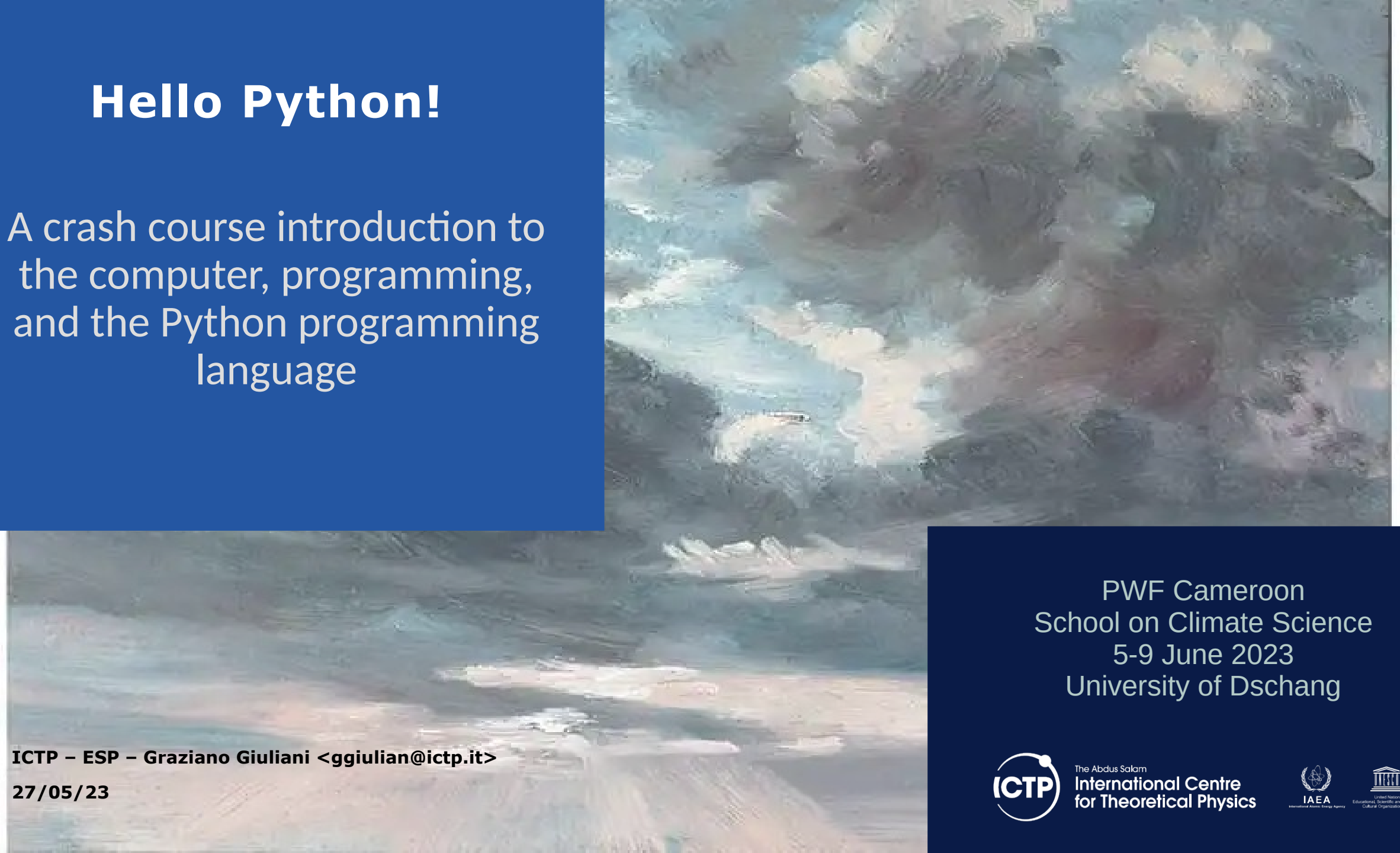
Hello Python!

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



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

27/05/23

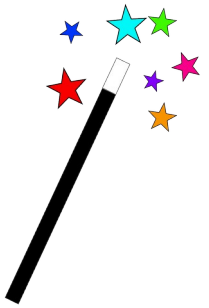
PWF Cameroon
School on Climate Science
5-9 June 2023
University of Dschang

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

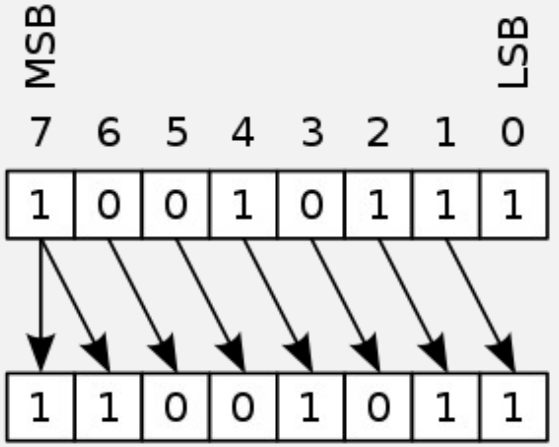
Computer Program



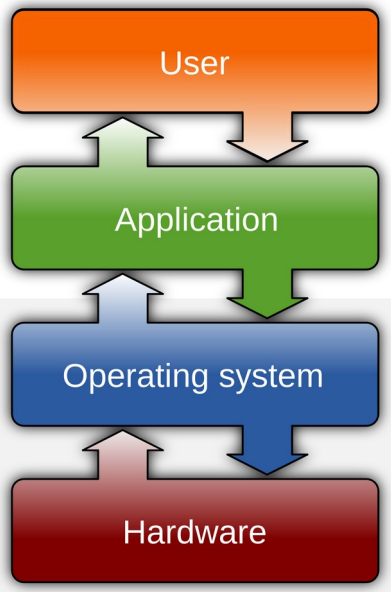
- A computer program is a sequence of instructions modifying a string of bits representing the status of a system. The initial status is the input of the program, the final state is the output of the program
- The instructions are stored in bit patterns themselves, following conventions defined by the 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

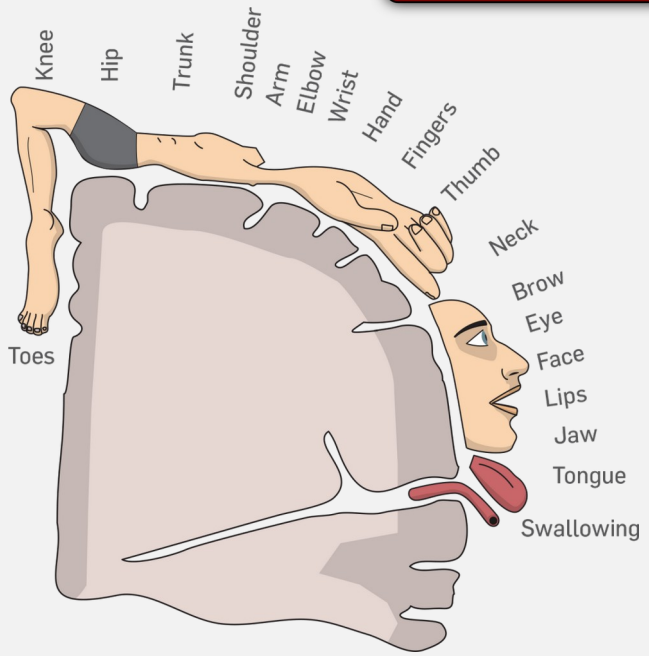


Hardware : the metal box in which the “magic” happens
Software : the instructions describing how to go from input to output state

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

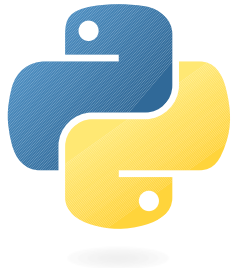
A programming language is a system of notation for writing computer programs. Most programming languages are text-based formal languages. [...]

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

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

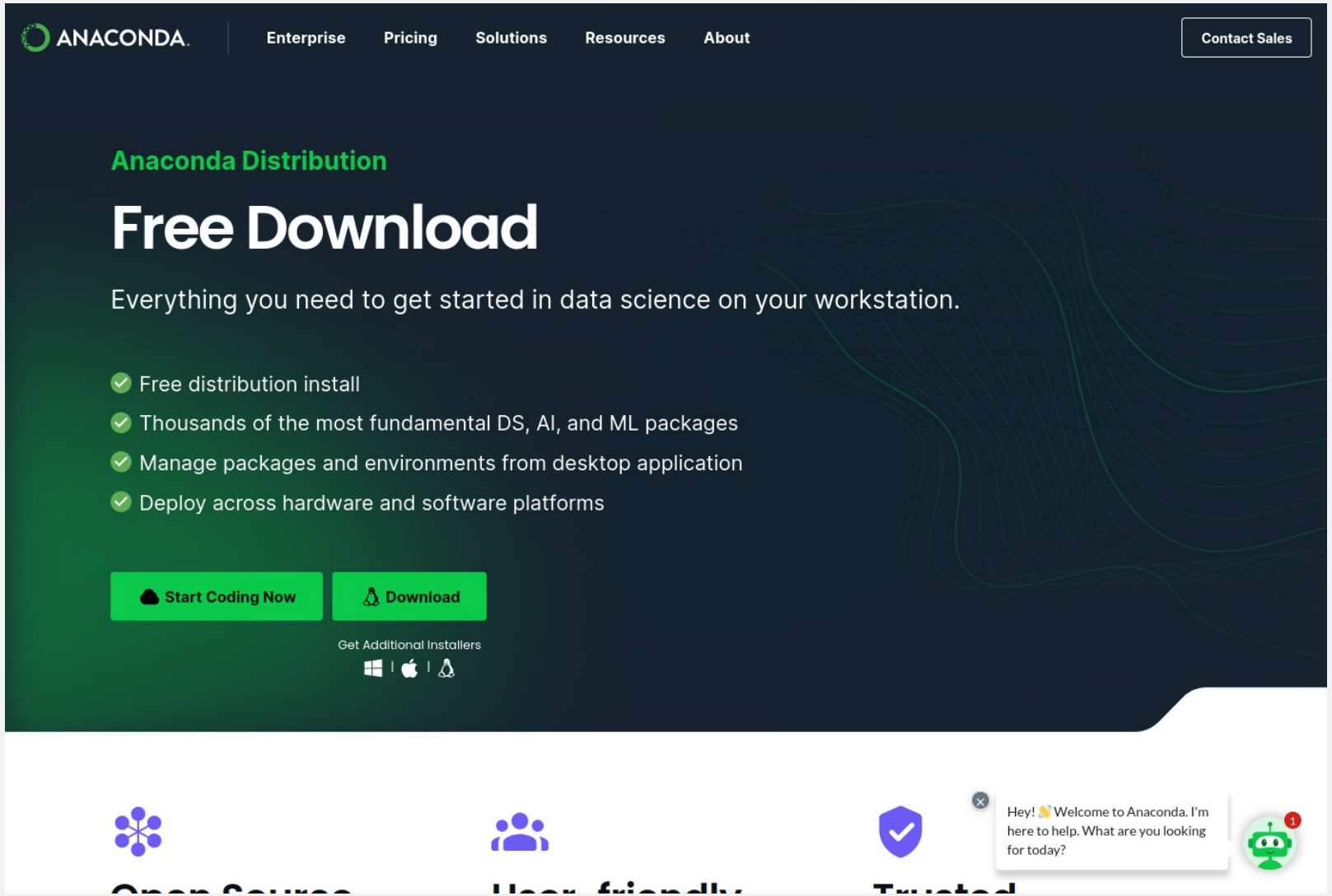
Active Python Releases

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

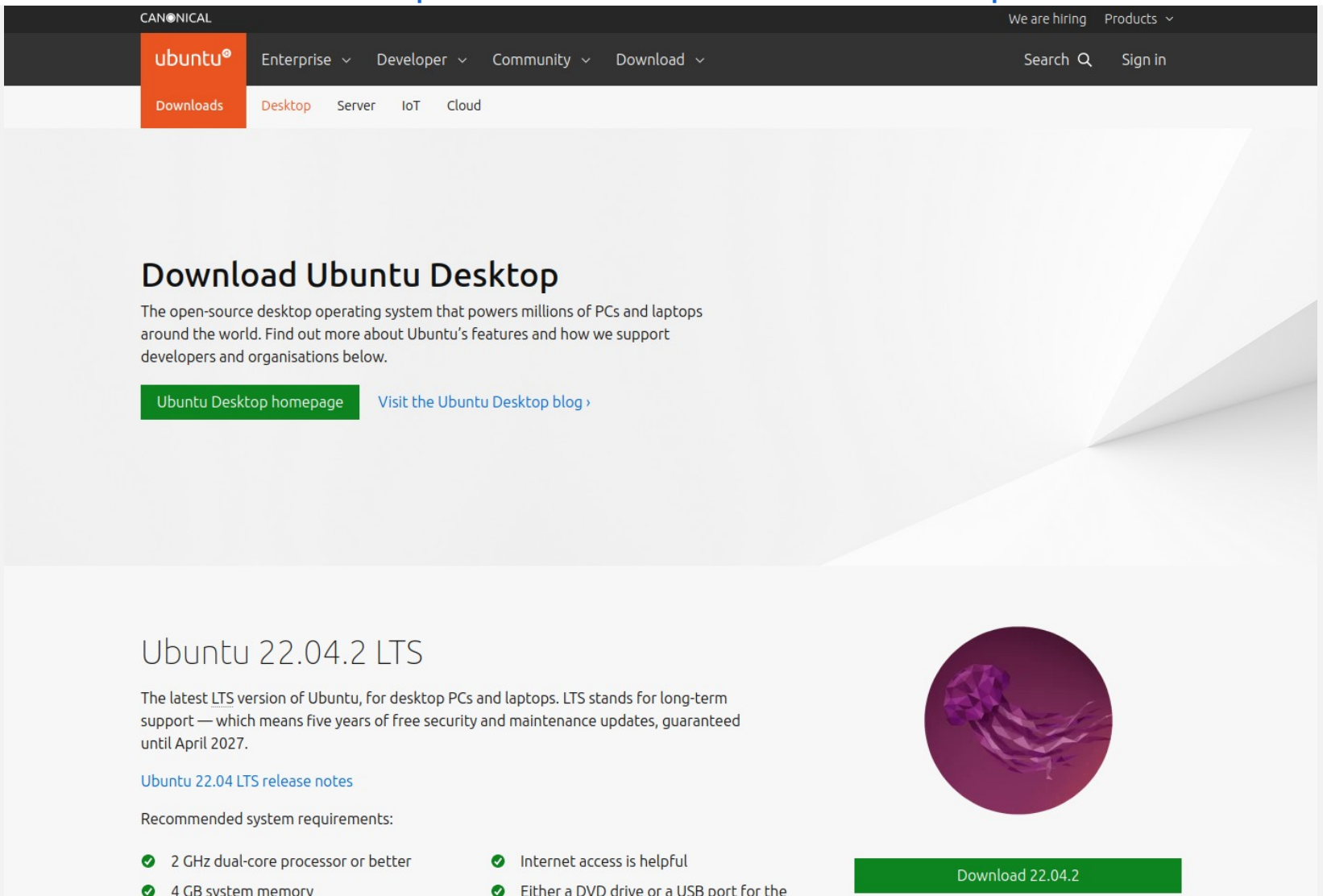
Install Science Python

https://www.anaconda.com/download



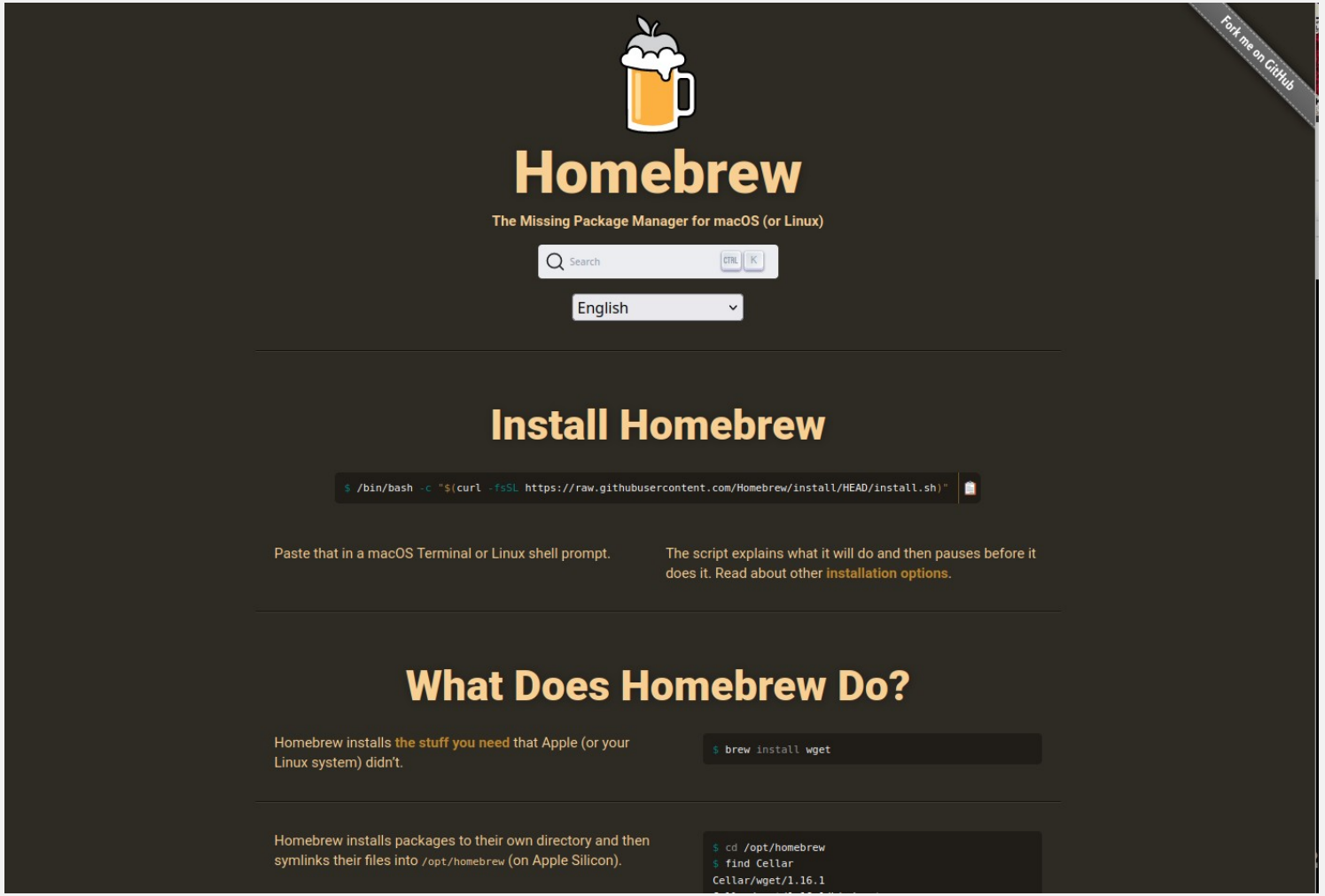
Install a full FREE OS

https://ubuntu.com/download/desktop

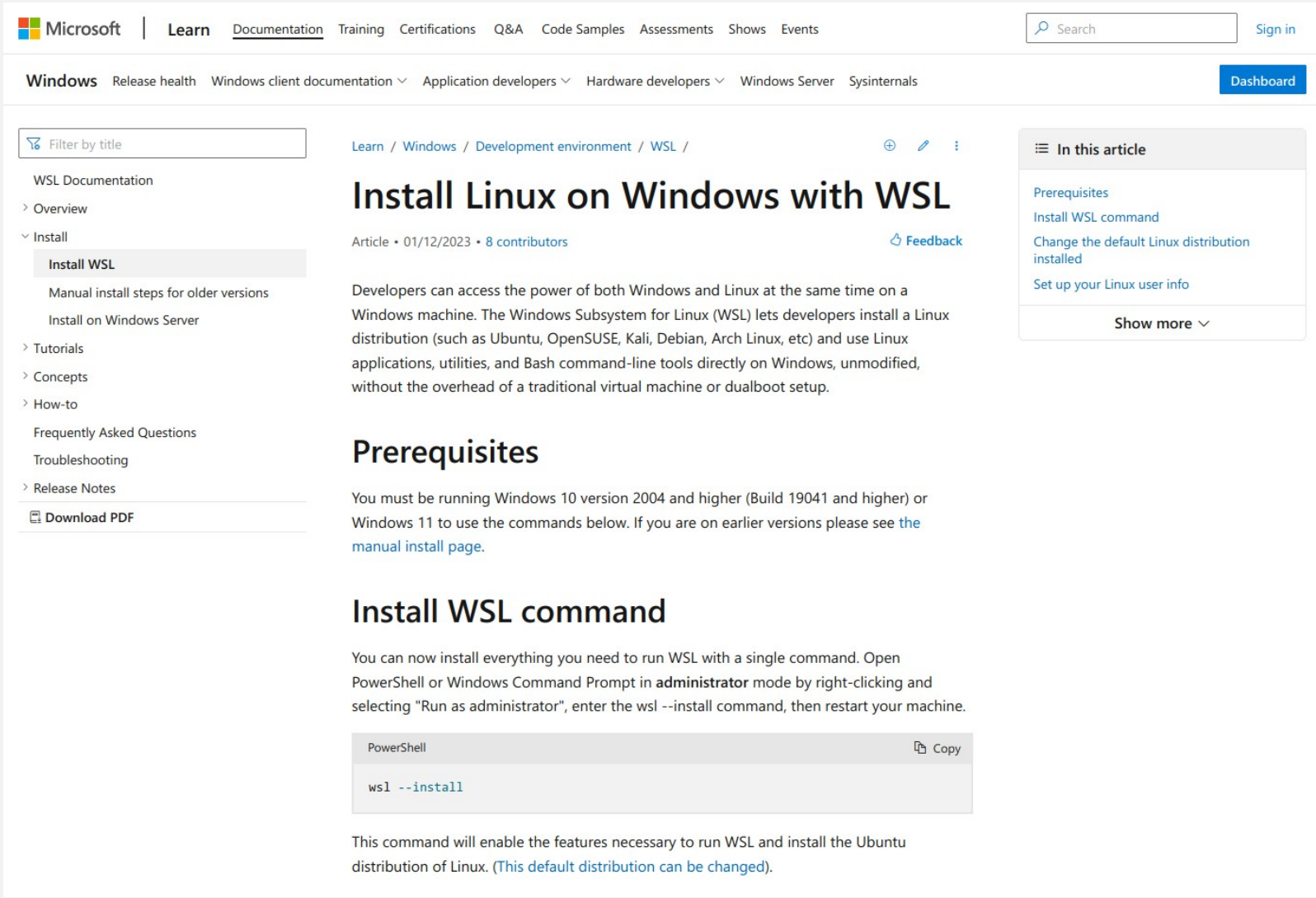


Free Software on MacOS

https://brew.sh



WSL Windows Subsystem for Linux



ICTP cloud desktop

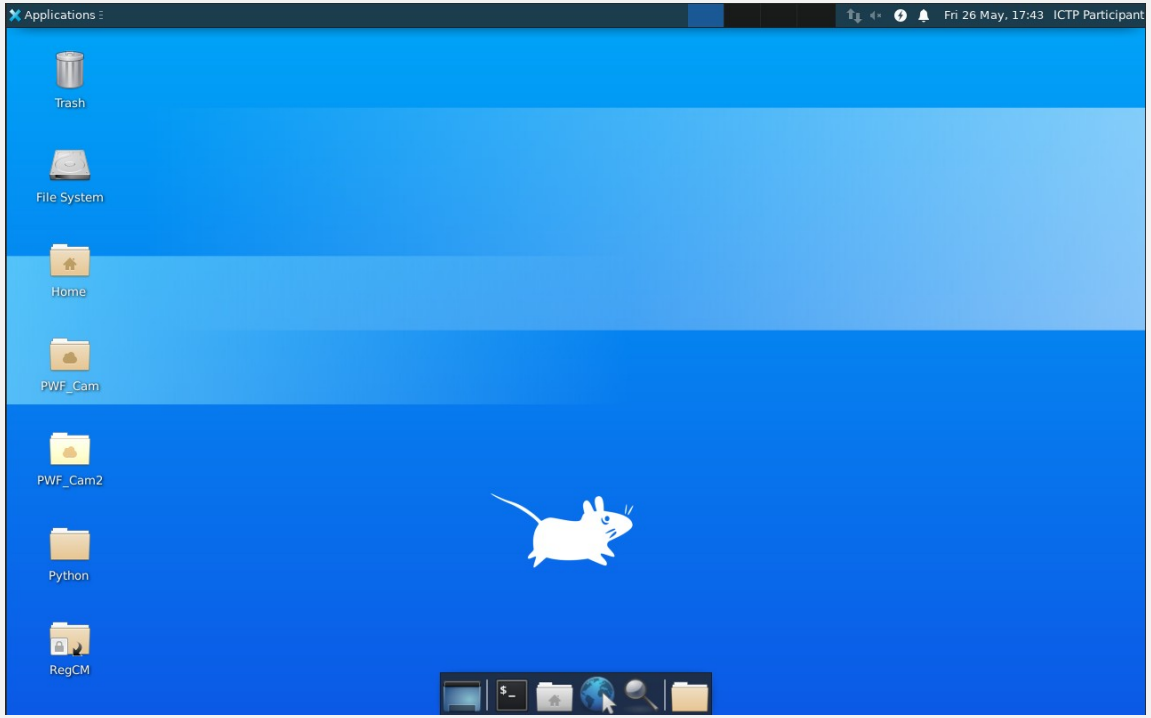
Every participant has access to a cloud based Linux desktop.

Who has NOT received the ICTP e-mail ?
Who has NOT managed to complete the key setup ?

NOW:

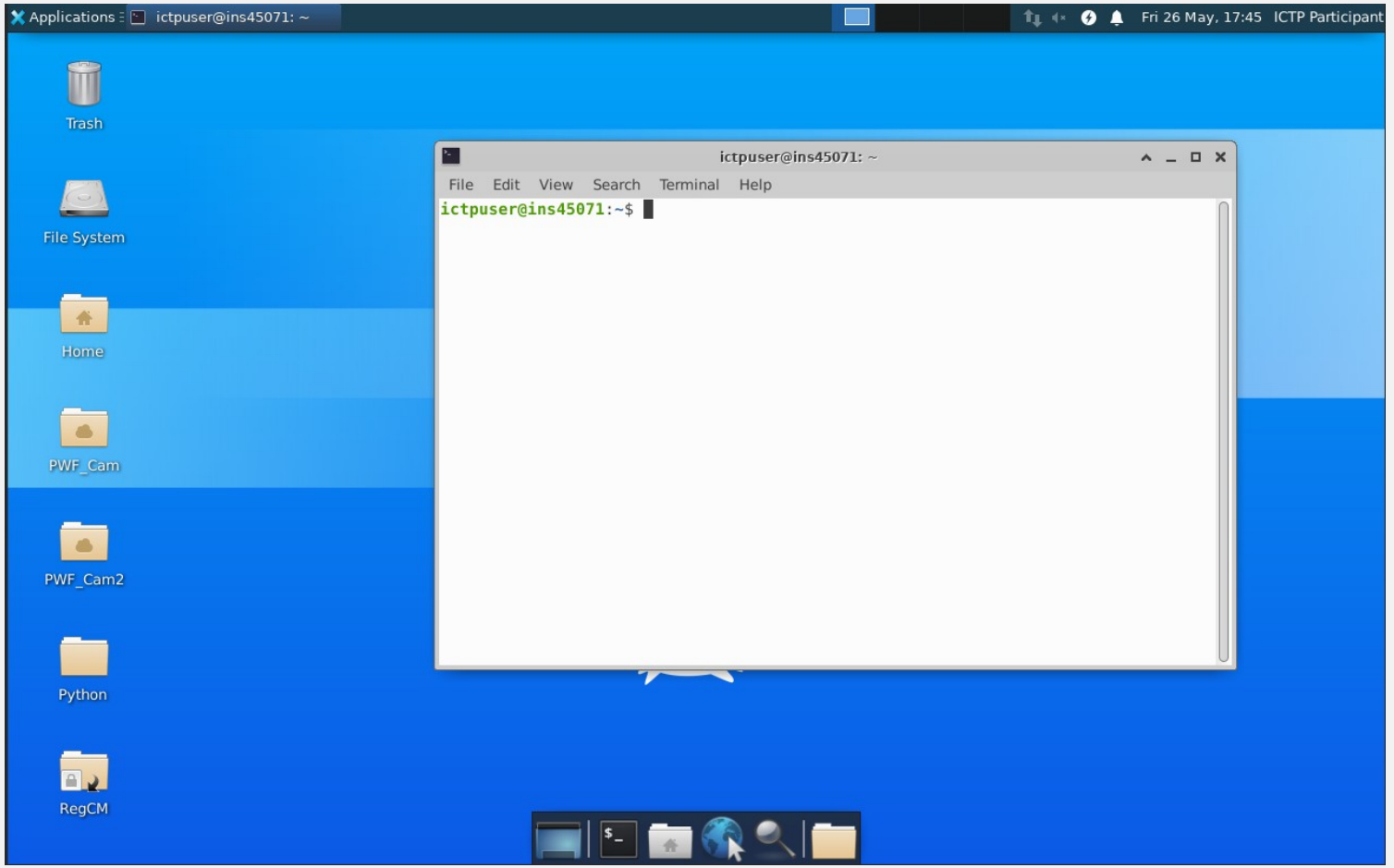
1) Can you connect to the system ?
2) Who has NO idea how to proceed ?

Open a terminal !



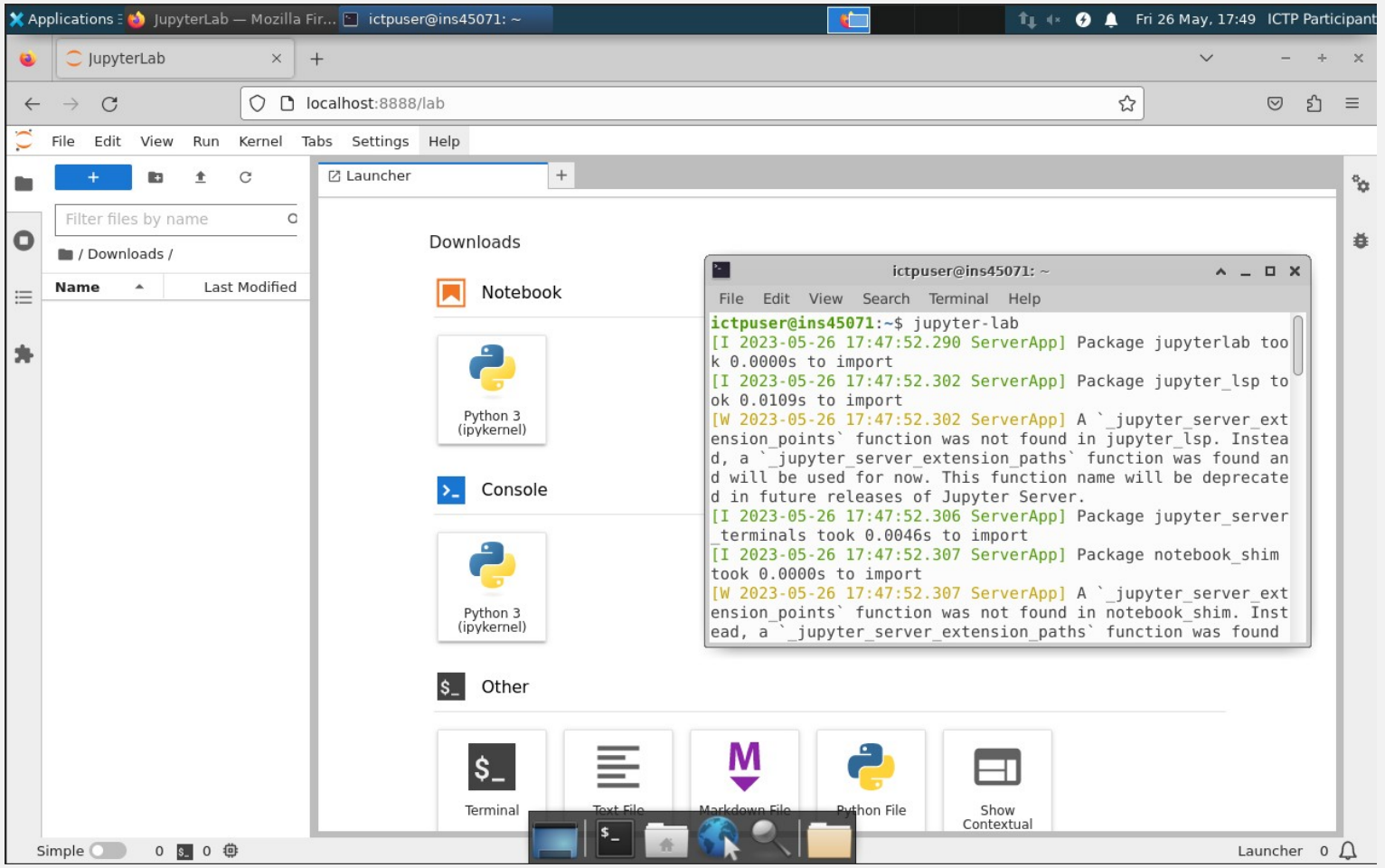
Linux Terminal

WHO HAS NOT IDEA WHAT THIS IS ???



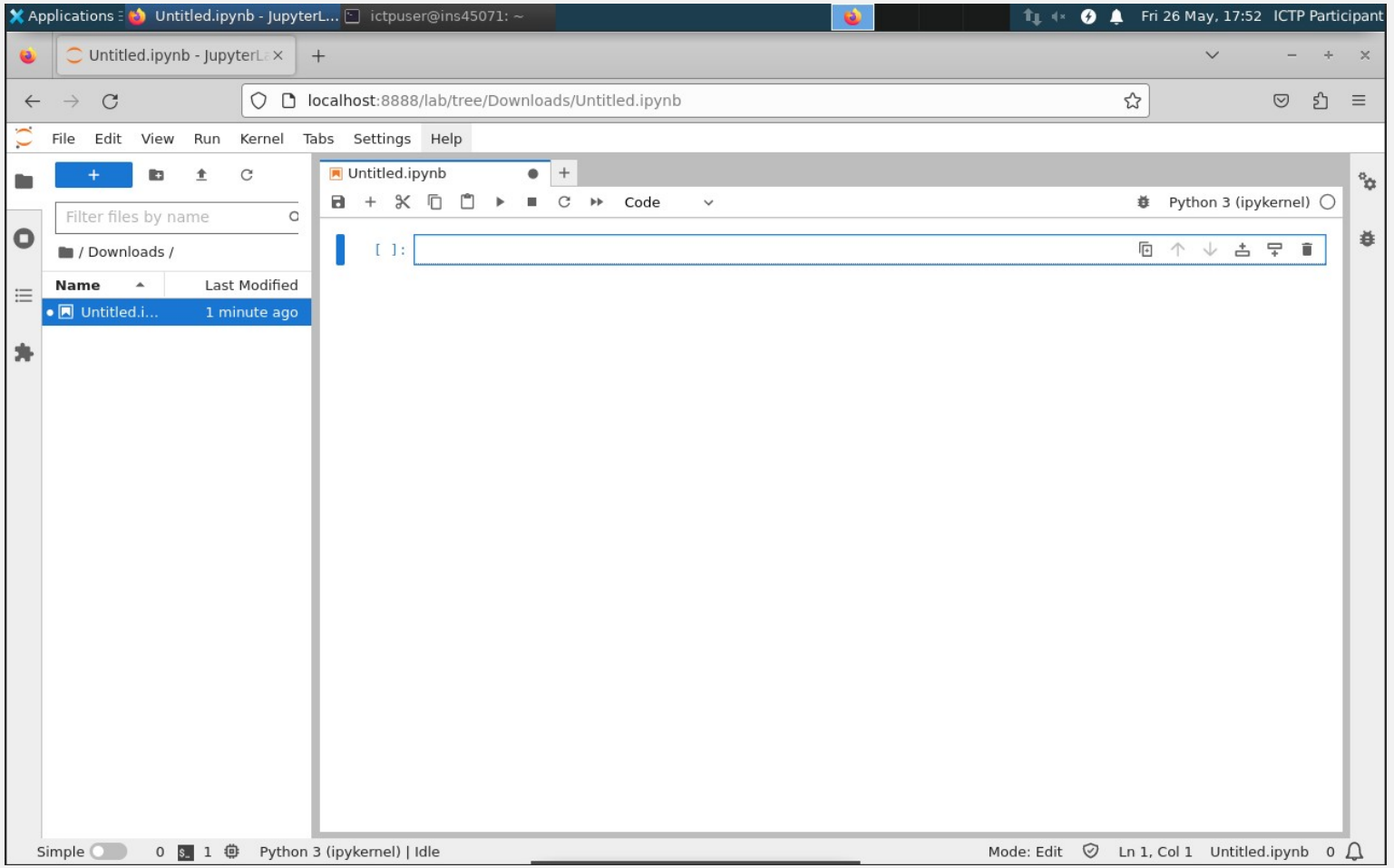
Jupyter Terminal : Start

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



Jupyter Terminal

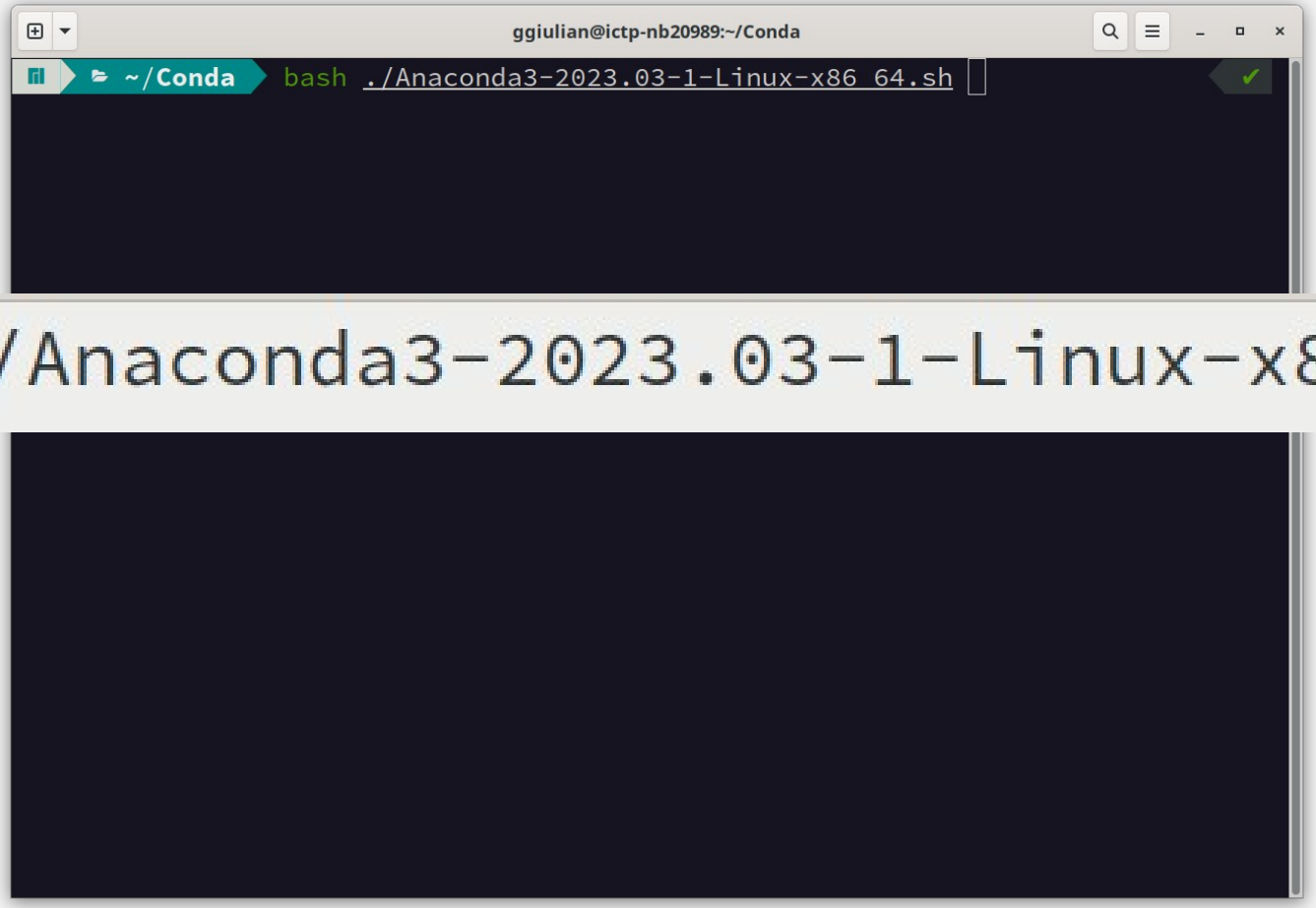
Anyone here is ready to go!



Anaconda installer

Run the installer available for your OS

On Linux :

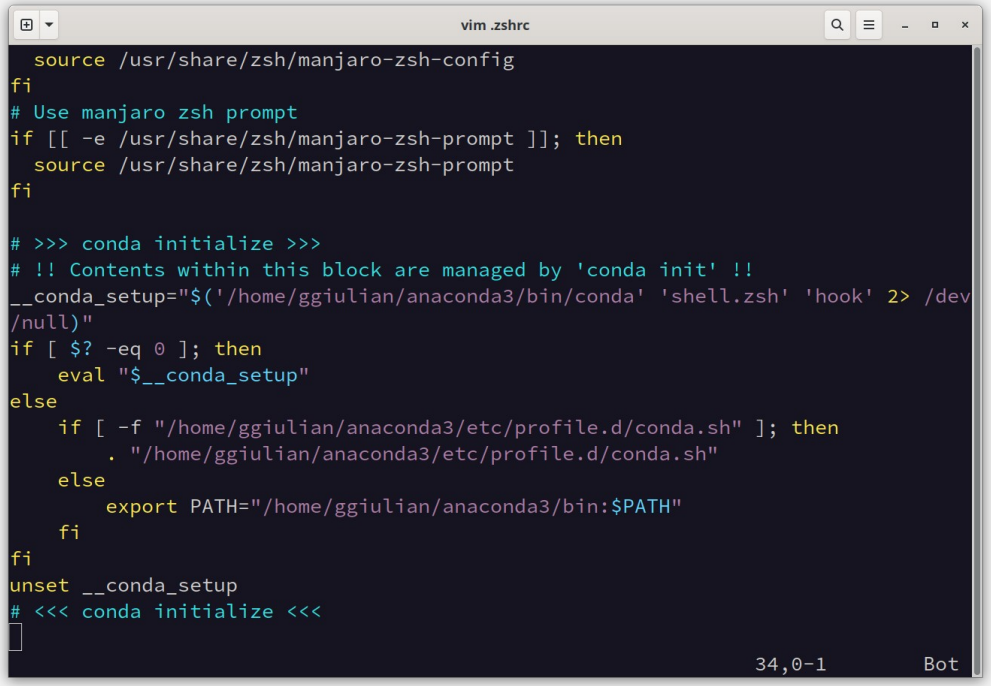


bash ./Anaconda3-2023.03-1-Linux-x86_64.sh

Activation

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

On my zsh :



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 ✘ base
# conda environments:
#
base                  *  /home/ggiulian/anaconda3

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

## Package Plan ##

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



Proceed ([y]/n)? yes

Preparing tran
Verifying tran
Executing tran
#
# To activate
#
#     $ conda
#
# To deactivat
#
#     $ conda deactivate
```

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

```
~ 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
```

```
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 Scienc
5-9 June 2023
University of Dschang

The Abdus Salam International Centre for Theoretical Physics

IAEA

The Computer

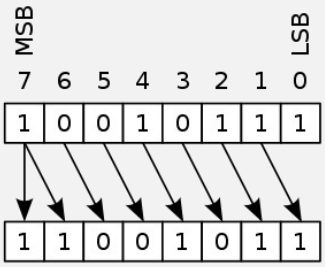
- A machine is a physical system using power to apply forces and control movement to perform an action.
- A digital bit is the minimal amount of information representing a logical state of a system with two possible values [0-1]
- A computer is an electronic digital machine carrying out logical operations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

Computer Program

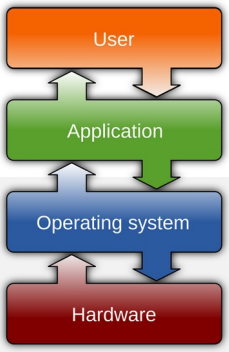
- A computer program is a sequence of instructions modifying a string of bits representing the status of a system. The initial status is the input of the program, the final state is the output of the program
- The instructions are stored in bit patterns themselves, following conventions defined by the designer of a piece 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

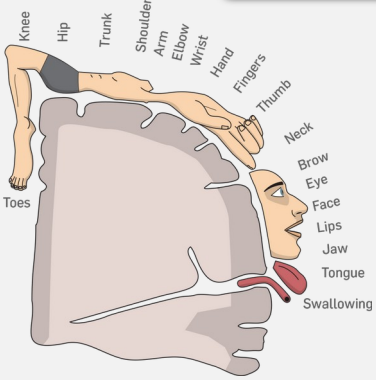


Hardware : the metal box in which the “magic” happens
Software : the instructions describing how to go from input to output state

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

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.

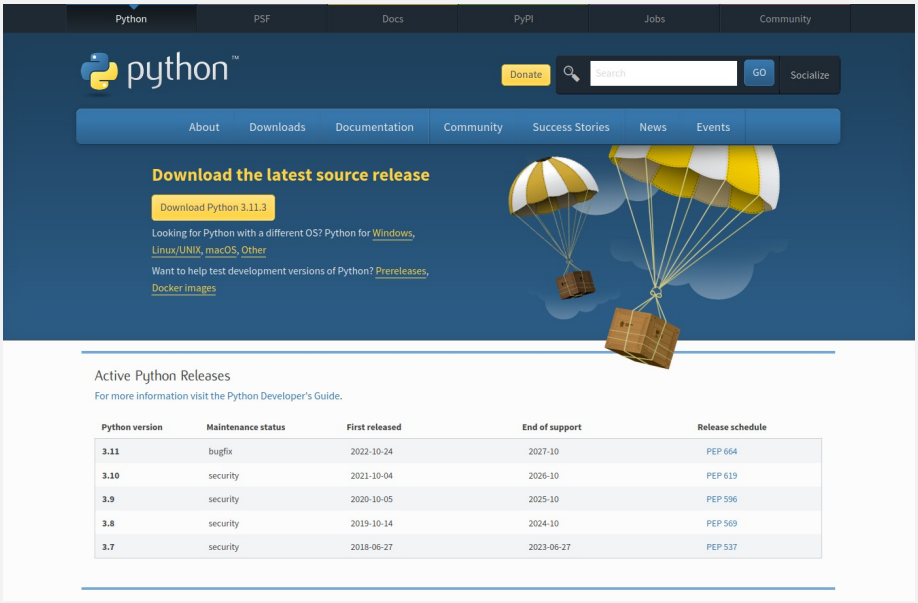
Input phase	: User types in the program, line by line or in a text file
Interpretation phase	: Python program reads the instruction, validates syntax
Execution phase	: The translated byte code is executed by the computer
Completion phase	: The program completes and the interpreter gives control to the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

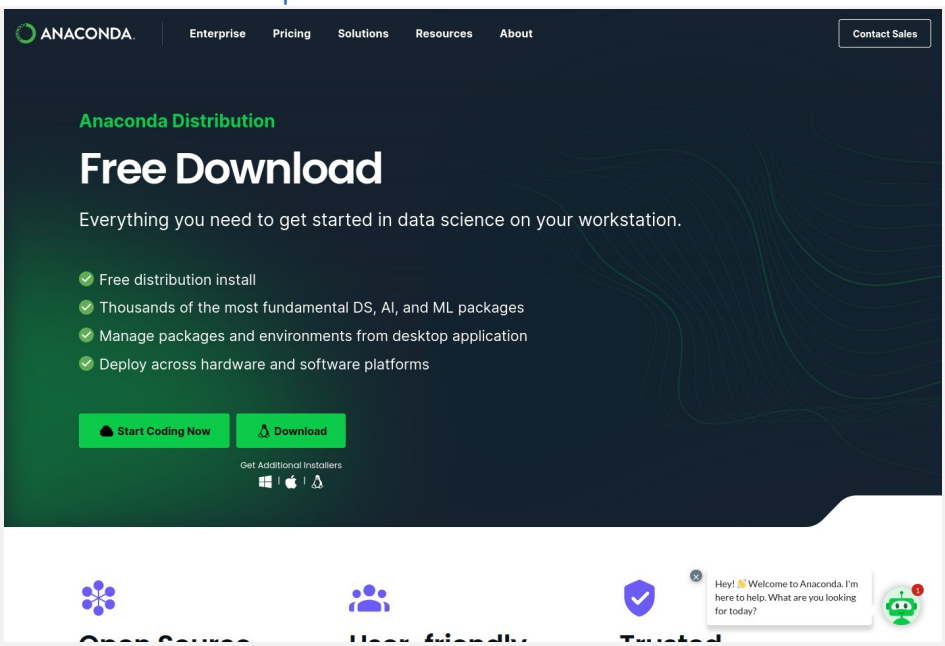
Install the interpreter

https://www.python.org/downloads



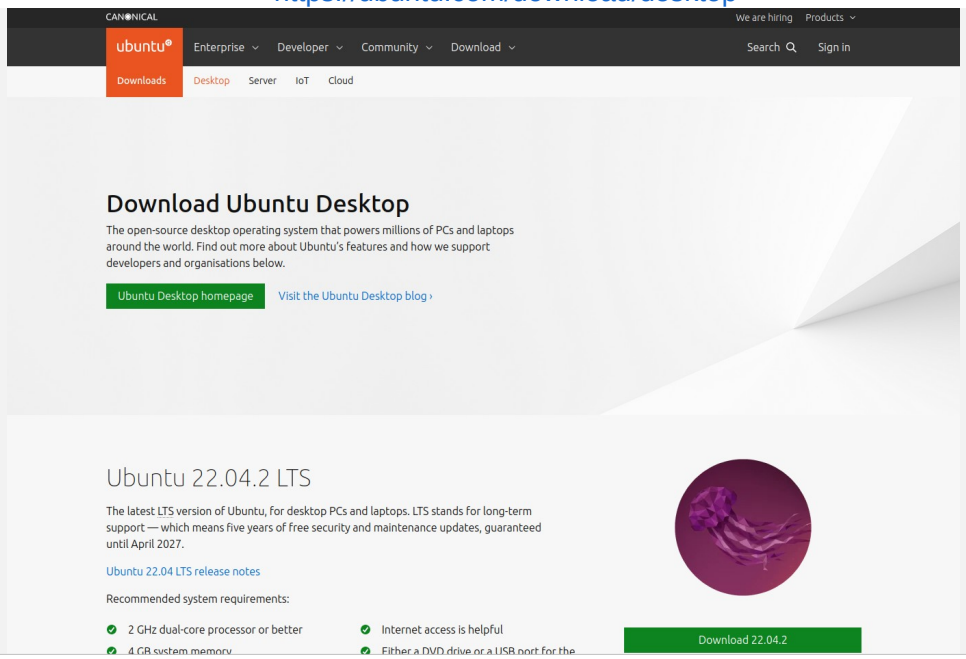
Install Science Python

https://www.anaconda.com/download



Install a full FREE OS

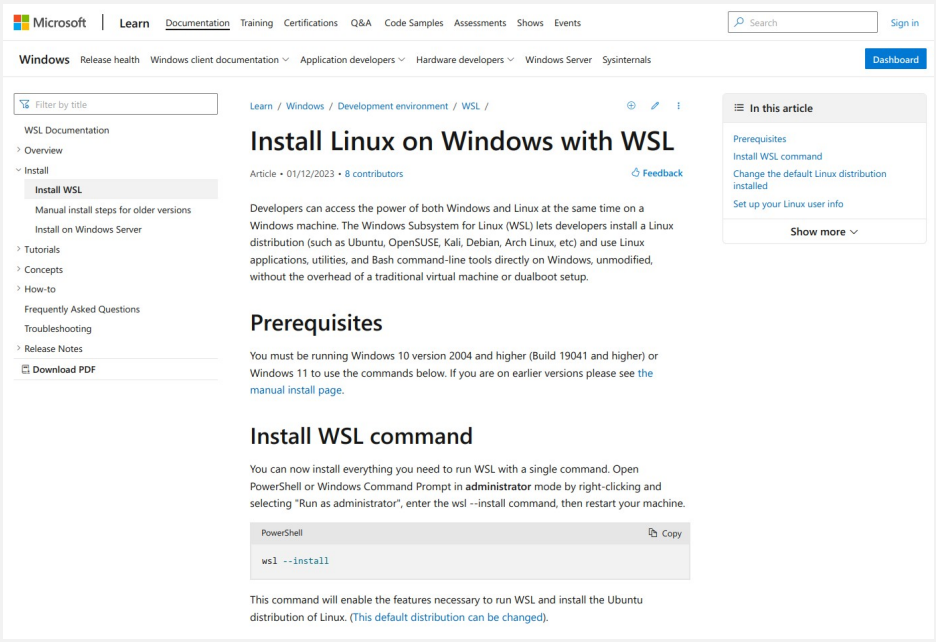
https://ubuntu.com/download/desktop



Free Software on MacOS

https://brew.sh

WSL Windows Subsystem for Linux



ICTP cloud desktop

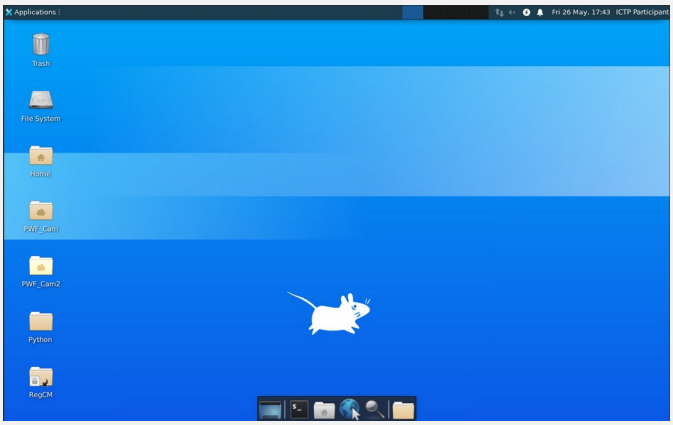
Every participant has access to a cloud based Linux desktop.

Who has NOT received the ICTP e-mail ?
Who has NOT managed to complete the key setup ?

NOW:

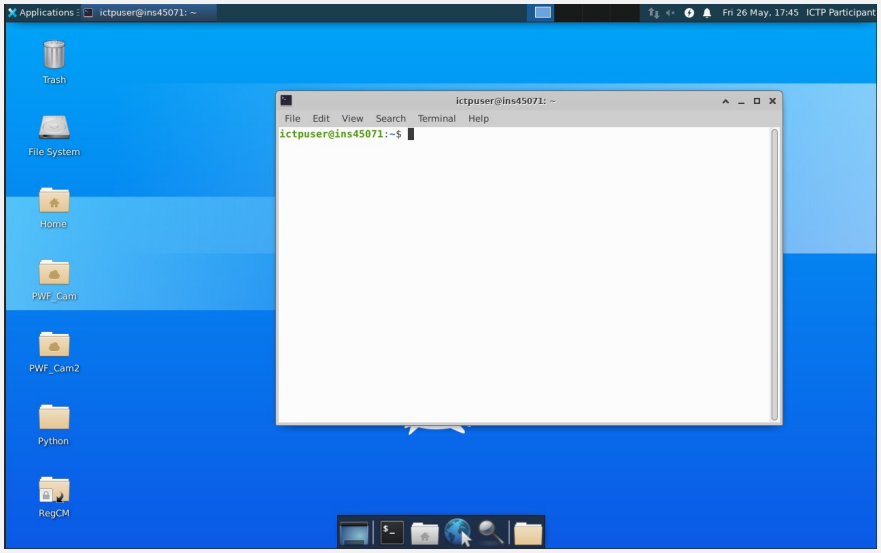
1) Can you connect to the system ?
2) Who has NO idea how to proceed ?

Open a terminal !



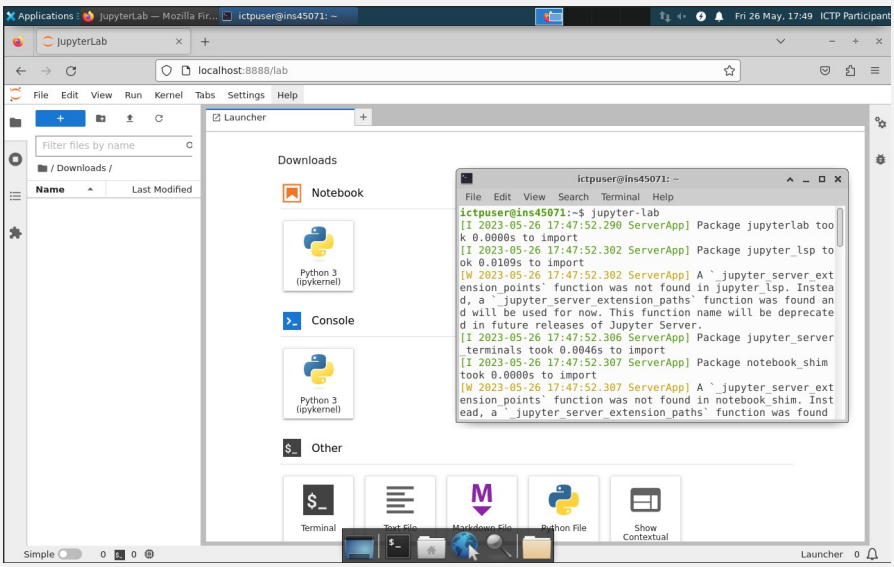
Linux Terminal

WHO HAS NOT IDEA WHAT THIS IS ???



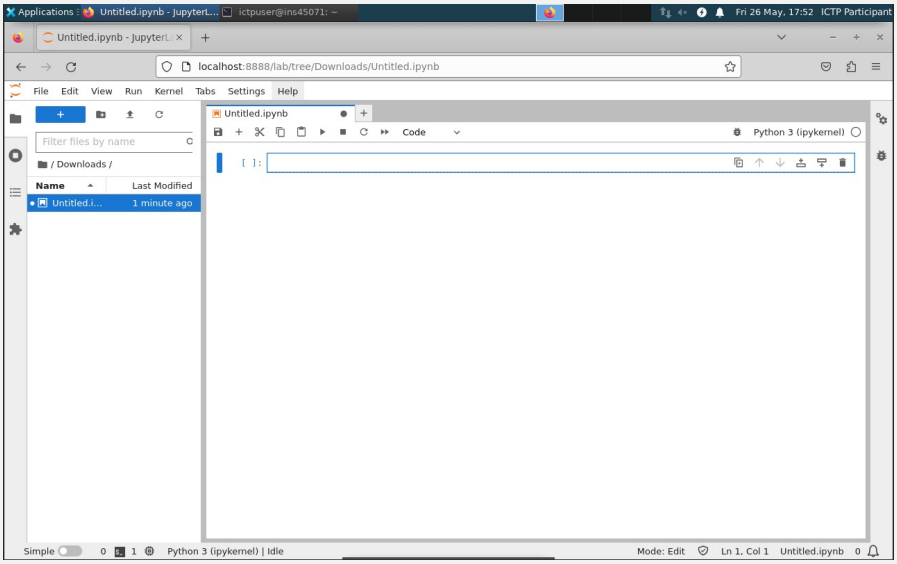
Jupyter Terminal : Start

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



Jupyter Terminal

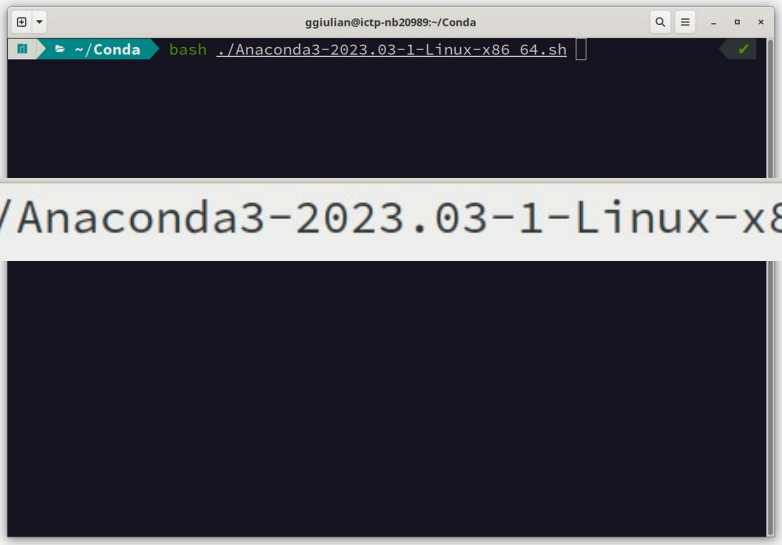
Anyone here is ready to go!



Anaconda installer

Run the installer available for your OS

On Linux :

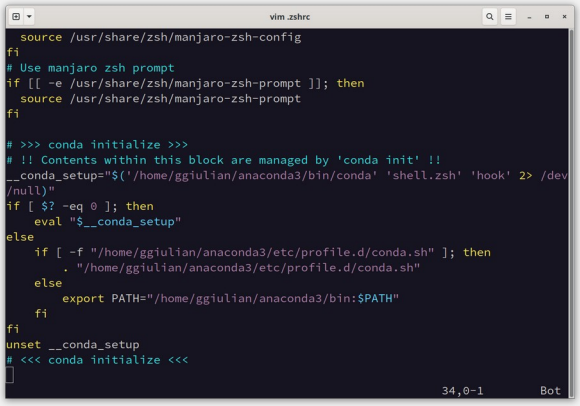



```
bash ./Anaconda3-2023.03-1-Linux-x86_64.sh
```

Activation

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

On my zsh :

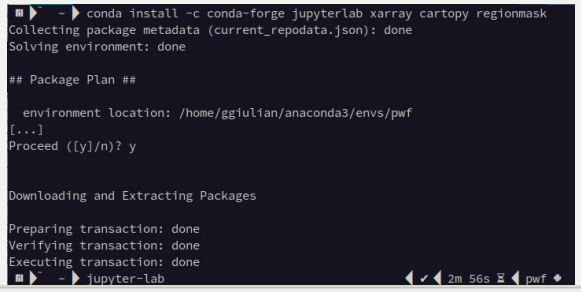


To remove conda, comment or delete the lines between the “conda initialize” and to get back the storage conda is using remove the anaconda3 directory (in my case it is in `/home/ggiulian/`)

Environment

Create a new environment for the PWF activity

```
conda 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:

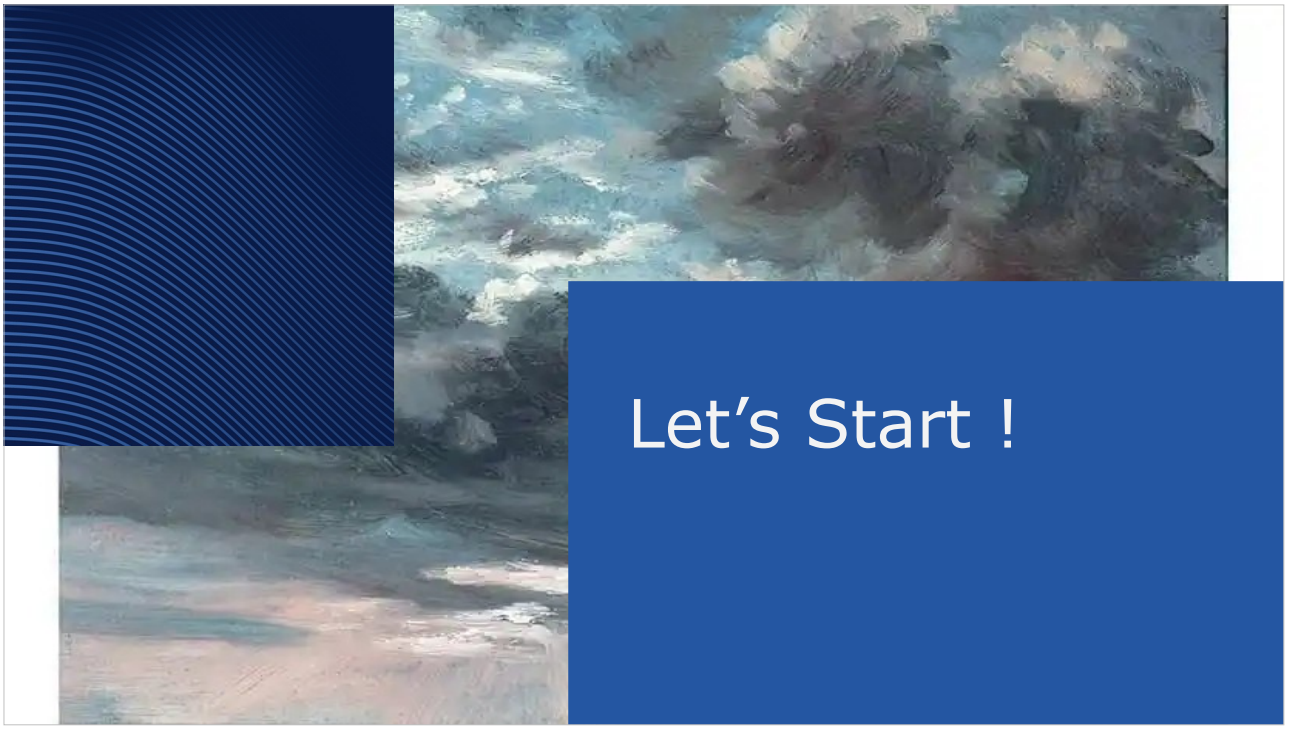
1) Python releases 3.11.3
   1) Source file for Linux
   2) Windows 64 bit interpreter
   3) MacOS universal interpreter
2) Anaconda 2023.05
   1) Linux x86_64
   2) Windows 64 bit installer
   3) MacOS installers

- 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 !