



Introducción a Python

Ciencia de Datos en Python

San Sebastián – 25 de enero de 2021



Introducción a Python

Ciencia de Datos en Python

San Sebastián – 25 de enero de 2021

Estructura del curso

INTRODUCCIÓN A PYTHON
(5 sesiones)

Jupyter, Spyder y
Funciones básicas

Asentar bases de la
programación en Python

MANIPULACIÓN DE DATOS
(5 sesiones)

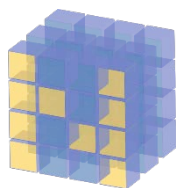
Numpy y Pandas

Manejo de datos y estructuras de datos.
Programación funcional

VISUALIZACIÓN
(4 sesiones)

Matplotlib, Bokeh, Dash, Plotly

Desarrollo de Aplicación



NumPy

Pandas



matplotlib



plotly



Dash



bokeh

Estructura del curso

INTRODUCCIÓN A PYTHON
(5 sesiones)

Jupyter, Spyder y
Funciones básicas

Asentar bases de la
programación en Python

MANIPULACIÓN DE DATOS
(5 sesiones)

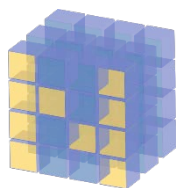
Numpy y Pandas

Manejo de datos y estructuras de datos.
Programación funcional

VISUALIZACIÓN
(4 sesiones)

Matplotlib, Bokeh, Dash, Plotly

Desarrollo de Aplicación



NumPy

Pandas



matplotlib



plotly



Dash



bokeh

¿Qué se puede hacer con Python?

Cuadros de mando avanzados

- <https://dash-gallery.plotly.host/dash-nlp/>

Automatización de informes

- <https://dash-gallery.plotly.host/dash-financial-report/full-view>

Desarrollo de aplicaciones

- <https://dash-gallery.plotly.host/dashr-uber-rasterizer/>

Inteligencia artificial

- <https://dash-gallery.plotly.host/dash-svm/>

Videojuegos...

¿Qué se puede hacer con Python?

Cuadros de mando avanzados

- <https://dash-gallery.plotly.host/dash-nlp/>

Automatización de informes

- <https://dash-gallery.plotly.host/dash-financial-report/full-view>

Desarrollo de aplicaciones

- <https://dash-gallery.plotly.host/dashr-uber-rasterizer/>

Inteligencia artificial

- <https://dash-gallery.plotly.host/dash-svm/>

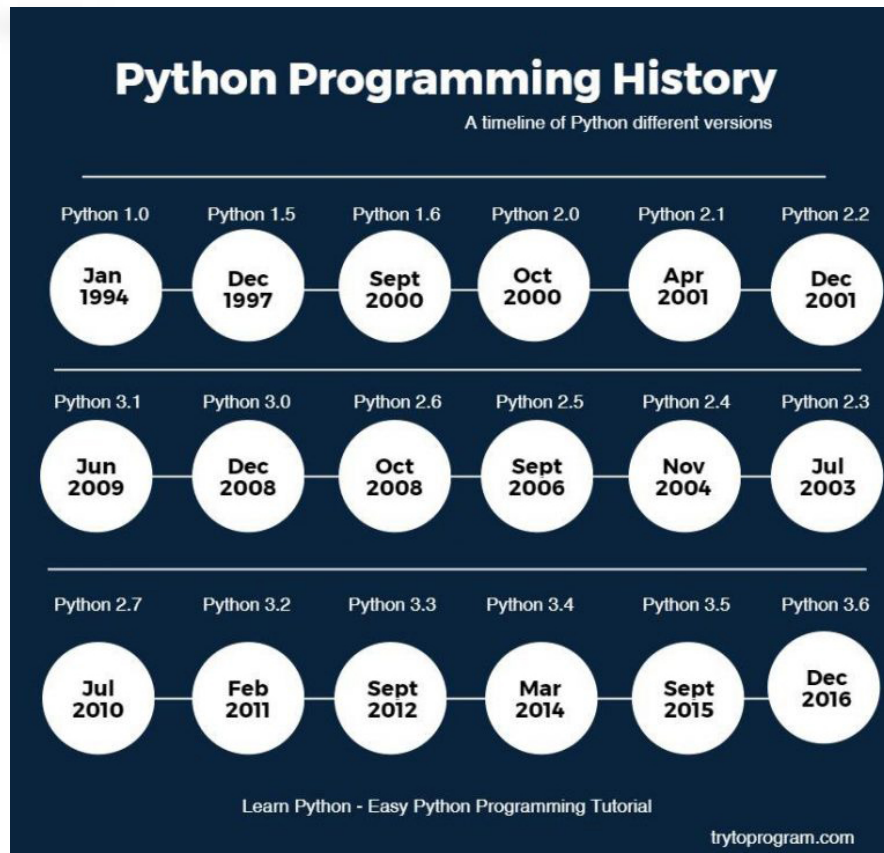
Videojuegos...

¿Qué es Python?



- Lenguaje de programación interpretado.
- No necesita compilación
- Lenguaje de programación de (muy) alto nivel.
- Programación Funcional.
- Programación Orientada a Objetos.
- Se basa en la idea de una sintaxis que de lugar a un código fácil de leer.

¿Qué es Python?



- Lenguaje de programación interpretado.
 - No necesita compilación
 - Lenguaje de programación de (muy) alto nivel.
 - Programación Funcional.
 - Programación Orientada a Objetos.
 - Se basa en la idea de una sintaxis que de lugar a un código fácil de leer.
-
- Creado en a finales de los 80 por Guido van Rossum.
 - Gran salto de la versión 2 a la versión 3. Se siguen actualizando ambas. Últimas versiones:
 - **v2.7.16**: 2 de marzo de 2019. (**Desaparecerá en 2020**).
 - **v3.7.3**: 25 de marzo de 2019.

¿Qué vamos a hacer en el curso?

- Aprender las bases de programación en Python.
- Aprender distintos entornos de programación en Python. → **IDE**
- Aprender las bases de un entorno de programación REAL → **GIT**
- Aprender distintas aplicaciones para las que se puede utilizar Python.



¿Cómo vamos a trabajar?



Distribución “open-source” para Python y R.

Simplifica el manejo de librerías permitiendo crear distintos entornos de trabajo.

De gran utilidad debido a incompatibilidades entre módulos para Python 2.x y 3.x.

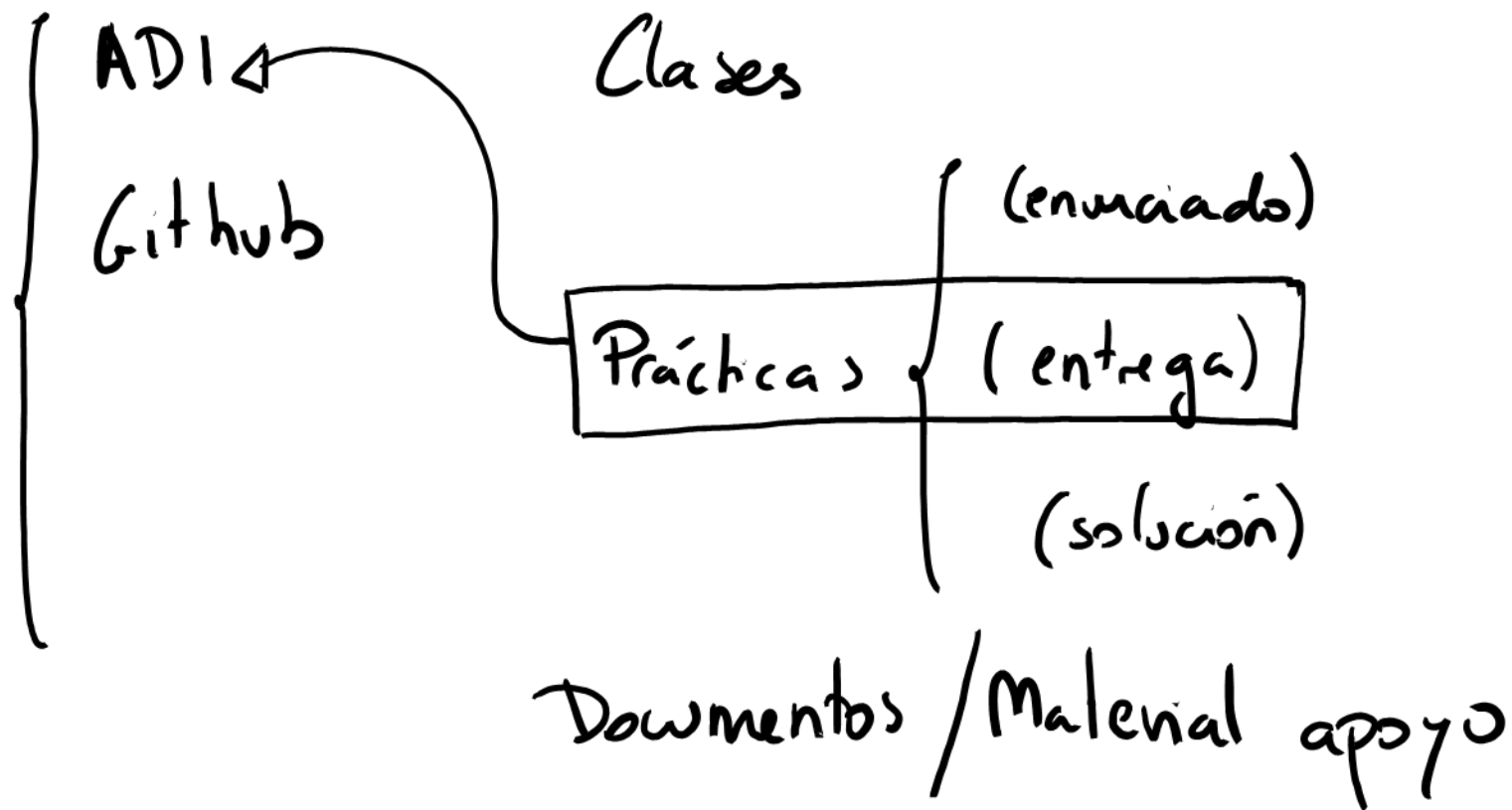
Evaluación

- ▶ **Trabajo (60%)**
- ▶ **Entregas (40%)**

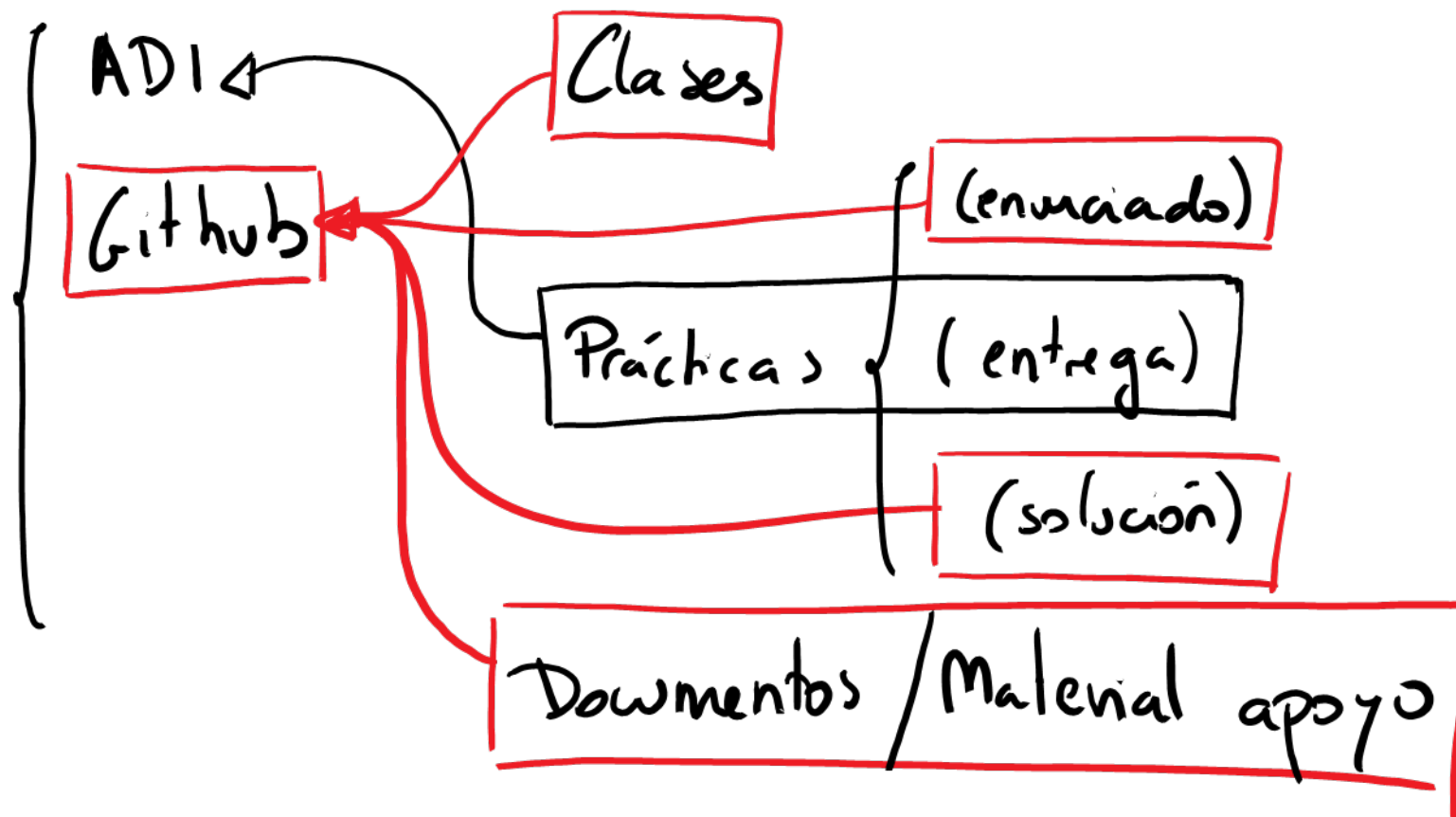
Evaluación

- ▶ **Trabajo (60%)**
- ▶ **Entregas (40%)**

¿Cómo vamos a trabajar?



¿Cómo vamos a trabajar?

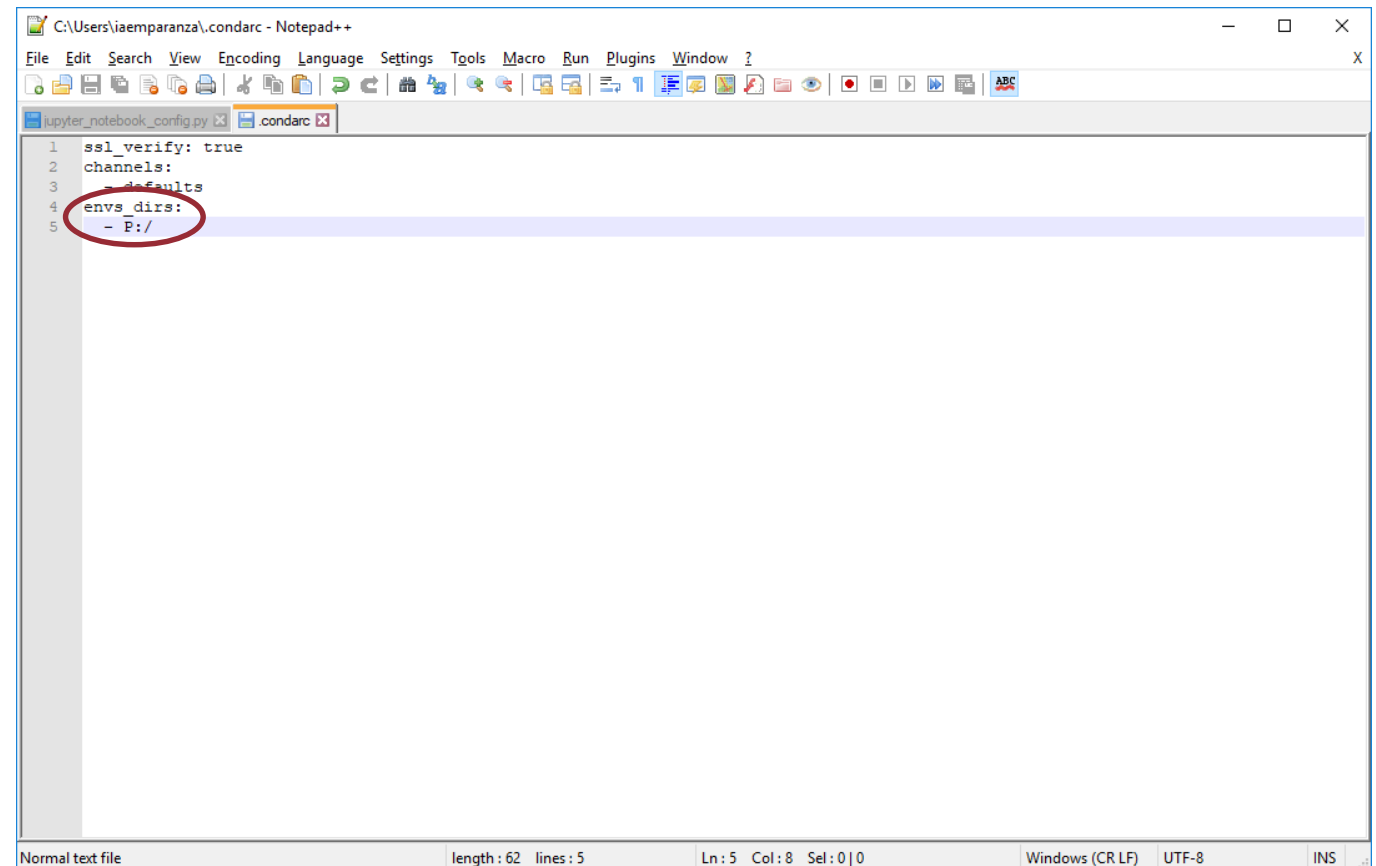


¿Cómo vamos a trabajar?

Para que los entornos que creamos no se borren cada vez que cerremos sesión en estos ordenadores tendremos que guardarlos en la unidad P:/. Para ello seguiremos los siguientes pasos:

- Abrir Anaconda Navigator.
- Ir a la carpeta C:\Users**fcarazo**\ y abrir el archivo .condarc con Notepad.
- Añadir el siguiente texto como se muestra en la imagen:
envs_dirs:
- P:/
- Cerrar Anaconda Navigator.

Tendremos que hacer esto cada vez que iniciemos sesión en estos ordenadores.



```
1 ssl_verify: true
2 channels:
3   defaults
4   envs_dirs:
5     - P:/
```

¿Cómo vamos a trabajar?

Anaconda Navigator

File Help

ANACONDA NAVIGATOR

Sign in to Anaconda Cloud

Home

Environments ¹

Learning

Community

Documentation

Developer Blog

Twitter YouTube GitHub

Applications on base (root) Channels Refresh

Application	Version	Description	Action
JupyterLab	0.35.4	An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.	Launch
Jupyter Notebook	5.7.8	Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.	Launch
Qt Console	4.4.3	PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.	Launch
Spyder	3.3.3	Scientific Python Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features	Launch
VS Code	1.35.0	Streamlined code editor with support for development operations like debugging, task running and version control.	Launch
Glueviz	0.13.3	Multidimensional data visualization across files. Explore relationships within and among related datasets.	Install
Orange 3	3.19.0	Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.	Install
RStudio	1.1.456	A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.	Install

¿Cómo vamos a trabajar?

Anaconda Navigator

File Help

ANACONDA NAVIGATOR

Sign in to Anaconda Cloud

Home

Environments

Learning

Community

Documentation

Developer Blog

Search Environments

base (root)

python3.6_Spotipy

python3.6_cplex1.2.8

python3.7

Installed Channels Update index... Search Packages

Name	T	Description	Version
✓ _ipyw_lab_nb_ex...	○	A configuration metapackage for enabling anaconda-bundled jupyter extensions	0.1.0
✓ alabaster	○	Configurable, python 2+3 compatible sphinx theme.	0.7.12
✓ anaconda	○	Simplifies package management and deployment of anaconda	2019.03
✓ anaconda-client	○	Anaconda.org command line client library	1.7.2
✓ anaconda-project	○	Tool for encapsulating, running, and reproducing data science projects	0.8.2
✓ asn1crypto	○	Python asn.1 library with a focus on performance and a pythonic api	0.24.0
✓ astroid	○	A abstract syntax tree for python with inference support.	2.2.5
✓ astropy	○	Community-developed python library for astronomy	3.1.2
✓ atomicwrites	○	Atomic file writes.	1.3.0
✓ attrs	○	Attrs is the python package that will bring back the joy of writing classes by relieving you from the drudgery of implementing object protocols (aka dunder methods).	19.1.0
✓ babel	○	Utilities to internationalize and localize python applications	2.6.0
✓ backcall	○	Specifications for callback functions passed in to an api	0.1.0
✓ backports	○		1.0
✓ backports.os	○	Backport of new features in python's os module	0.1.1
✓ backports.shutil_g...	○	A backport of the get_terminal_size function from python 3.3's shutil.	1.0.0
✓ beautifulsoup4	○	Python library designed for screen-scraping	4.7.1
✓ bitarray	○	Efficient arrays of booleans -- c extension	0.8.3
✓ bkcharts	○	High level chart types built on top of bokeh	0.2
✓ blas	○		1.0
✓ bleach	○	Easy, whitelist-based html-sanitizing tool	3.1.0

260 packages available

Create Clone Import Remove

2

¿Cómo vamos a trabajar?

Create new environment X

Name:

Location: `...anza\AppData\Local\Continuum\anaconda3\envs\python3.7_CDPy`

Packages: ☒ Python

☐ R

3

¿Cómo vamos a trabajar?

Anaconda Navigator

File Help

ANACONDA NAVIGATOR

Sign in to Anaconda Cloud

Home

Environments

Learning

Community

Documentation

Developer Blog

Twitter YouTube GitHub

Create Clone Import Remove

Search Environments

base (root)

python3.6_Spotipy

python3.6_cplex1.2.8

python3.7

python3.7_CDPy

Installed Channels Update index... Search Packages

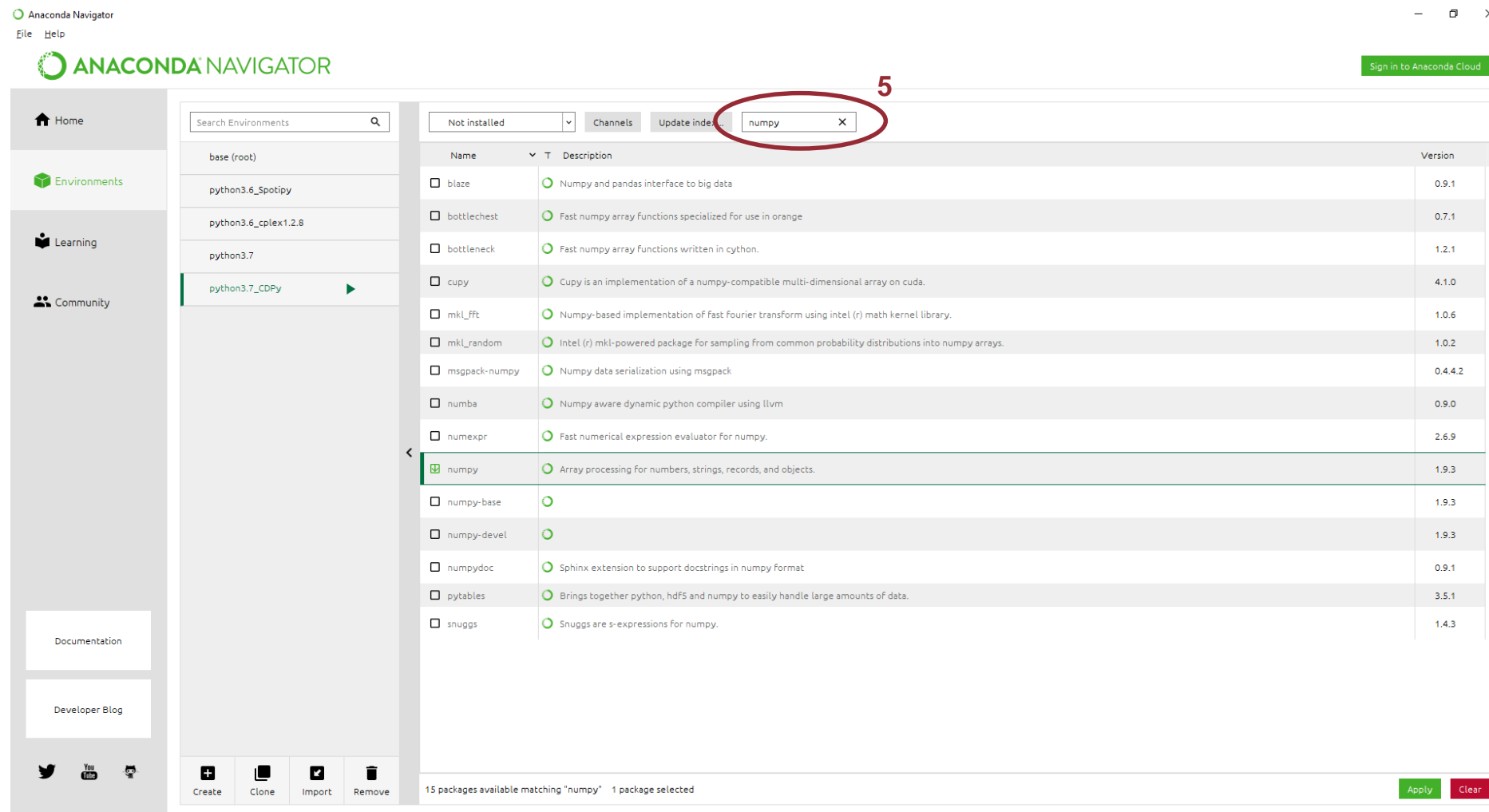
Name	T	Description	Version
✓ ca-certificates	○	Certificates for use with other packages.	2019.5.15
✓ certifi	○	Python package for providing mozilla's ca bundle.	2019.3.9
✓ openssl	○	OpenSSL is an open-source implementation of the ssl and tls protocols	1.1.1c
✓ pip	○	Pypa recommended tool for installing python packages	19.1.1
✓ python	○	General purpose programming language	3.7.3
✓ setuptools	○	Download, build, install, upgrade, and uninstall python packages	41.0.1
✓ sqlite	○	Implements a self-contained, zero-configuration, sql database engine.	3.28.0
✓ vc	○	A meta-package to impose mutual exclusivity among software built with different vs versions	14.1
✓ vs2015_runtime	○	Msvc runtimes associated with cl.exe version 19.15.26726 (vs 2017 update 8)	14.15.26...
✓ wheel	○	A built-package format for python.	0.33.4
✓ wincertstore	○	Python module to extract ca and crt certs from windows' cert store (ctypes based).	0.2

11 packages available

¿Cómo vamos a trabajar?

Librerías:

- bokeh
- matplotlib
- notebook
- numpy
- pandas
- scipy
- seaborn
- spyder



Anaconda Navigator

File Help

ANACONDA NAVIGATOR

Sign in to Anaconda Cloud

Home

Environments

Learning

Community

Documentation

Developer Blog

Search Environments

base (root)

python3.6_Spotipy

python3.6_cplex1.2.8

python3.7

python3.7_CDPy

Not installed

Channels

Update index

numpy

5

Name	Description	Version
<input type="checkbox"/> blaze	Numpy and pandas interface to big data	0.9.1
<input type="checkbox"/> bottleneck	Fast numpy array functions specialized for use in orange	0.7.1
<input type="checkbox"/> bottleneck	Fast numpy array functions written in cython.	1.2.1
<input type="checkbox"/> cupy	Cupy is an implementation of a numpy-compatible multi-dimensional array on cuda.	4.1.0
<input type="checkbox"/> mkl_fft	Numpy-based implementation of fast fourier transform using intel (r) math kernel library.	1.0.6
<input type="checkbox"/> mkl_random	Intel (r) mkl-powered package for sampling from common probability distributions into numpy arrays.	1.0.2
<input type="checkbox"/> msgpack-numpy	Numpy data serialization using msgpack	0.4.4.2
<input type="checkbox"/> numba	Numpy aware dynamic python compiler using llvm	0.9.0
<input type="checkbox"/> numexpr	Fast numerical expression evaluator for numpy.	2.6.9
<input checked="" type="checkbox"/> numpy	Array processing for numbers, strings, records, and objects.	1.9.3
<input type="checkbox"/> numpy-base		1.9.3
<input type="checkbox"/> numpy-devel		1.9.3
<input type="checkbox"/> numpydoc	Sphinx extension to support docstrings in numpy format	0.9.1
<input type="checkbox"/> pytables	Brings together python, hdf5 and numpy to easily handle large amounts of data.	3.5.1
<input type="checkbox"/> snuggs	Snuggs are s-expressions for numpy.	1.4.3

15 packages available matching "numpy" 1 package selected

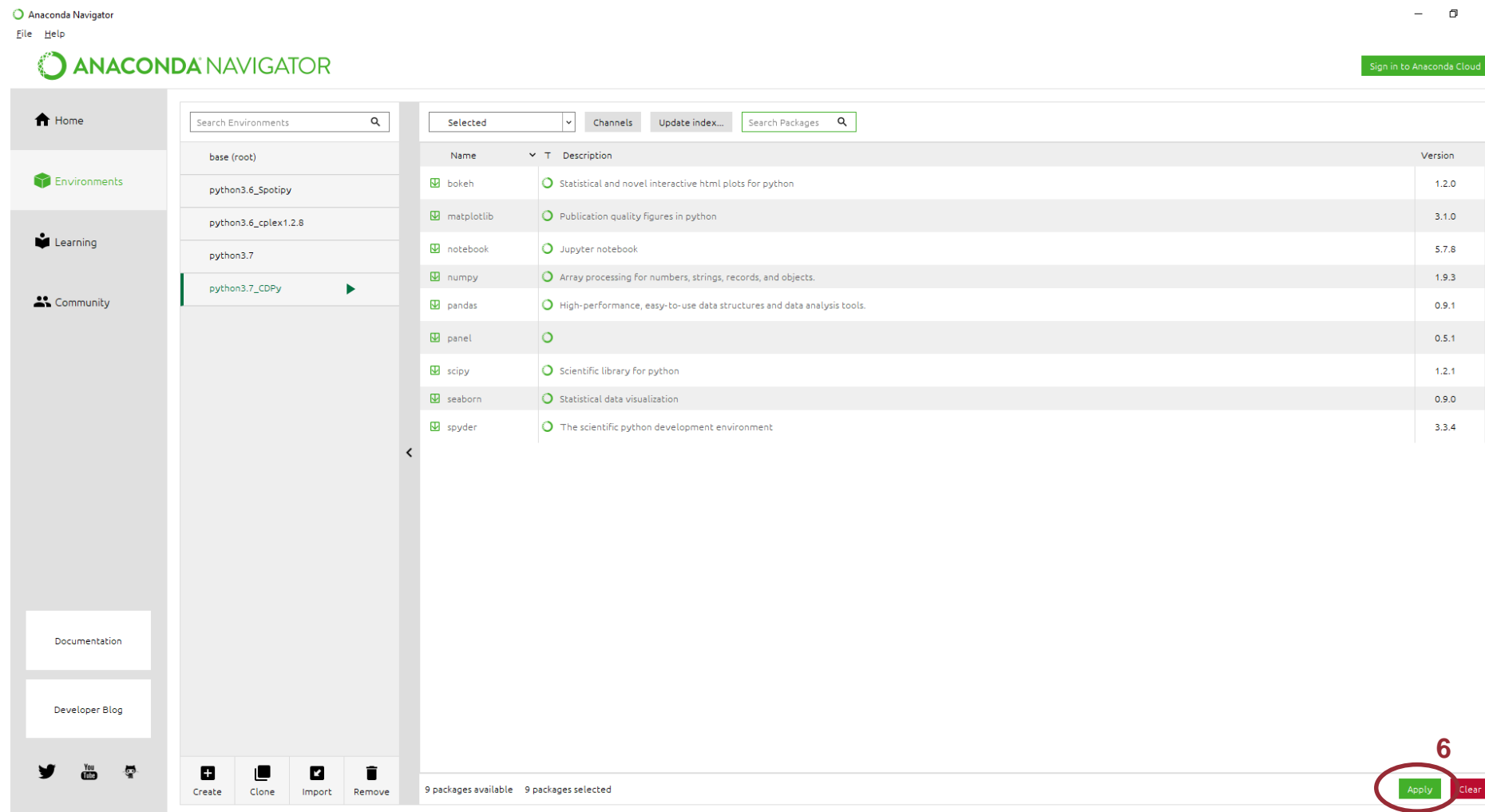
Apply Clear

Create Clone Import Remove

¿Cómo vamos a trabajar?

Librerías:

- bokeh
- matplotlib
- notebook
- numpy
- pandas
- scipy
- seaborn
- spyder



Anaconda Navigator

File Help

ANACONDA NAVIGATOR

Sign in to Anaconda Cloud

Home

Environments

Learning

Community

Documentation

Developer Blog

Search Environments

base (root)

python3.6_Spotipy

python3.6_cplex1.2.8

python3.7

python3.7_CDPy

Selected Channels Update index... Search Packages

Name	Description	Version
bokeh	Statistical and novel interactive html plots for python	1.2.0
matplotlib	Publication quality figures in python	3.1.0
notebook	Jupyter notebook	5.7.8
numpy	Array processing for numbers, strings, records, and objects.	1.9.3
pandas	High-performance, easy-to-use data structures and data analysis tools.	0.9.1
panel		0.5.1
scipy	Scientific library for python	1.2.1
seaborn	Statistical data visualization	0.9.0
spyder	The scientific python development environment	3.3.4

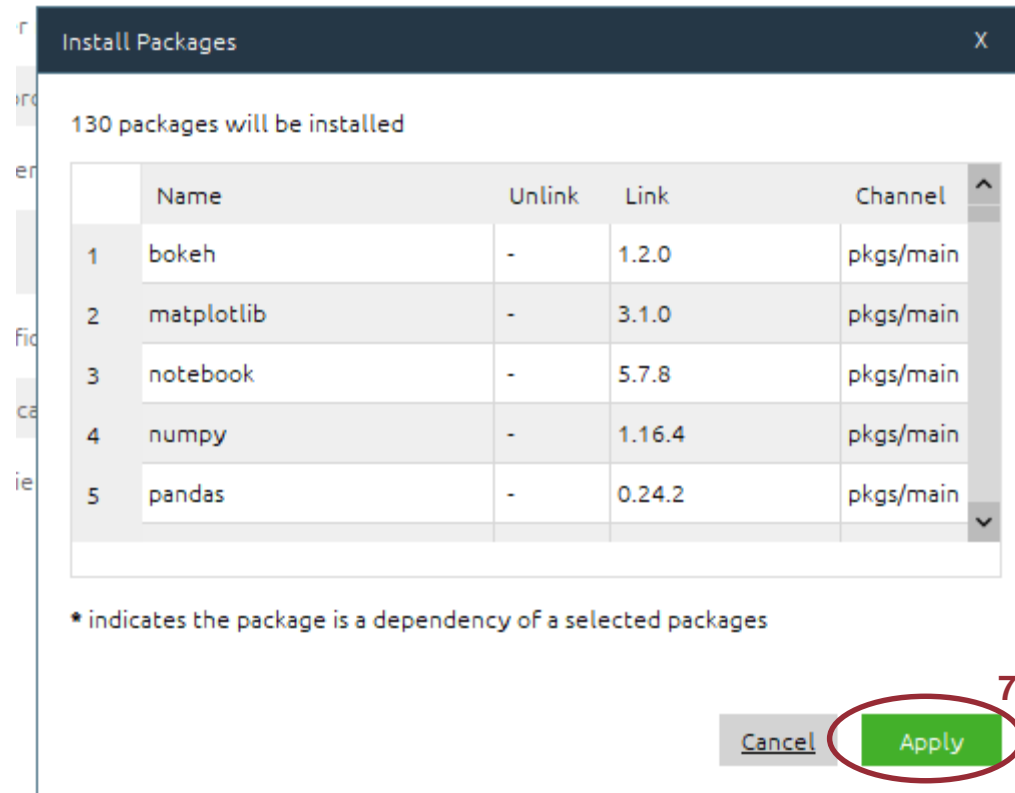
Create Clone Import Remove

9 packages available 9 packages selected

6

Apply Clear

¿Cómo vamos a trabajar?



¿Cómo vamos a trabajar?

Anaconda Navigator

File Help

ANACONDA NAVIGATOR

Sign in to Anaconda Cloud

Home

Environments

Learning

Community

Documentation

Developer Blog

Twitter YouTube GitHub

Create Clone Import Remove

Search Environments

base (root)

python3.6_Spotipy

python3.6_cplex1.2.8

python3.7

python3.7_CDPy

Installed Channels Update index... Search Packages

Name	T	Description	Version
✓ alabaster	○	Configurable, python 2+3 compatible sphinx theme.	0.7.12
✓ asn1crypto	○	Python asn.1 library with a focus on performance and a pythonic api	0.24.0
✓ astroid	○	A abstract syntax tree for python with inference support.	2.2.5
✓ attrs	○	Attrs is the python package that will bring back the joy of writing classes by relieving you from the drudgery of implementing object protocols (aka dunder methods).	19.1.0
✓ babel	○	Utilities to internationalize and localize python applications	2.6.0
✓ backcall	○	Specifications for callback functions passed in to an api	0.1.0
✓ blas	○		1.0
✓ bleach	○	Easy, whitelist-based html-sanitizing tool	3.1.0
✓ bokeh	○	Statistical and novel interactive html plots for python	1.2.0
✓ ca-certificates	○	Certificates for use with other packages.	2019.5.15
✓ certifi	○	Python package for providing mozilla's ca bundle.	2019.3.9
✓ cffi	○	Foreign Function interface for python calling c code.	1.12.3
✓ chardet	○	Universal character encoding detector	3.0.4
✓ cloudpickle	○	Extended pickling support for python objects	1.1.1
✓ colorama	○	Cross-platform colored terminal text.	0.4.1
✓ cryptography	○	Provides cryptographic recipes and primitives to python developers	2.7
✓ cycler	○	Composable style cycles.	0.10.0
✓ decorator	○	Better living through python with decorators.	4.4.0
✓ defusedxml	○	Xml bomb protection for python stdlib modules	0.6.0
✓ docutils	○	Docutils -- python documentation utilities	0.14

141 packages available

¿Cómo vamos a trabajar?

Anaconda Navigator

File Help

ANACONDA NAVIGATOR

Sign in to Anaconda Cloud

Home

Environments

Learning

Community

Documentation

Developer Blog

Search Environments

base (root)

python3.6_Spotipy

python3.6_cplex1.2.8

python3.7

python3.7_CDPy

Open Terminal

Open with Python

Open with IPython

Open with Jupyter Notebook

Installed

Channels

Update index...

Search Packages

Name	T	Description	Version
✓ alabaster	○	Configurable, python 2+3 compatible sphinx theme.	0.7.12
✓ asn1crypto	○	Python asn.1 library with a focus on performance and a pythonic api	0.24.0
✓ astroid	○	A abstract syntax tree for python with inference support.	2.2.5
✓ attrs	○	Attrs is the python package that will bring back the joy of writing classes by relieving you from the drudgery of implementing object protocols (aka dunder methods).	19.1.0
✓ babel	○	Utilities to internationalize and localize python applications	2.6.0
✓ blinker	○	Specifications for callback functions passed in to an api	0.1.0
✓ bleach	○	Easy, whitelist-based html-sanitizing tool	1.0
✓ bokeh	○	Statistical and novel interactive html plots for python	3.1.0
✓ ca-certificates	○	Certificates for use with other packages.	1.2.0
✓ certifi	○	Python package for providing mozilla's ca bundle.	2019.5.15
✓ cffi	○	Foreign Function interface for python calling c code.	2019.3.9
✓ chardet	○	Universal character encoding detector	1.12.3
✓ cloudpickle	○	Extended pickling support for python objects	3.0.4
✓ colorama	○	Extended pickling support for python objects	1.1.1
✓ cryptography	○	Cross-platform colored terminal text.	0.4.1
✓ cyclr	○	Provides cryptographic recipes and primitives to python developers	0.7
✓ decorator	○	Composable style cycles.	0.10.0
✓ defusedxml	○	Better living through python with decorators.	4.4.0
✓ docutils	○	Xml bomb protection for python stdlib modules	0.6.0
✓ docutils	○	Docutils -- python documentation utilities	0.14

141 packages available

Create Clone Import Remove

fcarazo / introduccion-a-python

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

master 2 branches 0 tags

Go to file

Add file

Code



fcarazo FC docs added

e76ca45 13 minutes ago 3 commits

class/MODULE_1-python_introduction	FC: Add module 1 of the introduction to python course	5 hours ago
docs-public	FC docs added	13 minutes ago
.gitignore	FC initial git	3 hours ago
LICENSE	FC initial git	3 hours ago
README.md	FC initial git	3 hours ago

README.md



Introduccion-a-Python

Curso de Introducción a la Programación en Python - Impartido en TECNUN Universidad de Navarra

¿Cómo vamos a trabajar?



- Aplicación web “open-source” para Python.
- Permite crear documentos que contienen código ejecutable.
- Emplearemos Jupyter Notebooks en las clases más teóricas.

¿Cómo vamos a trabajar?

En el caso de querer ubicar la carpeta inicial de jupyter en otra dirección que no sea la predeterminada (P:/, por ejemplo), **tendremos que seguir estos pasos cada vez que iniciemos sesión.**

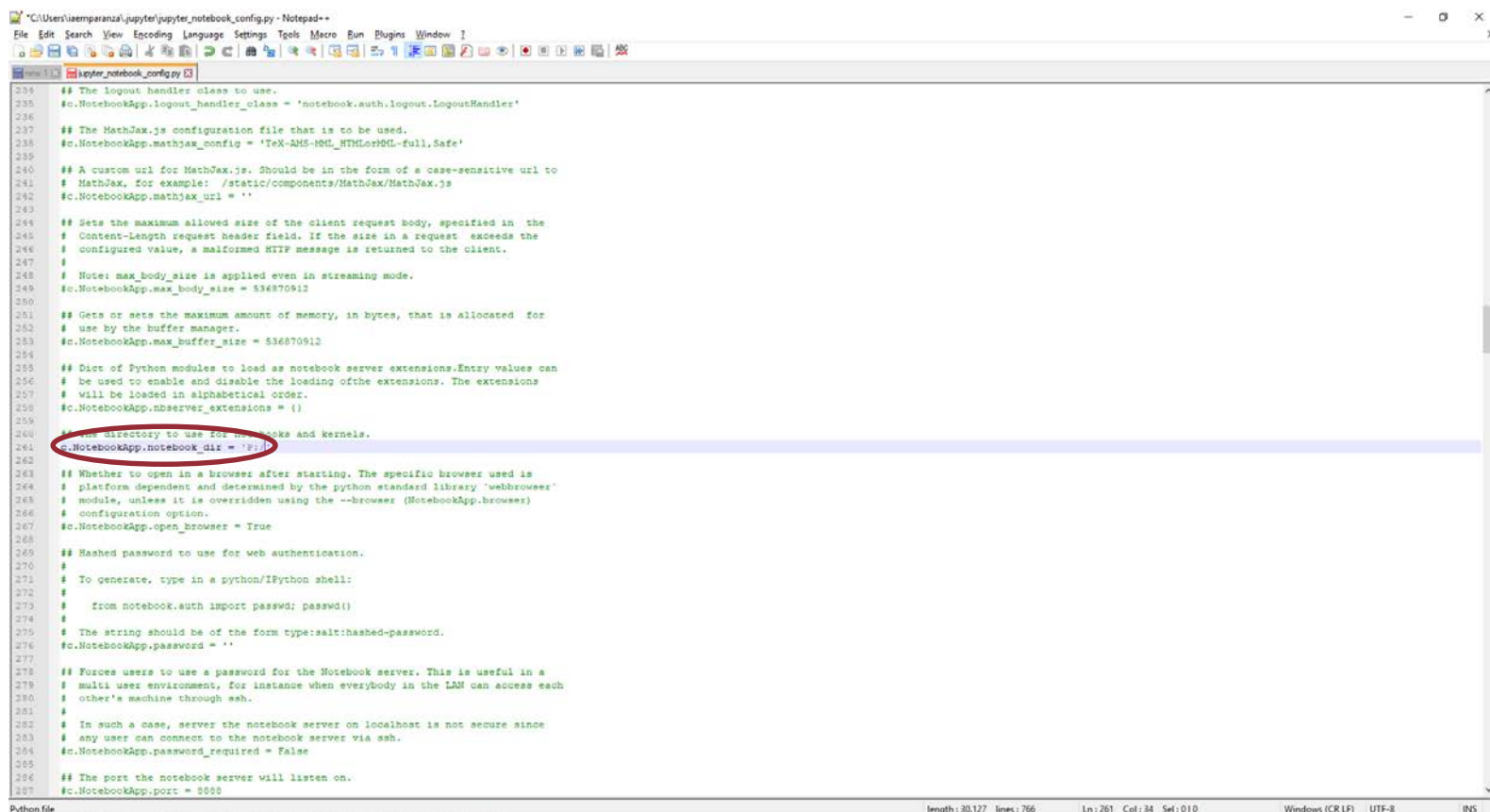
- Desde cualquier entorno de Anaconda abrimos Jupyter Notebook primero y la terminal después.
- Escribimos el siguiente comando en la terminal:

jupyter notebook --generate-config

- Se habrá creado un archivo titulado **jupyter_notebook_config.py** en la dirección que nos diga la terminal.
- Abrimos ese archivo con Notepad.
- En el campo `c.NotebookApp.notebook_dir` borramos la `#` del comienzo de la línea y escribimos:

`c.NotebookApp.notebook_dir = 'P:/'`

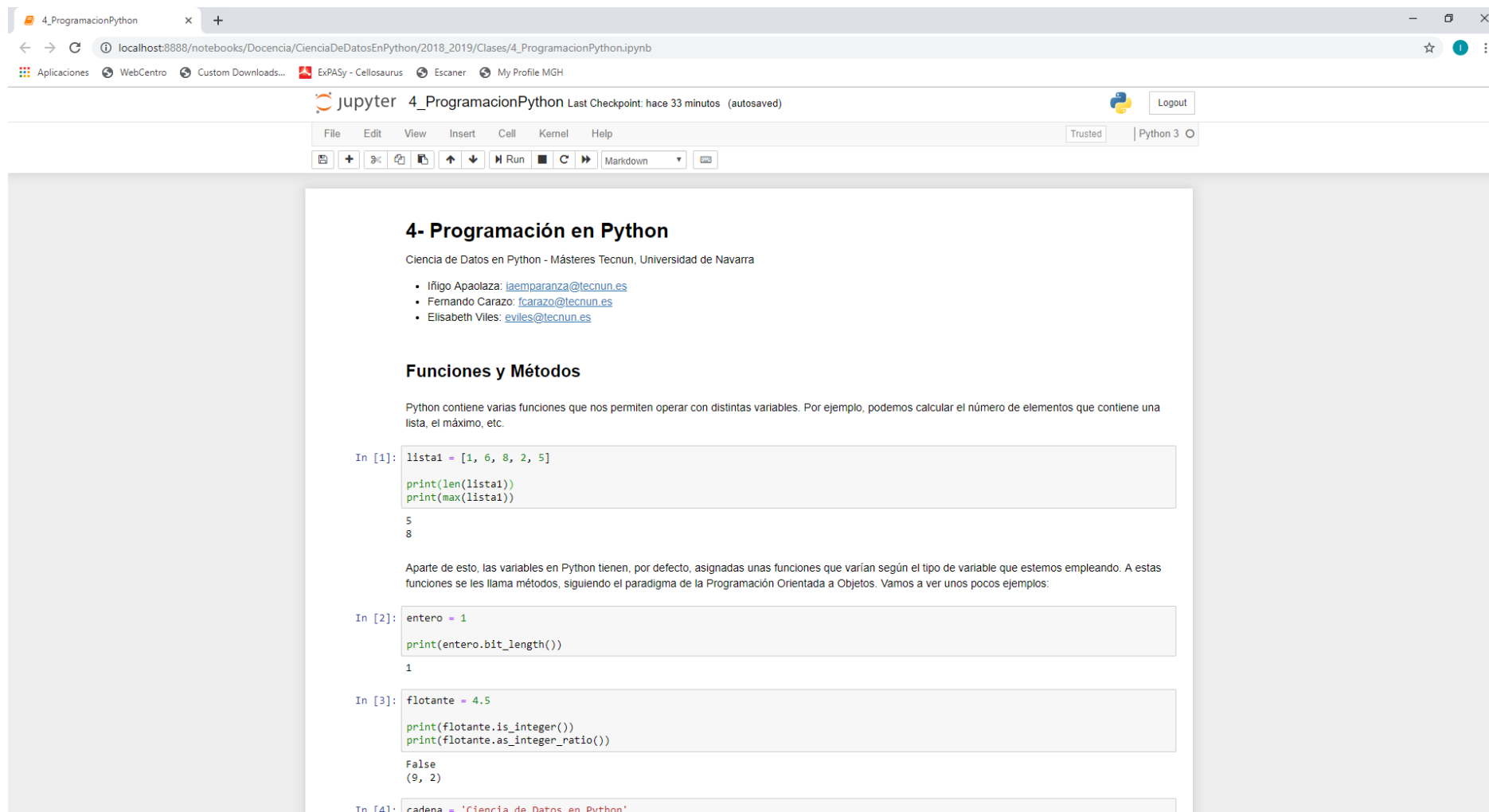
- Guardamos el archivo y cerramos Jupyter Notebook.



```
234  ## The logout handler class to use.
235  #c.NotebookApp.logout_handler_class = 'notebook.auth.logout.LogoutHandler'
236
237  ## The MathJax.js configuration file that is to be used.
238  #c.NotebookApp.mathjax_config = 'TeX-AMS-MML_HTMLorMML-full,Safe'
239
240  ## A custom url for MathJax.js. Should be in the form of a case-sensitive url to
241  ## MathJax, for example: /static/components/MathJax/MathJax.js
242  #c.NotebookApp.mathjax_url = ''
243
244  ## Sets the maximum allowed size of the client request body, specified in the
245  ## Content-Length request header field. If the size in a request exceeds the
246  ## configured value, a malformed HTTP message is returned to the client.
247  ##
248  ## Note: max_body_size is applied even in streaming mode.
249  #c.NotebookApp.max_body_size = 536870912
250
251  ## Gets or sets the maximum amount of memory, in bytes, that is allocated for
252  ## use by the buffer manager.
253  #c.NotebookApp.max_buffer_size = 536870912
254
255  ## Dict of Python modules to load as notebook server extensions. Entry values can
256  ## be used to enable and disable the loading of the extensions. The extensions
257  ## will be loaded in alphabetical order.
258  #c.NotebookApp.nbserver_extensions = {}
259
260  ## The directory to use for notebooks and kernels.
261  c.NotebookApp.notebook_dir = 'P:/'
262
263  ## Whether to open a browser after starting. The specific browser used is
264  ## platform dependent and determined by the python standard library 'webbrowser'
265  ## module, unless it is overridden using the --browser (NotebookApp.browser)
266  ## configuration option.
267  #c.NotebookApp.open_browser = True
268
269  ## Hashed password to use for web authentication.
270  ##
271  ## To generate, type in a python/ipython shell:
272  ##
273  ## from notebook.auth import passwd; passwd()
274  ##
275  ## The string should be of the form type:salt:hashed-password.
276  #c.NotebookApp.password = ''
277
278  ## Forces users to use a password for the Notebook server. This is useful in a
279  ## multi user environment, for instance when everybody in the LAN can access each
280  ## other's machine through ssh.
281  ##
282  ## In such a case, server the notebook server on localhost is not secure since
283  ## any user can connect to the notebook server via ssh.
284  #c.NotebookApp.password_required = False
285
286  ## The port the notebook server will listen on.
287  #c.NotebookApp.port = 8888
```



¿Cómo vamos a trabajar?



The screenshot shows a Jupyter Notebook interface in a web browser. The browser tab is titled '4_ProgramacionPython'. The address bar shows the URL: `localhost:8888/notebooks/Docencia/CienciaDeDatosEnPython/2018_2019/Clases/4_ProgramacionPython.ipynb`. The Jupyter interface includes a top bar with the Jupyter logo, the notebook title '4_ProgramacionPython', and a 'Logout' button. Below this is a menu bar with options: File, Edit, View, Insert, Cell, Kernel, and Help. A toolbar contains icons for file operations, running, and other notebook functions. The main content area displays the notebook's content, which includes a title, a subtitle, a list of authors, and several code cells with Python code and their outputs.

4- Programación en Python

Ciencia de Datos en Python - Másteres Tecnun, Universidad de Navarra

- Iñigo Apaolaza: iaemparanza@tecnun.es
- Fernando Carazo: fcarazo@tecnun.es
- Elisabeth Viles: eviles@tecnun.es

Funciones y Métodos

Python contiene varias funciones que nos permiten operar con distintas variables. Por ejemplo, podemos calcular el número de elementos que contiene una lista, el máximo, etc.

```
In [1]: lista1 = [1, 6, 8, 2, 5]
        print(len(lista1))
        print(max(lista1))
```

5
8

Aparte de esto, las variables en Python tienen, por defecto, asignadas unas funciones que varían según el tipo de variable que estemos empleando. A estas funciones se les llama métodos, siguiendo el paradigma de la Programación Orientada a Objetos. Vamos a ver unos pocos ejemplos:

```
In [2]: entero = 1
        print(entero.bit_length())
```

1

```
In [3]: flotante = 4.5
        print(flotante.is_integer())
        print(flotante.as_integer_ratio())
```

False
(9, 2)

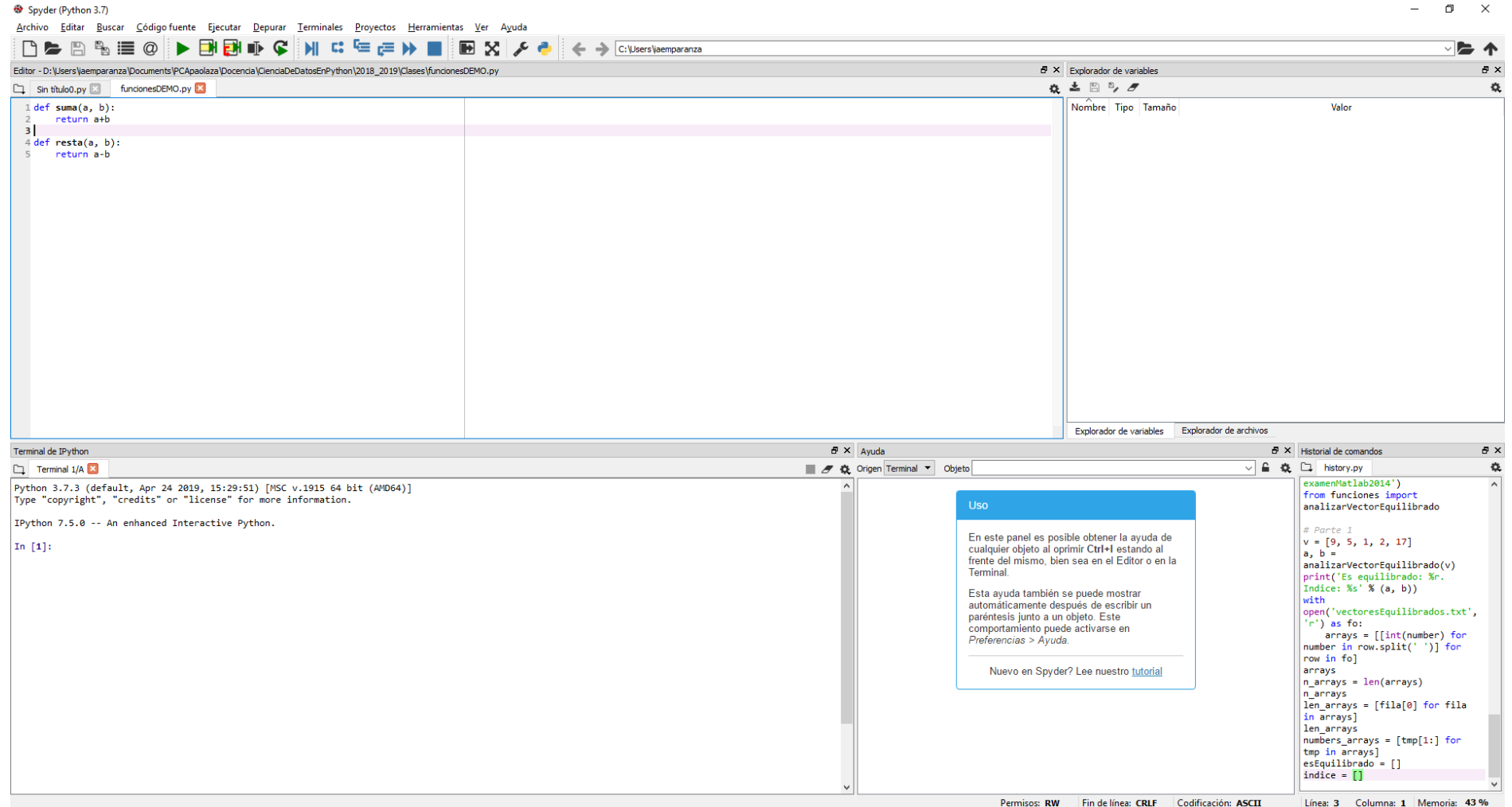
```
In [4]: cadena = 'Ciencia de Datos en Python'
```

¿Cómo vamos a trabajar?



- **Scientific Python Development EnviRonment.**
- Entorno de desarrollo interactivo similar a RStudio.
- Técnicamente: “entorno de desarrollo integrado y multiplataforma de código abierto (IDE)”

¿Cómo vamos a trabajar?





Introducción a Python

Ciencia de Datos en Python

San Sebastián – 25 de enero de 2021