anaconda. I'm here to help. What are you looking for today?' and a small red notification icon with the number '1'.

ANACONDA

Enterprise Pricing Solutions Resources About

Contact Sales

Anaconda Distribution

## Free Download

Everything you need to get started in data science on your workstation.

- ✓ Free distribution install
- ✓ Thousands of the most fundamental DS, AI, and ML packages
- ✓ Manage packages and environments from desktop application
- ✓ Deploy across hardware and software platforms

[Start Coding Now](#) [Download](#)

Get Additional Installers

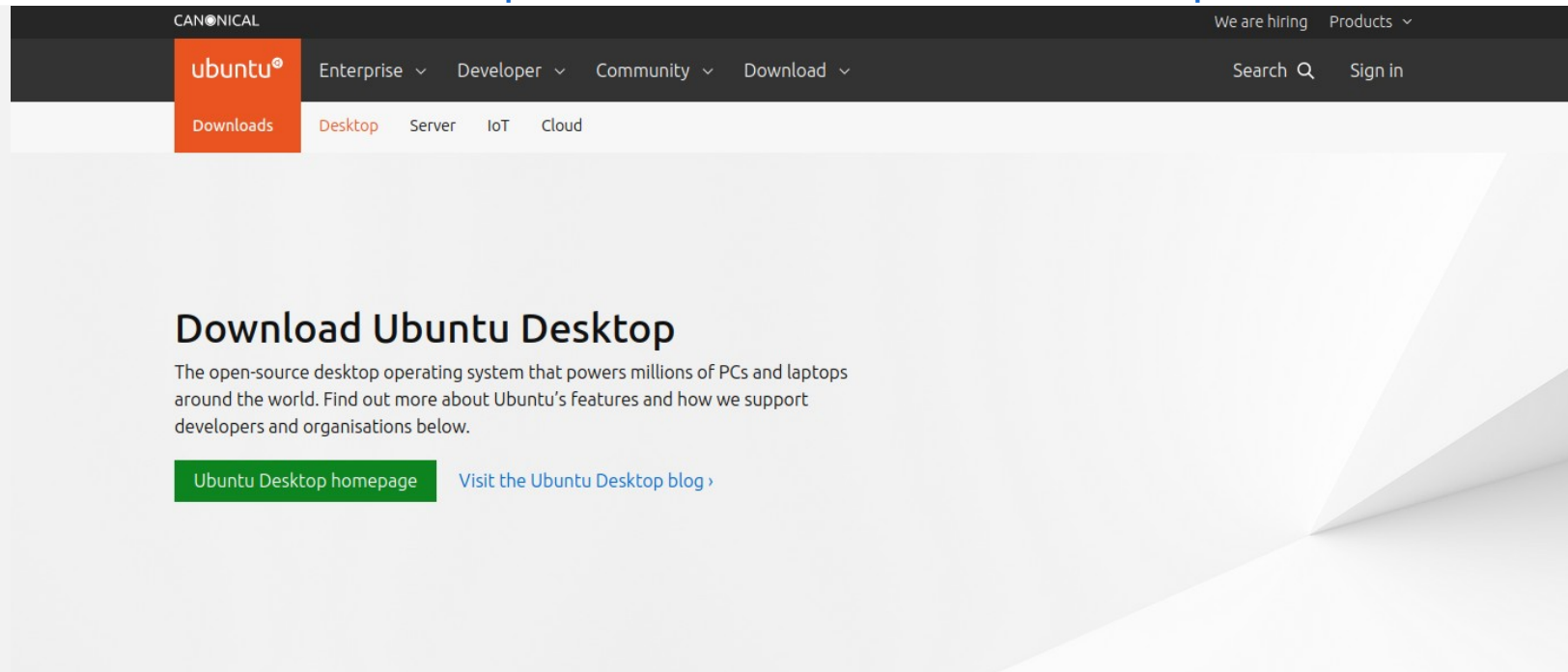
Windows macOS Linux

Open Source User-friendly Trusted

Hey! Welcome to Anaconda. I'm here to help. What are you looking for today?

# Install a full FREE OS

<https://ubuntu.com/download/desktop>



The screenshot shows the top navigation bar of the Ubuntu website. It includes the Canonical logo, the Ubuntu logo, and a menu with options: Enterprise, Developer, Community, and Download. There are also links for 'We are hiring', 'Products', 'Search', and 'Sign in'. Below the navigation bar, the 'Downloads' section is active, with sub-links for Desktop, Server, IoT, and Cloud. The main content area features a large heading 'Download Ubuntu Desktop' and a paragraph describing it as an open-source desktop operating system. Two buttons are present: a green 'Ubuntu Desktop homepage' button and a blue 'Visit the Ubuntu Desktop blog' link.

## Download Ubuntu Desktop

The open-source desktop operating system that powers millions of PCs and laptops around the world. Find out more about Ubuntu's features and how we support developers and organisations below.

[Ubuntu Desktop homepage](#) [Visit the Ubuntu Desktop blog](#)

## Ubuntu 22.04.2 LTS

The latest LTS version of Ubuntu, for desktop PCs and laptops. LTS stands for long-term support — which means five years of free security and maintenance updates, guaranteed until April 2027.

[Ubuntu 22.04 LTS release notes](#)

Recommended system requirements:

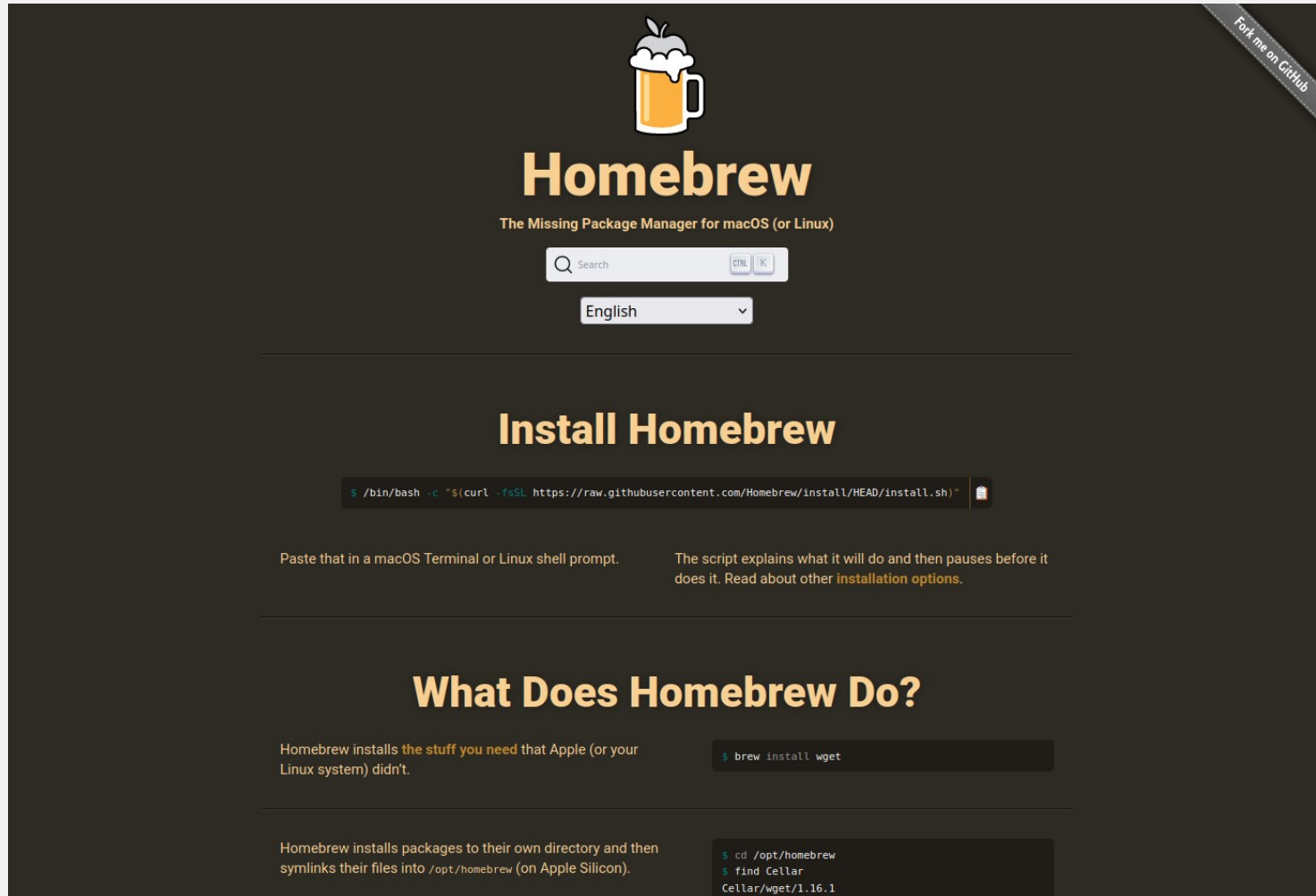
- ✓ 2 GHz dual-core processor or better
- ✓ 4 GB system memory
- ✓ Internet access is helpful
- ✓ Either a DVD drive or a USB port for the



[Download 22.04.2](#)

# Free Software on MacOS

<https://brew.sh>



The screenshot shows the Homebrew website homepage. At the top center is the Homebrew logo, a beer mug with a crown of foam. Below the logo is the word "Homebrew" in a large, bold, orange font. Underneath that is the tagline "The Missing Package Manager for macOS (or Linux)". There is a search bar with a magnifying glass icon and the text "Search", and a language dropdown menu currently set to "English". A diagonal banner in the top right corner says "Fork me on GitHub".

## Install Homebrew

```
$ /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Paste that in a macOS Terminal or Linux shell prompt. The script explains what it will do and then pauses before it does it. Read about other [installation options](#).

## What Does Homebrew Do?

Homebrew installs **the stuff you need** that Apple (or your Linux system) didn't.

```
$ brew install wget
```

Homebrew installs packages to their own directory and then symlinks their files into `/opt/homebrew` (on Apple Silicon).

```
$ cd /opt/homebrew
$ find Cellar
Cellar/wget/1.16.1
```

# WSL Windows Subsystem for Linux

The screenshot shows the Microsoft Learn website interface. At the top, there is a navigation bar with the Microsoft logo, 'Learn' link, and various categories like 'Documentation', 'Training', 'Certifications', etc. A search bar and 'Sign in' link are on the right. Below this is a secondary navigation bar with 'Windows' selected, and a 'Dashboard' button. The main content area features a breadcrumb trail: 'Learn / Windows / Development environment / WSL /'. The article title is 'Install Linux on Windows with WSL', dated 01/12/2023, with 8 contributors and a 'Feedback' link. The article text explains that developers can run both Windows and Linux on the same machine using WSL, allowing for the installation of Linux distributions like Ubuntu, OpenSUSE, Kali, Debian, and Arch Linux. It notes that WSL lets users use Linux applications and tools directly on Windows without the overhead of a virtual machine. The article is divided into sections: 'Prerequisites', 'Install WSL command', and 'In this article'. The 'Prerequisites' section states that Windows 10 version 2004 or higher (Build 19041 or higher) or Windows 11 is required. The 'Install WSL command' section provides a PowerShell command to install WSL: `wsl --install`. A 'Download PDF' link is available in the left sidebar. The 'In this article' sidebar lists links for 'Prerequisites', 'Install WSL command', 'Change the default Linux distribution installed', and 'Set up your Linux user info', with a 'Show more' dropdown.

Microsoft | Learn | Documentation | Training | Certifications | Q&A | Code Samples | Assessments | Shows | Events

Windows | Release health | Windows client documentation | Application developers | Hardware developers | Windows Server | Sysinternals | **Dashboard**

Filter by title

WSL Documentation

- Overview
- Install
  - Install WSL**
  - Manual install steps for older versions
  - Install on Windows Server
- Tutorials
- Concepts
- How-to
- Frequently Asked Questions
- Troubleshooting
- Release Notes

[Download PDF](#)

Learn / Windows / Development environment / WSL /

## Install Linux on Windows with WSL

Article • 01/12/2023 • 8 contributors [Feedback](#)

Developers can access the power of both Windows and Linux at the same time on a Windows machine. The Windows Subsystem for Linux (WSL) lets developers install a Linux distribution (such as Ubuntu, OpenSUSE, Kali, Debian, Arch Linux, etc) and use Linux applications, utilities, and Bash command-line tools directly on Windows, unmodified, without the overhead of a traditional virtual machine or dualboot setup.

### Prerequisites

You must be running Windows 10 version 2004 and higher (Build 19041 and higher) or Windows 11 to use the commands below. If you are on earlier versions please see [the manual install page](#).

### Install WSL command

You can now install everything you need to run WSL with a single command. Open PowerShell or Windows Command Prompt in **administrator** mode by right-clicking and selecting "Run as administrator", enter the `wsl --install` command, then restart your machine.

```
PowerShell Copy  
  
wsl --install
```

This command will enable the features necessary to run WSL and install the Ubuntu distribution of Linux. ([This default distribution can be changed](#)).

#### In this article

- Prerequisites
- Install WSL command
- Change the default Linux distribution installed
- Set up your Linux user info

[Show more](#)

# ICTP cloud desktop

Every participant has access to a cloud based Linux desktop.

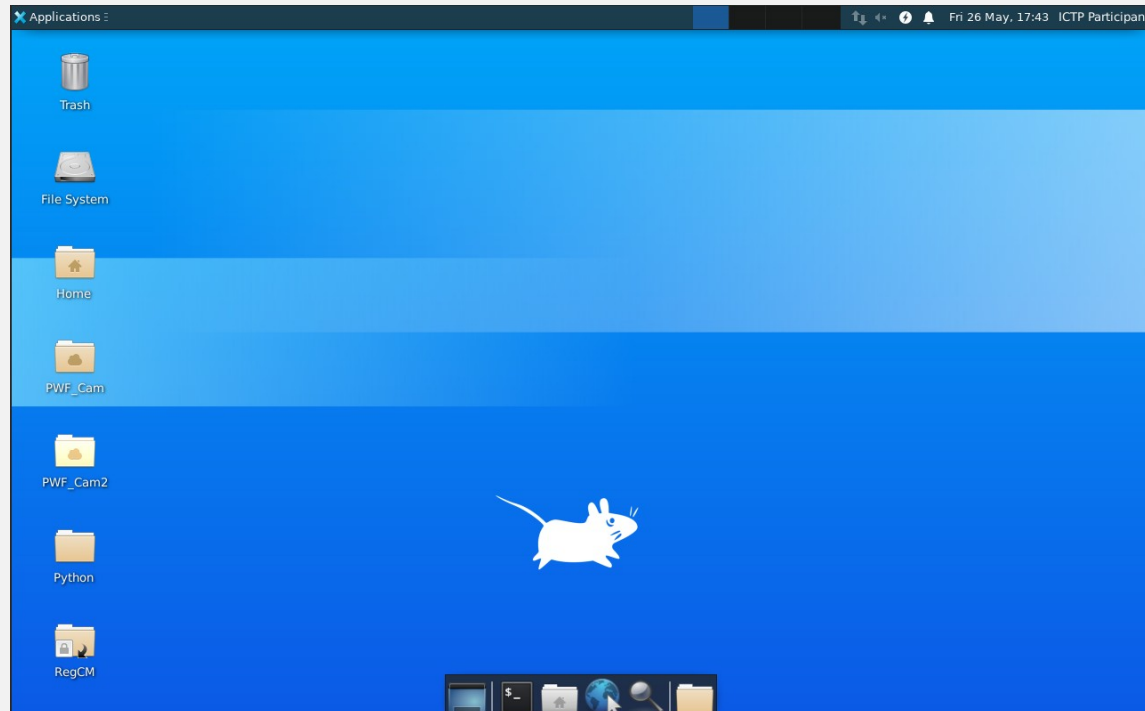
Who has NOT received the ICTP e-mail ?

Who has NOT managed to complete the key setup ?

NOW:

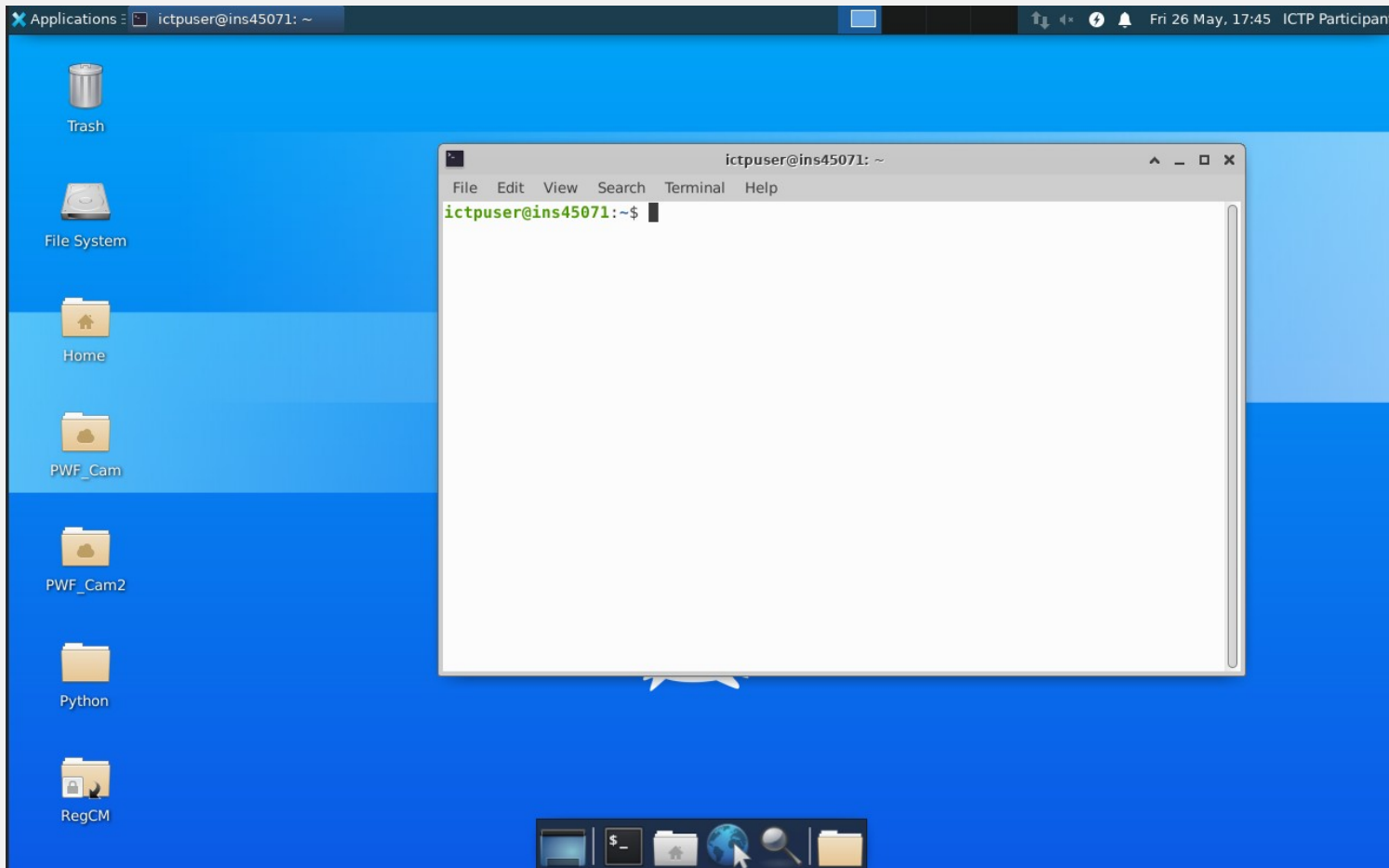
- 1) Can you connect to the system ?
- 2) Who has NO idea how to proceed ?

Open a terminal !



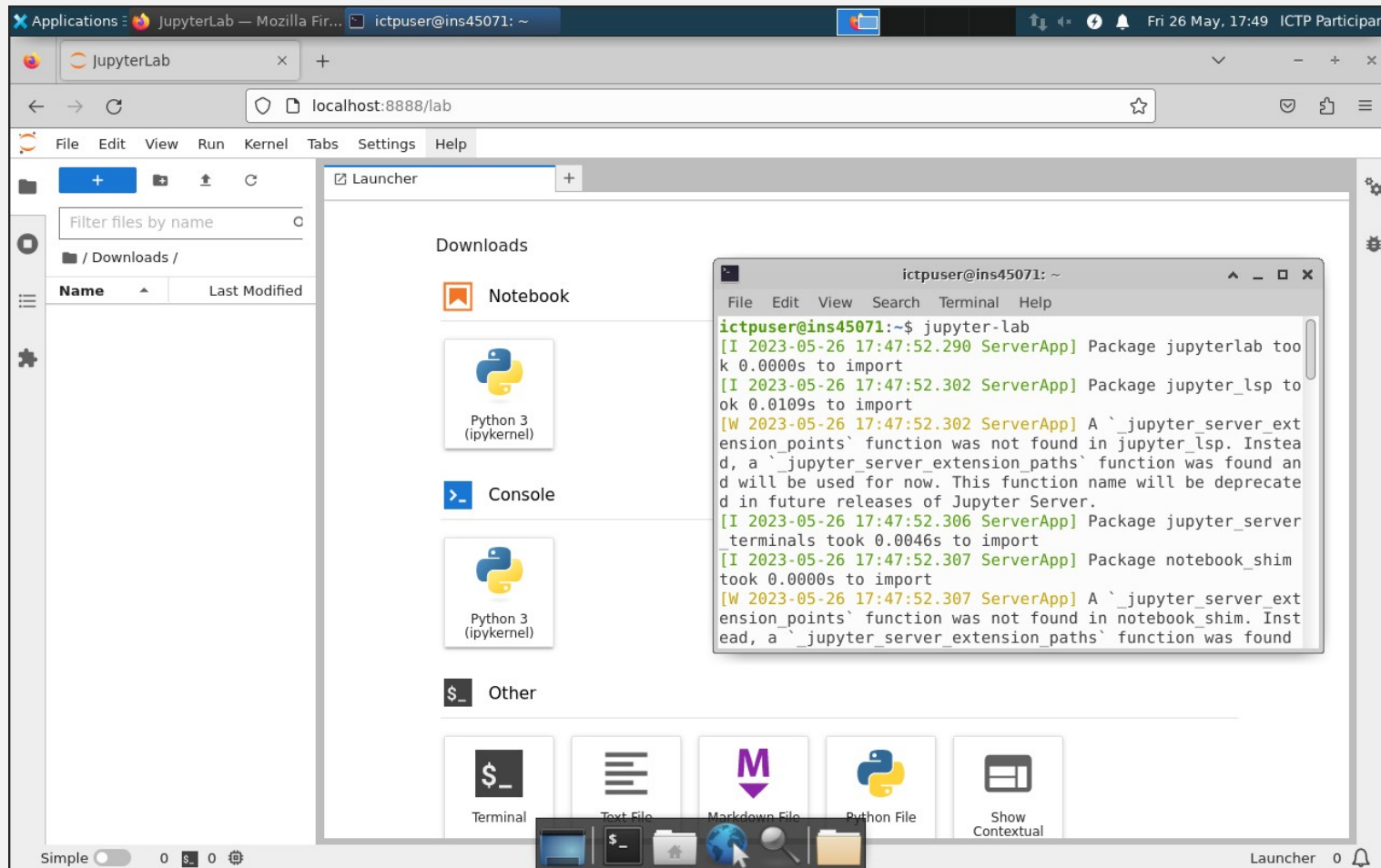
# Linux Terminal

WHO HAS NOT IDEA WHAT THIS IS ???



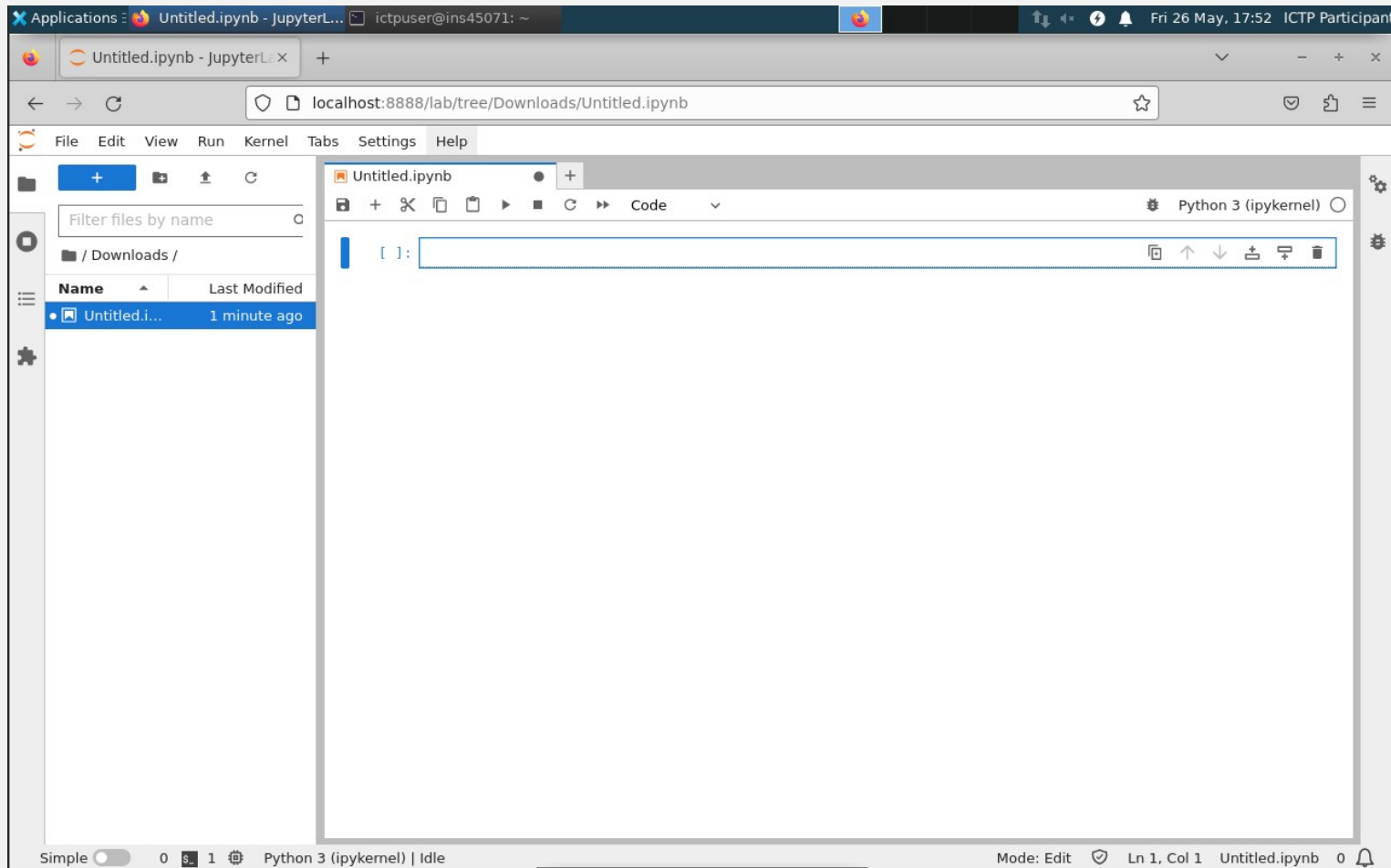
# Jupyter Terminal : Start

Type in **jupyter-lab**, switch to the browser and open the Python3 notebook



# Jupyter Terminal

Anyone here is ready to go!

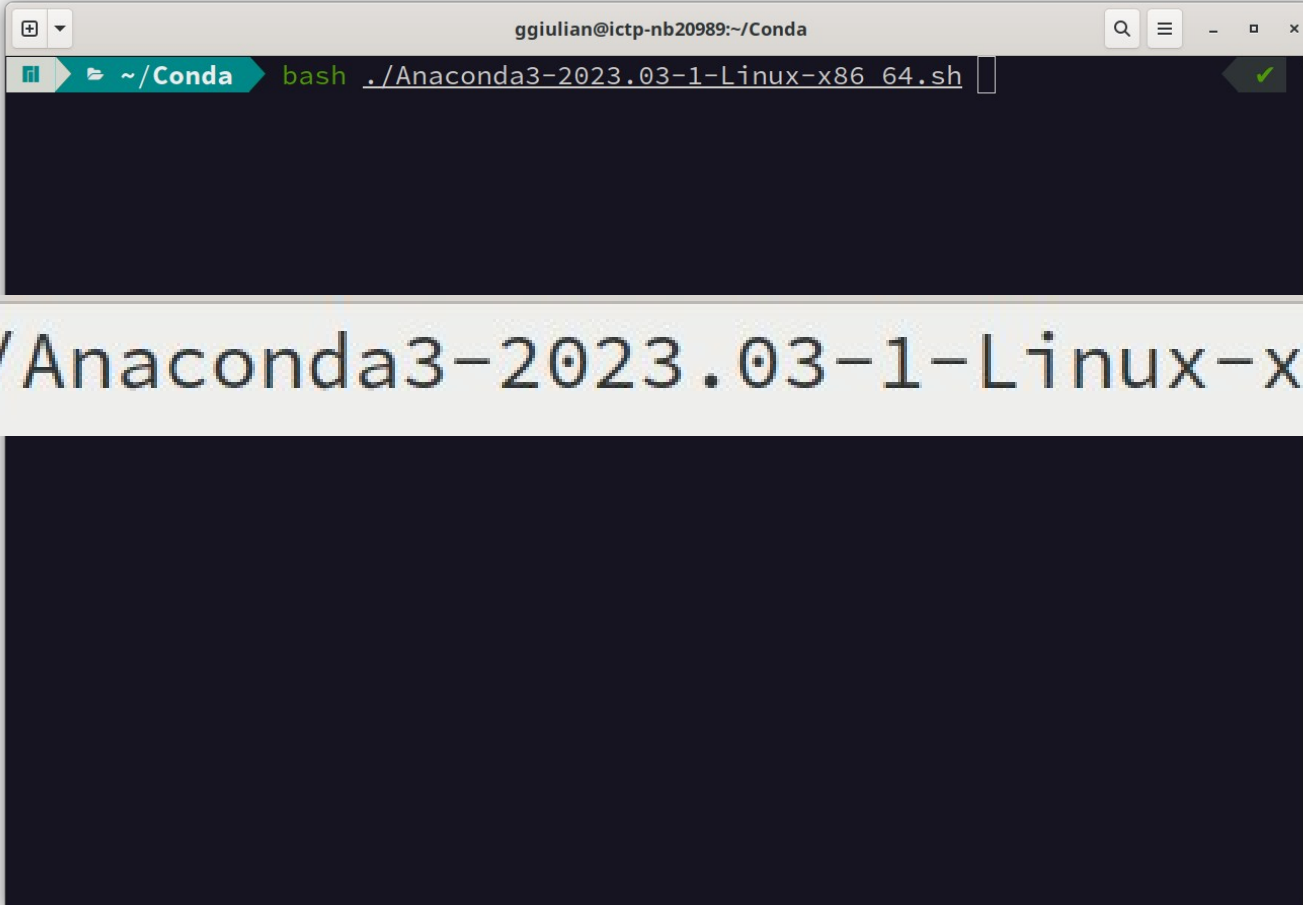




# Anaconda installer

Run the installer available for your OS

On Linux :

A terminal window with a dark background. The title bar shows the user 'ggiulian@ictp-nb20989' and the directory '~/Conda'. The prompt is 'bash' and the command being executed is './Anaconda3-2023.03-1-Linux-x86\_64.sh'. A green checkmark is visible in the top right corner of the terminal window.

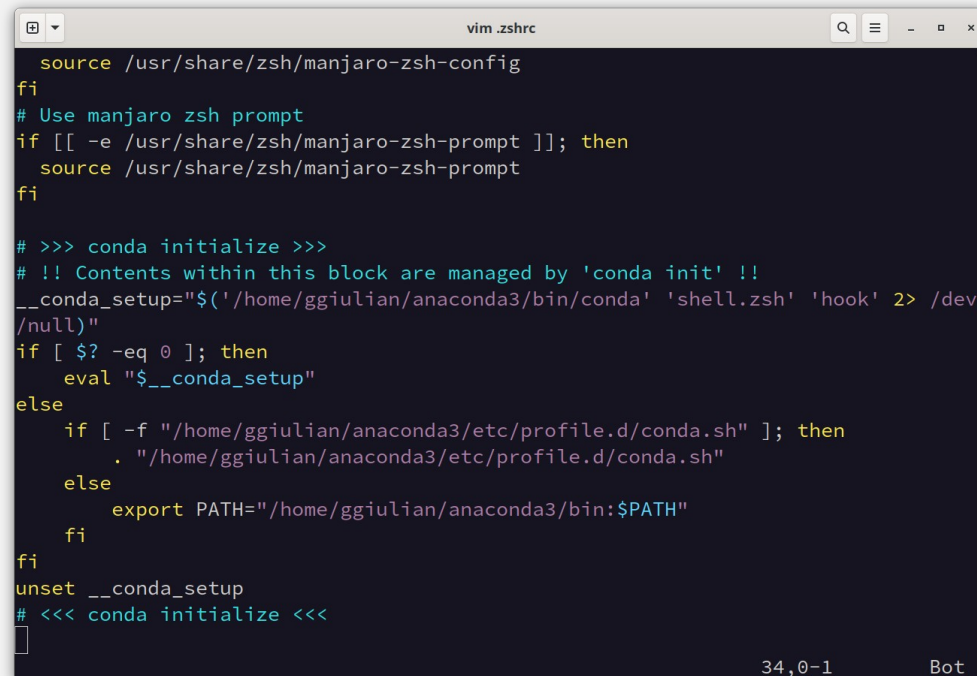
```
ggiulian@ictp-nb20989:~/Conda
bash ./Anaconda3-2023.03-1-Linux-x86_64.sh
```

```
bash ./Anaconda3-2023.03-1-Linux-x86_64.sh
```

# Activation

On Linux the conda installer changes the shell initialization file. Open a new terminal to activate.

On my zsh :



```
vim .zshrc
source /usr/share/zsh/manjaro-zsh-config
fi
# Use manjaro zsh prompt
if [[ -e /usr/share/zsh/manjaro-zsh-prompt ]]; then
  source /usr/share/zsh/manjaro-zsh-prompt
fi

# >>> conda initialize >>>
# !! Contents within this block are managed by 'conda init' !!
__conda_setup="$(('/home/ggiulian/anaconda3/bin/conda' 'shell.zsh' 'hook' 2> /dev
/null)"
if [ $? -eq 0 ]; then
  eval "$__conda_setup"
else
  if [ -f "/home/ggiulian/anaconda3/etc/profile.d/conda.sh" ]; then
    . "/home/ggiulian/anaconda3/etc/profile.d/conda.sh"
  else
    export PATH="/home/ggiulian/anaconda3/bin:$PATH"
  fi
fi
unset __conda_setup
# <<< conda initialize <<<
[]
34,0-1 Bot
```

To remove conda, comment or delete the lines between the “conda initialize” and to get back the storage conda is using remove the anaconda3 directory (in my case it is in /home/ggiulian/)

# Environment

Create a new environment for the PWF activity

```
~ | conda env list | 1 x base
# conda environments:
#
base          * /home/ggiulian/anaconda3

~ | conda create --name pwf | 1 x base
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: /home/ggiulian/anaconda3/envs/pwf

Proceed ([y]/n)? yes
```

```
~ | conda install -c conda-forge jupyterlab xarray cartopy regionmask
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: /home/ggiulian/anaconda3/envs/pwf
[...]
Proceed ([y]/n)? y

Downloading and Extracting Packages

Preparing transaction: done
Verifying transaction: done
Executing transaction: done

~ | jupyter-lab | 2m 56s pwf
```

```
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#   $ conda activate pwf
#
# To deactivate this environment, use
#
#   $ conda deactivate

~ | conda activate pwf | 7s base
~ | conda list | pwf
# packages in environment at /home/ggiulian/anaconda3/envs/pwf:
#
# Name                Version      Build      Channel
```

```
conda create --name pwf
conda activate pwf
conda install -c conda-forge jupyterlab xarray cartopy regionmask
jupyter-lab
```

# Jupyter on the web

Jupyter “free” servers

<https://www.anaconda.com/code-in-the-cloud>

<https://colab.research.google.com/>



# USB key

Contains:

- 1) Python releases 3.11.3
    - 1) Source file for Linux
    - 2) Windows 64 bit interpreter
    - 3) MacOS universal interpreter
  - 2) Anaconda 2023.05
    - 1) Linux x86\_64
    - 2) Windows 64 bit installer
    - 3) MacOS installers
- This document in ODP (Open Document Presentation) and PDF.
  - Jupyter Notebooks material
  - RegCM model tutorial presentations
  -

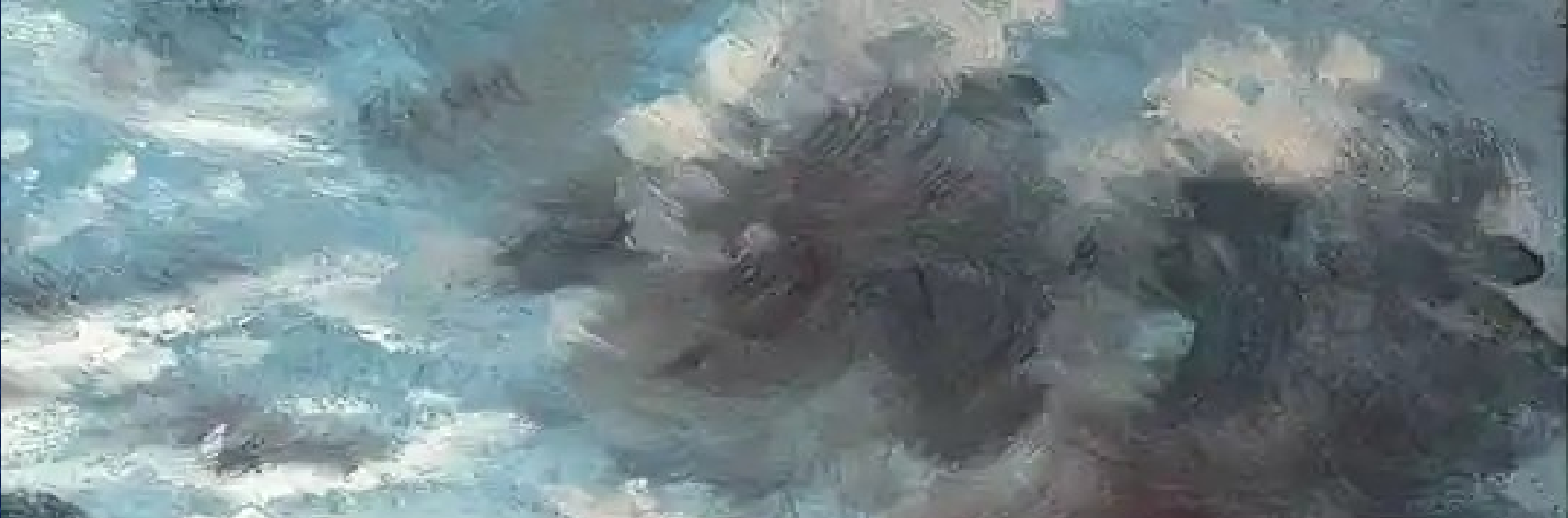


# Ready?

## Open the Lesson 1 Notebook :

Lesson1.ipynb





Let's Start !



# Hello Python!

A crash course introduction to  
the computer, programming,  
and the Python programming  
language

ICTP – ESP – Graziano Giuliani <ggiulian@ictp.it>

27/05/23

PWF Cameroon  
School on Climate Science  
5-9 June 2023  
University of Dschang



The Abdus Salam  
International Centre  
for Theoretical Physics

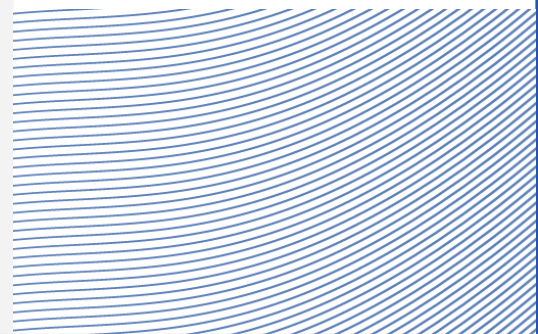


IAEA



# The Computer

- A machine is a physical system using power to apply forces and control movement to perform an action.
- A digital bit is the minimal amount of information representing a logical state of a system with two possible values [0-1]
- A computer is an electronic digital machine carrying out logical operations on binary bits of information
- Switching a bit is an action and work must be done on the system to change it



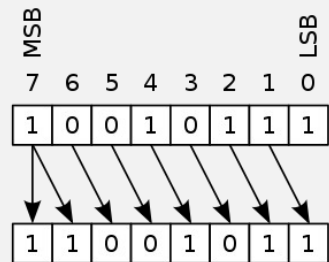
# Computer Program



- A computer program is a sequence of instructions modifying a string of bits representing the status of a system. The initial status is the input of the program, the final state is the output of the program
- The instructions are stored in bit patterns themselves, following conventions defined by the designer of a piece of hardware at the heart of the computer called the **Processor**.
- The program and the input data are sequences of bits registered in magnetic **Memory** on the computer. The output of the program is registered in the memory, either updating the initial system status or creating a new status.

Where is the magic?

- The information is CODED : a → 01100001
- There are codes for
  - Characters
  - Exact Integer numbers
  - Approximate representation of Real numbers
- Special processing units in the processor can perform
  - Character string manipulation
  - Arithmetic operation
  - Mathematical complex evaluation



# Write a program

Hardware : the metal box in which the “magic” happens  
Software : the instructions describing how to go from input to output state

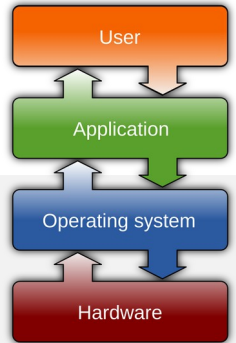
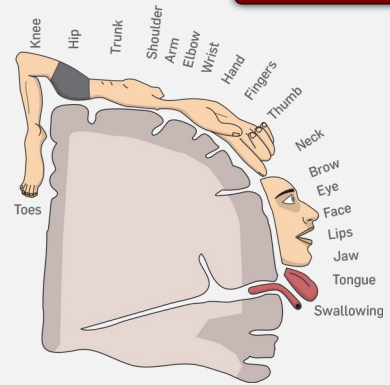
Operating System : The software part directly controlling the Hardware circuits  
Application : The program you use

User : The human, or another program

The at the User level, we need an interface a human can use:

## Eyes, Hands

Eyes : written language → monitor, paper  
Hands : tools to write → keyboard, mouse



We need a Language to write a Program which translates into Program

# Programming language

*A programming language is a system of notation for writing computer programs. Most programming languages are text-based formal languages. [...]*

*The description of a programming language is usually split into the two components of syntax (form) and semantics (meaning), which are usually defined by a formal language. Some languages are defined by a specification document (for example, the C programming language is specified by an ISO Standard) while other languages (such as Perl) have a dominant implementation that is treated as a reference. Some languages have both, with the basic language defined by a standard and extensions taken from the dominant implementation being common.*

*Programming language theory is the [field] of computer science that studies the design, implementation, analysis, characterization, and classification of programming languages.*

[https://en.wikipedia.org/wiki/Programming\\_language](https://en.wikipedia.org/wiki/Programming_language)

# Python Language



Python is an interpreted programming language with a commitment on code beauty

```
Python 3.11.0 (main, Oct 25 2022, 14:34:08) [GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import this
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
>>>
```

# The interpreter

The Python interpreter is a computer program that is able to translate syntactically correct python language instructions semantically describing a program execution into the sequence of byte code bit sequences that can be executed by the computer processor.

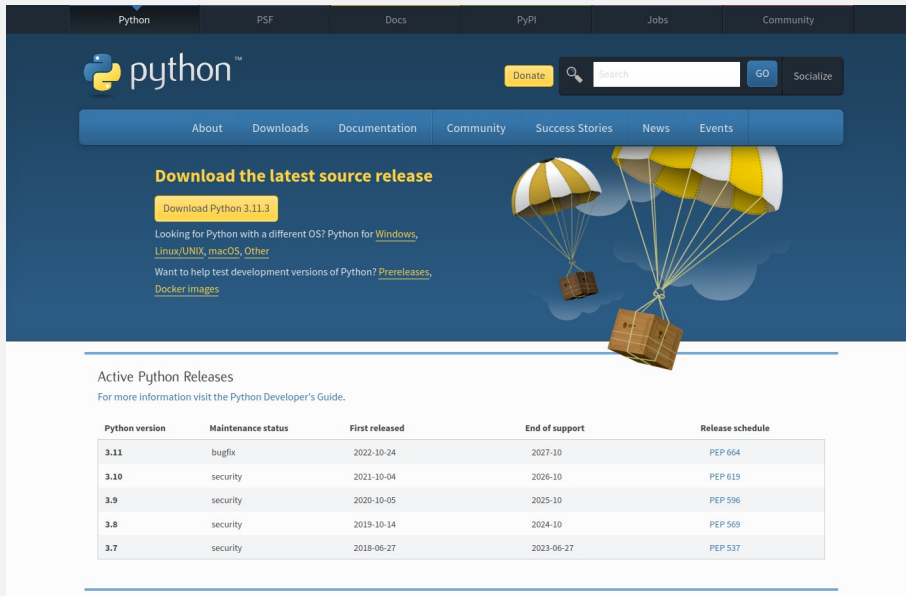
Input phase	: User types in the program, line by line or in a text file
Interpretation phase	: Python program reads the instruction, validates syntax
Execution phase	: The translated byte code is executed by the computer
Completion phase	: The program completes and the interpreter gives control to the OS

REPL : Read Evaluate Print Loop

```
Python 3.10.8 (main, Oct 25 2022, 15:21:47) [GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> a = 10
>>> b = 20
>>> a+b
30
>>> █
```

# Install the interpreter

<https://www.python.org/downloads>



The screenshot shows the Python.org website's download page. At the top, there is a navigation bar with links for Python, PSF, Docs, PyPI, Jobs, and Community. Below this is the Python logo, a search bar, and a 'Donate' button. A secondary navigation bar contains links for About, Downloads, Documentation, Community, Success Stories, News, and Events. The main content area features a large heading 'Download the latest source release' and a prominent yellow button labeled 'Download Python 3.11.3'. Below the button, there are links for different operating systems: Windows, Linux/UNIX, macOS, and Other. There are also links for prereleases and Docker images. An illustration of two parachutes with cargo boxes is positioned to the right of the text. Below this section, there is a heading 'Active Python Releases' with a link to the Python Developer's Guide. A table follows, detailing the release schedule for various Python versions.

Python version	Maintenance status	First released	End of support	Release schedule
3.11	bugfix	2022-10-24	2027-10	PEP 664
3.10	security	2021-10-04	2026-10	PEP 619
3.9	security	2020-10-05	2025-10	PEP 596
3.8	security	2019-10-14	2024-10	PEP 569
3.7	security	2018-06-27	2023-06-27	PEP 537

# Install Science Python

<https://www.anaconda.com/download>

The screenshot shows the Anaconda website's 'Free Download' page. The header includes the Anaconda logo and navigation links for Enterprise, Pricing, Solutions, Resources, and About, along with a 'Contact Sales' button. The main content area features the heading 'Anaconda Distribution' and 'Free Download', followed by the text 'Everything you need to get started in data science on your workstation.' Below this is a list of four bullet points: 'Free distribution install', 'Thousands of the most fundamental DS, AI, and ML packages', 'Manage packages and environments from desktop application', and 'Deploy across hardware and software platforms'. Two buttons, 'Start Coding Now' and 'Download', are prominently displayed. Below the buttons, there is a link to 'Get Additional Installers' with icons for Windows, macOS, and Linux. The footer of the page includes icons for 'Open Source', 'User friendly', and 'Trusted', and a chatbot window with the text: 'Hey! 🐍 Welcome to Anaconda. I'm here to help. What are you looking for today?'.

ANACONDA. Enterprise Pricing Solutions Resources About Contact Sales

Anaconda Distribution

## Free Download

Everything you need to get started in data science on your workstation.

- ✓ Free distribution install
- ✓ Thousands of the most fundamental DS, AI, and ML packages
- ✓ Manage packages and environments from desktop application
- ✓ Deploy across hardware and software platforms

[Start Coding Now](#) [Download](#)

Get Additional Installers

Windows macOS Linux

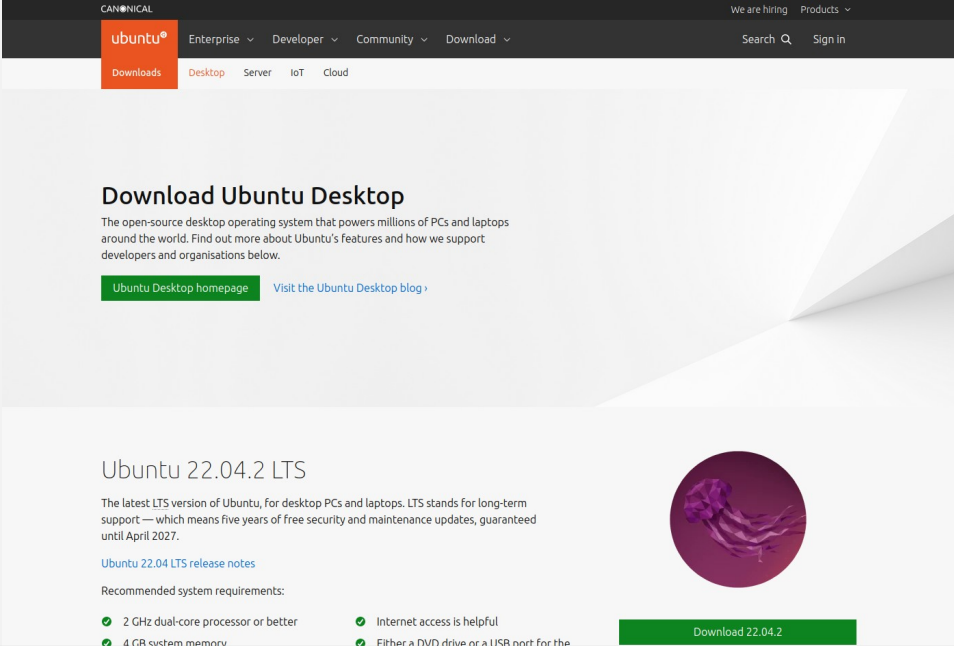
Open Source User friendly Trusted

Hey! 🐍 Welcome to Anaconda. I'm here to help. What are you looking for today?



# Install a full FREE OS

<https://ubuntu.com/download/desktop>



The screenshot shows the Ubuntu website's download page for the desktop version. At the top, there is a navigation bar with the Ubuntu logo and links for Enterprise, Developer, Community, and Download. Below this, a secondary navigation bar lists Downloads, Desktop, Server, IoT, and Cloud. The main content area features a large heading 'Download Ubuntu Desktop' followed by a paragraph describing it as an open-source desktop operating system. There are two links: 'Ubuntu Desktop homepage' and 'Visit the Ubuntu Desktop blog'. Below this, the section for 'Ubuntu 22.04.2 LTS' is visible, including a description of LTS support, a link to 'Ubuntu 22.04 LTS release notes', and a list of recommended system requirements. A circular image of a purple dragon is on the right, and a green 'Download 22.04.2' button is at the bottom right.

CANONICAL We are hiring Products ▾

ubuntu Enterprise ▾ Developer ▾ Community ▾ Download ▾ Search 🔍 Sign in

Downloads Desktop Server IoT Cloud

## Download Ubuntu Desktop

The open-source desktop operating system that powers millions of PCs and laptops around the world. Find out more about Ubuntu's features and how we support developers and organisations below.

[Ubuntu Desktop homepage](#) [Visit the Ubuntu Desktop blog](#)

### Ubuntu 22.04.2 LTS

The latest LTS version of Ubuntu, for desktop PCs and laptops. LTS stands for long-term support — which means five years of free security and maintenance updates, guaranteed until April 2027.

[Ubuntu 22.04 LTS release notes](#)

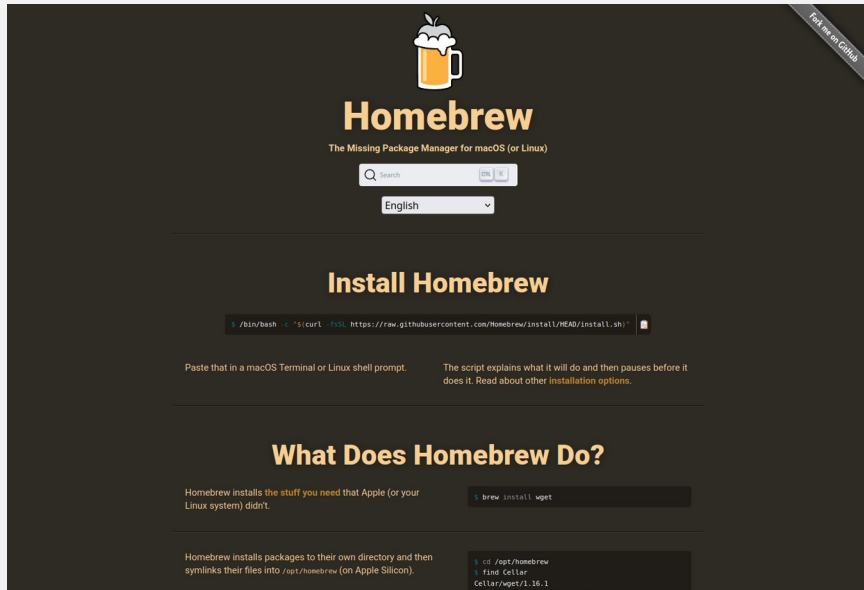
Recommended system requirements:

- ✔ 2 GHz dual-core processor or better
- ✔ Internet access is helpful
- ✔ 4 GB system memory
- ✔ Either a DVD drive or a USB port for the

[Download 22.04.2](#)

# Free Software on MacOS

<https://brew.sh>



The screenshot shows the Homebrew website with a dark theme. At the top, there is a logo of a beer mug with the text "Homebrew" and "The Missing Package Manager for macOS (or Linux)". Below the logo is a search bar and a language dropdown menu set to "English". The main heading is "Install Homebrew", followed by a terminal command: `! /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"`. Below the command, there are two columns of text: "Paste that in a macOS Terminal or Linux shell prompt." and "The script explains what it will do and then pauses before it does it. Read about other installation options." The next section is "What Does Homebrew Do?", which explains that Homebrew installs the stuff you need that Apple (or your Linux system) didn't. Below this, there are two columns of terminal output: `! brew install wget` and `! cd /opt/homebrew; find Cellar; Cellar/wget/1.16.1`. A small "Not on GitHub" badge is visible in the top right corner of the screenshot.

# WSL Windows Subsystem for Linux

The screenshot shows a Microsoft Learn article page. At the top, there is a navigation bar with 'Microsoft | Learn | Documentation' and various links like 'Training', 'Certifications', 'Q&A', 'Code Samples', 'Assessments', 'Shows', and 'Events'. A search bar and 'Sign in' link are on the right. Below the navigation, there are breadcrumb links: 'Windows > Release health > Windows client documentation > Application developers > Hardware developers > Windows Server > Sysinternals'. A 'Dashboard' button is in the top right. The main content area has a title 'Install Linux on Windows with WSL' and a sub-header 'Learn / Windows / Development environment / WSL /'. Below the title, it says 'Article • 01/12/2023 • 8 contributors' and a 'Feedback' link. The article text explains that developers can run both Windows and Linux on the same machine using WSL. It lists prerequisites: Windows 10 version 2004 or higher (Build 19041 or higher) or Windows 11. The 'Install WSL command' section shows a PowerShell terminal snippet: `wsl --install`. A 'Copy' button is next to the code. Below the code, it states: 'This command will enable the features necessary to run WSL and install the Ubuntu distribution of Linux. (This default distribution can be changed)'. On the left, there is a sidebar with a search box and a list of navigation items: 'WSL Documentation', 'Overview', 'Install', 'Install WSL', 'Manual install steps for older versions', 'Install on Windows Server', 'Tutorials', 'Concepts', 'How-to', 'Frequently Asked Questions', 'Troubleshooting', 'Release Notes', and 'Download PDF'. On the right, there is an 'In this article' section with links for 'Prerequisites', 'Install WSL command', 'Change the default Linux distribution installed', and 'Set up your Linux user info', along with a 'Show more' button.

# ICTP cloud desktop

Every participant has access to a cloud based Linux desktop.

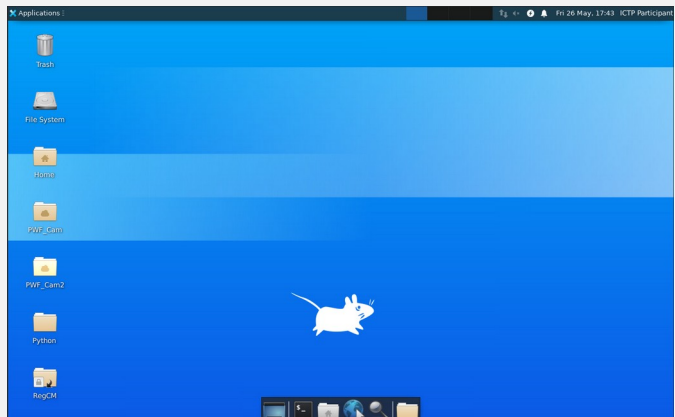
Who has NOT received the ICTP e-mail ?

Who has NOT managed to complete the key setup ?

NOW:

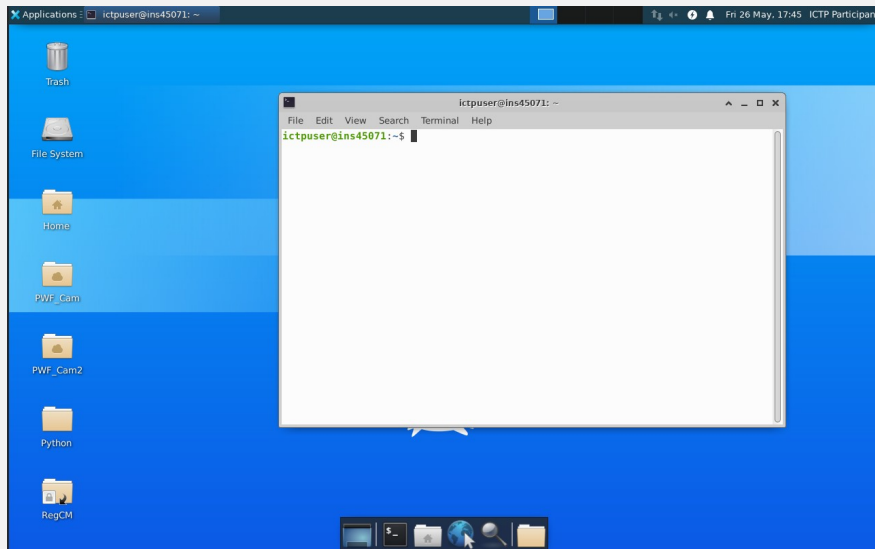
- 1) Can you connect to the system ?
- 2) Who has NO idea how to proceed ?

Open a terminal !



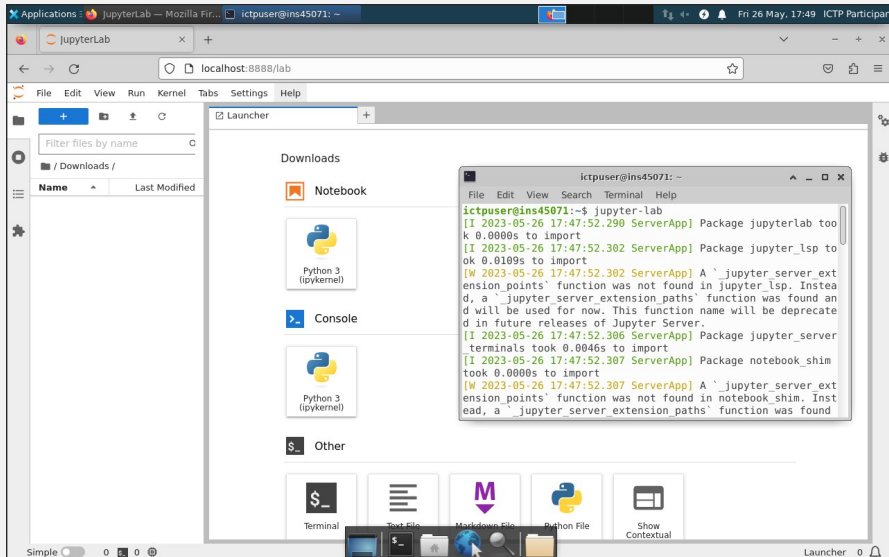
# Linux Terminal

WHO HAS NOT IDEA WHAT THIS IS ???



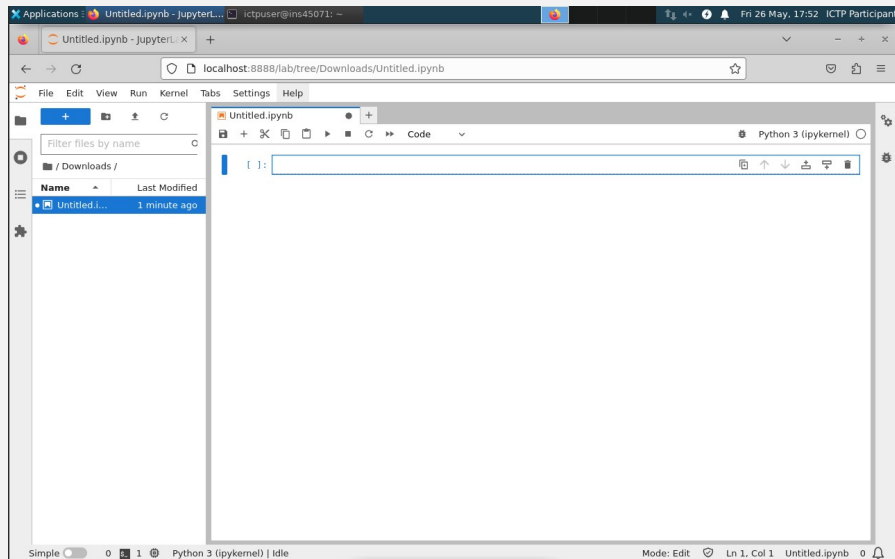
# Jupyter Terminal : Start

Type in `jupyter-lab`, switch to the browser and open the Python3 notebook



# Jupyter Terminal

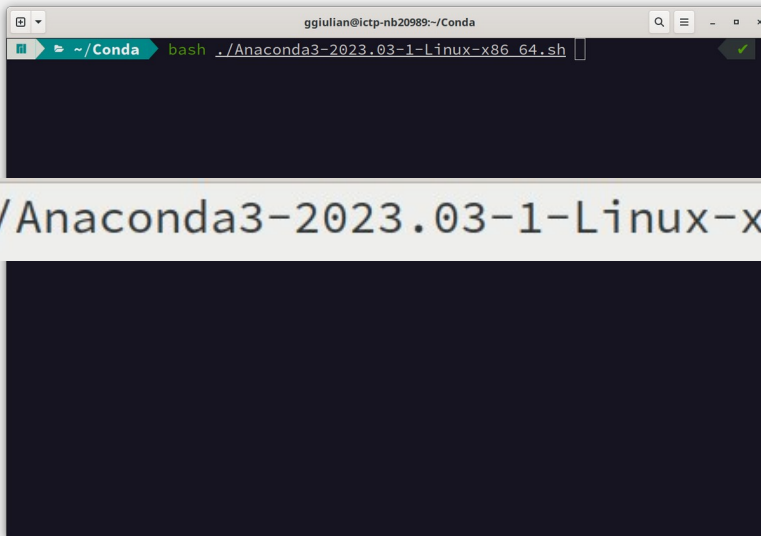
Anyone here is ready to go!



# Anaconda installer

Run the installer available for your OS

On Linux :

A terminal window with a dark background. The title bar reads 'ggiulian@ictp-nb20989:~/Conda'. The prompt is '~ /Conda' and the command being executed is 'bash ./Anaconda3-2023.03-1-Linux-x86\_64.sh'. A green checkmark is visible in the top right corner of the terminal window.

```
ggiulian@ictp-nb20989:~/Conda
~/Conda bash ./Anaconda3-2023.03-1-Linux-x86_64.sh
```

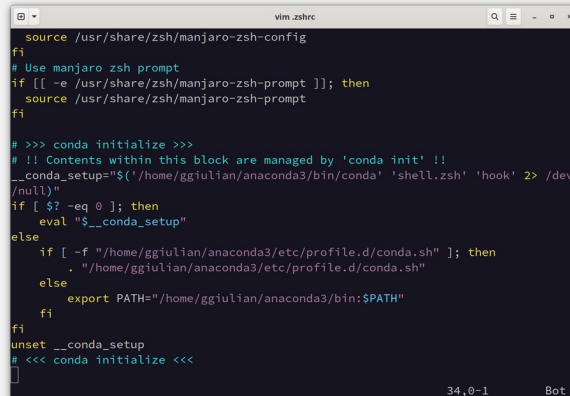
```
bash ./Anaconda3-2023.03-1-Linux-x86_64.sh
```



# Activation

On Linux the conda installer changes the shell initialization file. Open a new terminal to activate.

On my zsh :



```
vim_zshrc
source /usr/share/zsh/manjaro-zsh-config
fi
# Use manjaro zsh prompt
if [[ -e /usr/share/zsh/manjaro-zsh-prompt ]]; then
    source /usr/share/zsh/manjaro-zsh-prompt
fi

# >>> conda initialize >>>
# !! Contents within this block are managed by 'conda init' !!
__conda_setup="$('/home/ggiulian/anaconda3/bin/conda' 'shell.zsh' 'hook' 2> /dev
/null)"
if [ $? -eq 0 ]; then
    eval "$__conda_setup"
else
    if [ -f "/home/ggiulian/anaconda3/etc/profile.d/conda.sh" ]; then
        . "/home/ggiulian/anaconda3/etc/profile.d/conda.sh"
    else
        export PATH="/home/ggiulian/anaconda3/bin:$PATH"
    fi
fi
unset __conda_setup
# <<< conda initialize <<<
]
34,0-1 Bot
```

To remove conda, comment or delete the lines between the “conda initialize” and to get back the storage conda is using remove the anaconda3 directory (in my case it is in /home/ggiulian/)

# Environment

Create a new environment for the PWF activity

```
❯ ~ ▶ conda env list
# conda environments:
#
base                * /home/ggiulian/anaconda3

❯ ~ ▶ conda create --name pwf
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: /home/ggiulian/anaconda3/envs/pwf

Proceed ([y]/n)? yes
```

```
❯ ~ ▶ conda install -c conda-forge jupyterlab xarray cartopy regionmask
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: /home/ggiulian/anaconda3/envs/pwf
  [...]
Proceed ([y]/n)? y

Downloading and Extracting Packages

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
❯ ~ ▶ jupyter-lab
```

```
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
# To activate this environment, use
#
# $ conda activate pwf
#
# To deactivate this environment, use
#
# $ conda deactivate

❯ ~ ▶ conda activate pwf
❯ ~ ▶ conda list
```

```
conda create --name pwf
conda activate pwf
conda install -c conda-forge jupyterlab xarray cartopy regionmask
jupyter-lab
```

```
# packages in environment at /home/ggiulian/anaconda3/envs/pwf:
# Name                Version                Build                Channel
```

# Jupyter on the web

Jupyter “free” servers

<https://www.anaconda.com/code-in-the-cloud>

<https://colab.research.google.com/>



**ANACONDA**®



# USB key

Contains:

- 1) Python releases 3.11.3
    - 1) Source file for Linux
    - 2) Windows 64 bit interpreter
    - 3) MacOS universal interpreter
  - 2) Anaconda 2023.05
    - 1) Linux x86\_64
    - 2) Windows 64 bit installer
    - 3) MacOS installers
- This document in ODP (Open Document Presentation) and PDF.
  - Jupyter Notebooks material
  - RegCM model tutorial presentations
  -



# Ready?

## Open the Lesson 1 Notebook :

Lesson1.ipynb





Let's Start !