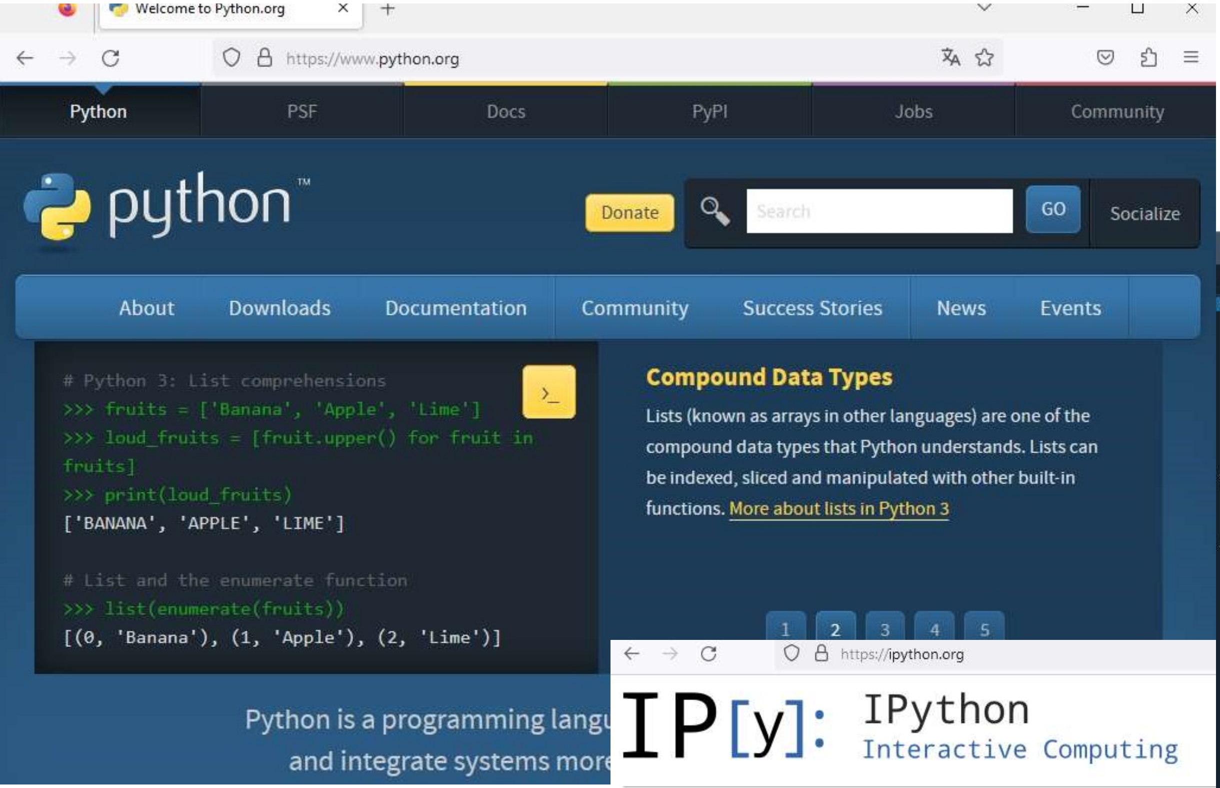
ntic master
revolucionamos la comunicación

UNIVERSIDAD COMPLUTENSE DE MADRID

Programación. Python

Las herramientas

Cristóbal Pareja Flores



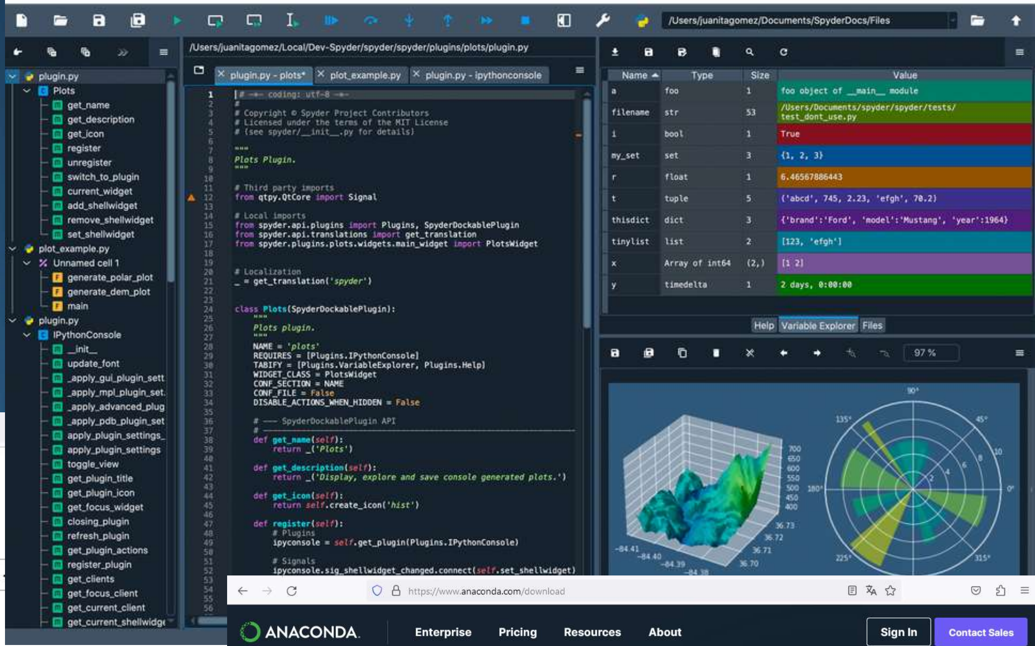
```
# Python 3: List comprehensions
>>> fruits = ['Banana', 'Apple', 'Lime']
>>> loud_fruits = [fruit.upper() for fruit in
fruits]
>>> print(loud_fruits)
['BANANA', 'APPLE', 'LIME']

# List and the enumerate function
>>> list(enumerate(fruits))
[(0, 'Banana'), (1, 'Apple'), (2, 'Lime')]
```

Compound Data Types

Lists (known as arrays in other languages) are one of the compound data types that Python understands. Lists can be indexed, sliced and manipulated with other built-in functions. More about lists in Python 3

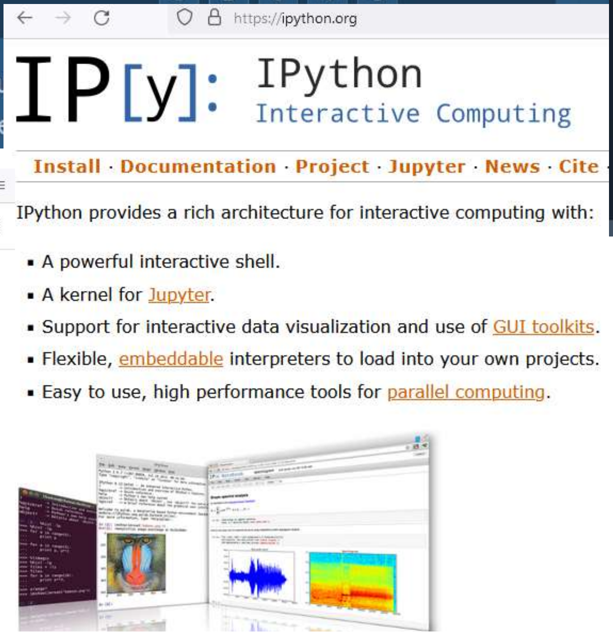
Python is a programming langu
and integrate systems more

spyder | The Scientific Python Development Environment



```
# -*- coding: utf-8 -*-
#
# Copyright © Spyder Project Contributors
# Licensed under the terms of the MIT License
# (see spyder/__init__.py for details)

"""
Plots Plugin.
"""

# Third party imports
from qtpy.QtCore import Signal

# Local imports
from spyder.api.plugins import Plugins, SpyderDockablePlugin
from spyder.api.translations import get_translation
from spyder.plugins.plots.widgets.main_widget import PlotsWidget


# Localization
_ = get_translation('spyder')


class Plots(SpyderDockablePlugin):
    """
    Plots plugin.
    """
    NAME = 'plots'
    REQUIRES = [Plugins.IPythonConsole]
    TABIFY = [Plugins.VariableExplorer, Plugins.Help]
    WIDGET_CLASS = PlotsWidget
    CONF_SECTION = NAME
    CONF_FILE = False
    DISABLE_ACTIONS_WHEN_HIDDEN = False

    # ---- SpyderDockablePlugin API
    # ------------------------------------------------------------------
    def get_name(self):
        return _('Plots')

    def get_description(self):
        return _('Display, explore and save console generated plots.')

    def get_icon(self):
        return self.create_icon('hist')

    def register(self):
        # Plugins
        ipyconsole = self.get_plugin(Plugins.IPythonConsole)

        # Signals
        ipyconsole.sig_shellwidget_changed.connect(self.set_shellwidget)
```

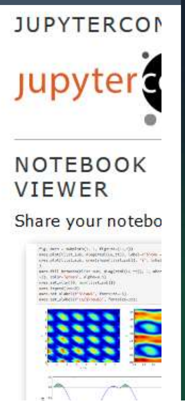
Name	Type	Size	Value
a	foo	1	foo object of __main__ module
filename	str	53	/Users/Documents/spyder/spyder/tests/test_dont_use.py
i	bool	1	True
my_set	set	3	{1, 2, 3}
r	float	1	6.46567886443
t	tuple	5	('abcd', 745, 2.23, 'efgh', 70.2)
thisdict	dict	3	{'brand':'Ford', 'model':'Mustang', 'year':1964}
tinylist	list	2	[123, 'efgh']
x	Array of int64	(2,)	[1 2]
y	timedelta	1	2 days, 0:00:00

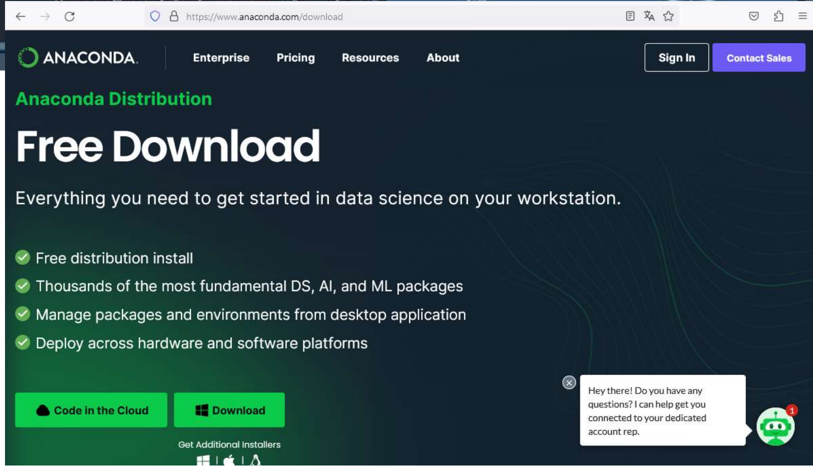


IPython provides a rich architecture for interactive computing with:

- A powerful interactive shell.
- A kernel for Jupyter.
- Support for interactive data visualization and use of GUI toolkits.
- Flexible, embeddable interpreters to load into your own projects.
- Easy to use, high performance tools for parallel computing.

Software gratuito, estándares abiertos y servicios web para la computación interactiva en todos los lenguajes de programación





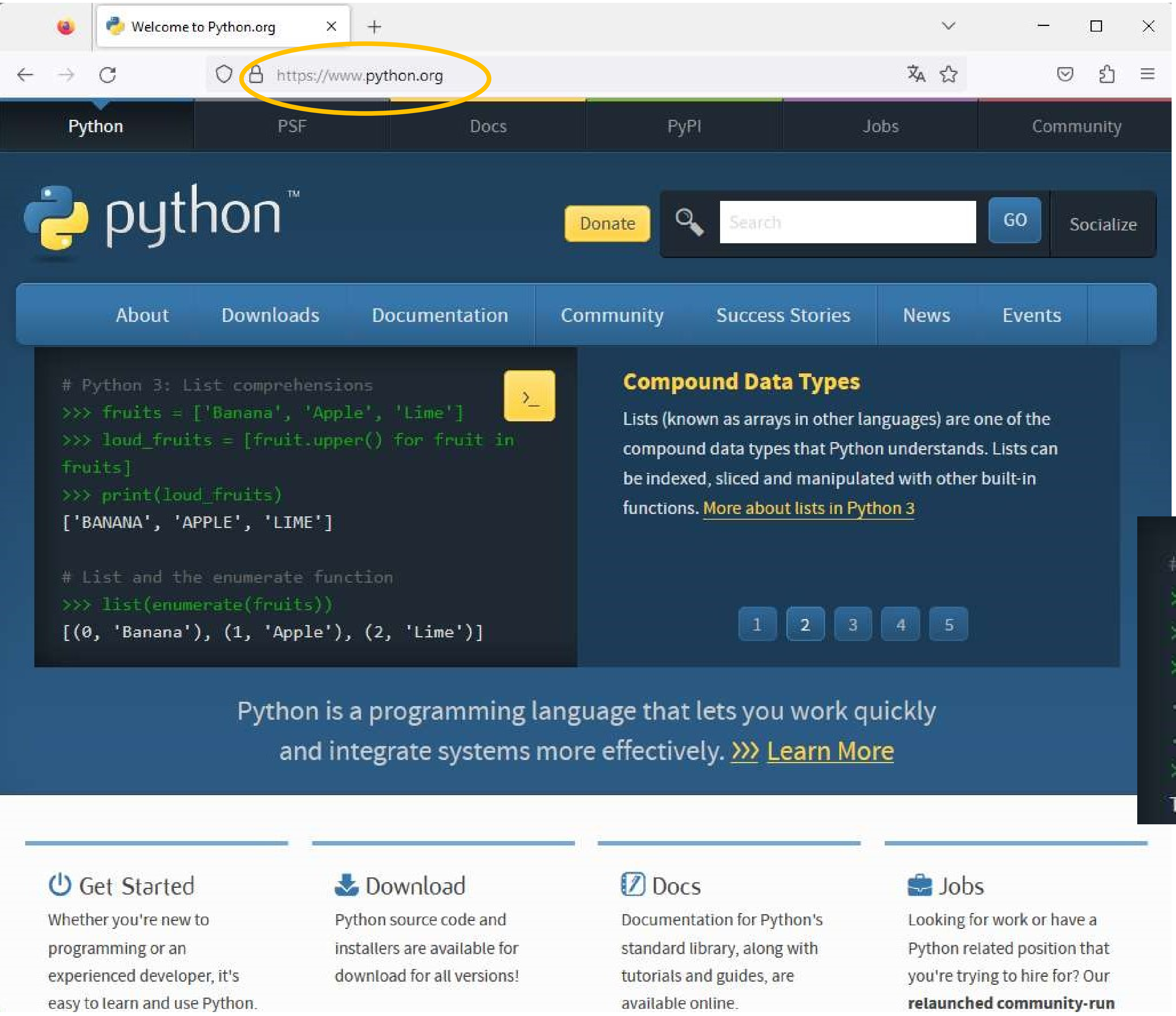
Anaconda Distribution

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

Hey there! Do you have any questions? I can help get you connected to your dedicated account rep.



```
# Python 3: List comprehensions
>>> fruits = ['Banana', 'Apple', 'Lime']
>>> loud_fruits = [fruit.upper() for fruit in
fruits]
>>> print(loud_fruits)
['BANANA', 'APPLE', 'LIME']

# List and the enumerate function
>>> list(enumerate(fruits))
[(0, 'Banana'), (1, 'Apple'), (2, 'Lime')]
```

Compound Data Types

Lists (known as arrays in other languages) are one of the compound data types that Python understands. Lists can be indexed, sliced and manipulated with other built-in functions. More about lists in Python 3

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> Learn More

Get Started

Whether you're new to programming or an experienced developer, it's easy to learn and use Python.

Download

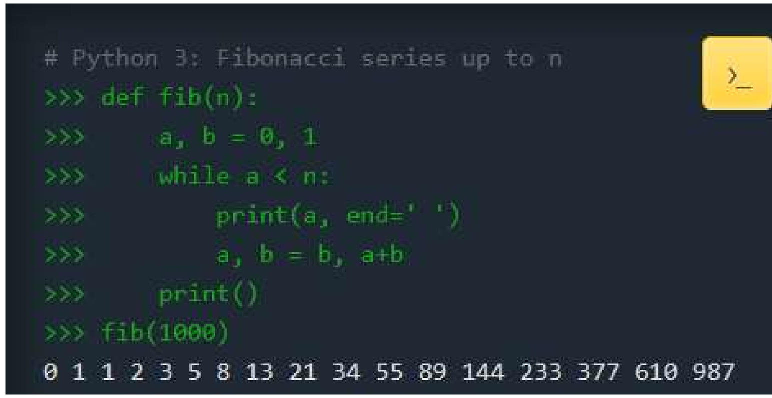
Python source code and installers are available for download for all versions!

Docs

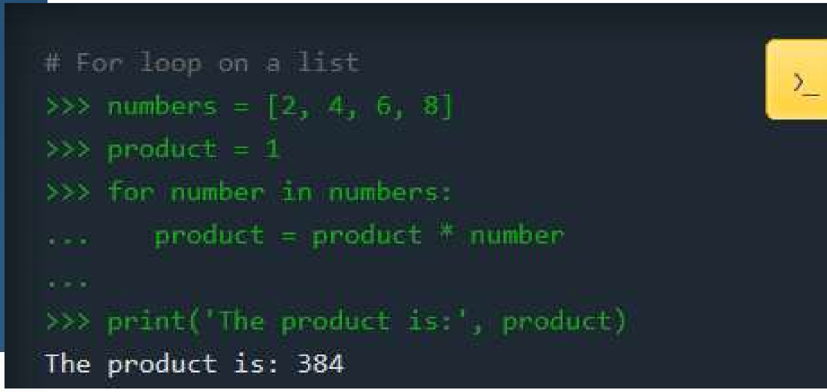
Documentation for Python's standard library, along with tutorials and guides, are available online.

Jobs

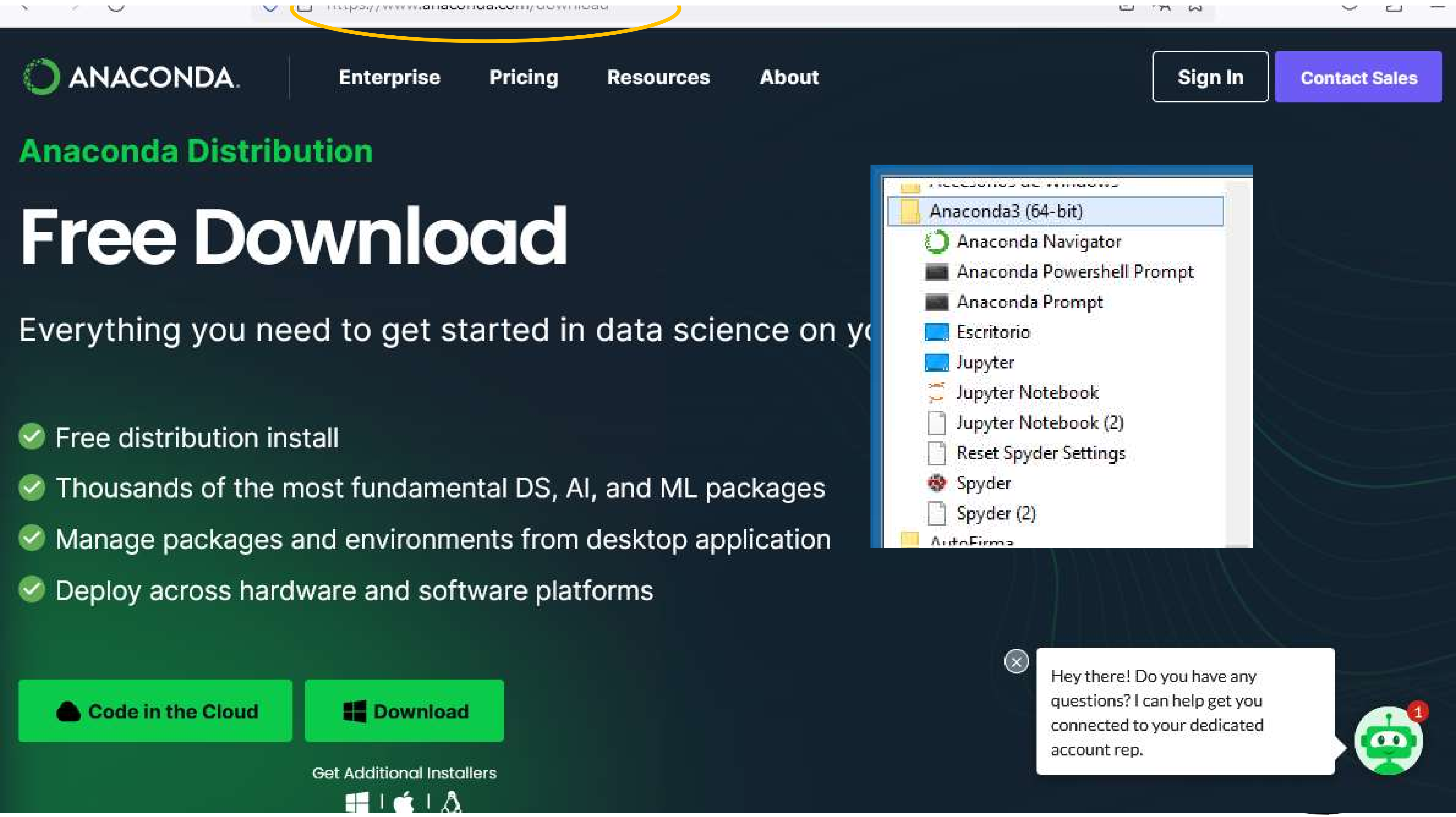
Looking for work or have a Python related position that you're trying to hire for? Our **relaunched community-run**



```
# Python 3: Fibonacci series up to n
>>> def fib(n):
>>>     a, b = 0, 1
>>>     while a < n:
>>>         print(a, end=' ')
>>>         a, b = b, a+b
>>>     print()
>>> fib(1000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987
```



```
# For loop on a list
>>> numbers = [2, 4, 6, 8]
>>> product = 1
>>> for number in numbers:
...     product = product * number
...
>>> print('The product is:', product)
The product is: 384
```



ANACONDA

Enterprise Pricing Resources About

Sign In Contact Sales

Anaconda Distribution

Free Download

Everything you need to get started in data science on yo

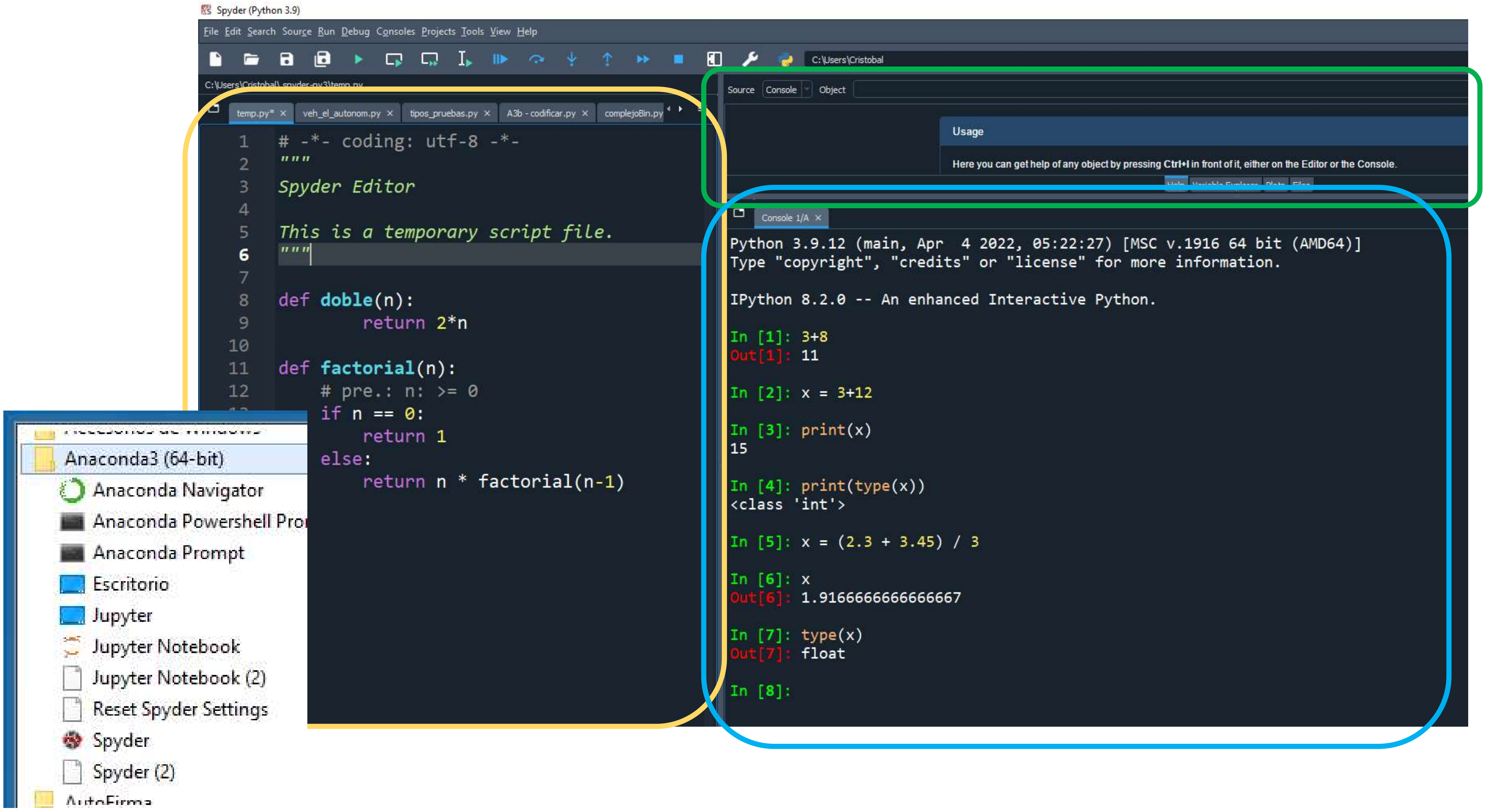
- Free distribution install
- Thousands of the most fundamental DS, AI, and ML packages
- Manage packages and environments from desktop application
- Deploy across hardware and software platforms

Code in the Cloud Download

Get Additional Installers

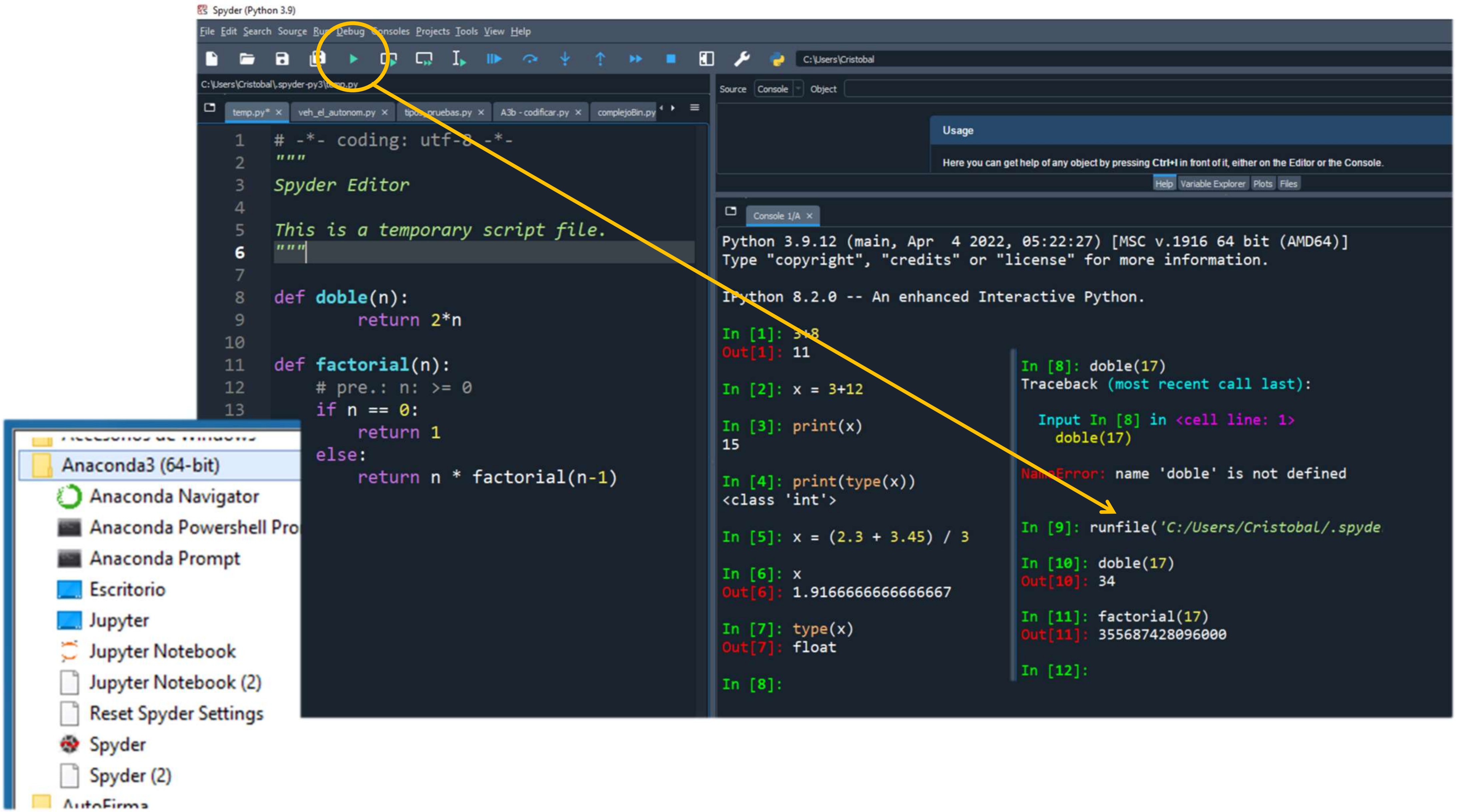
Anaconda3 (64-bit)
- Anaconda Navigator
- Anaconda Powershell Prompt
- Anaconda Prompt
- Escritorio
- Jupyter
- Jupyter Notebook
- Jupyter Notebook (2)
- Reset Spyder Settings
- Spyder
- Spyder (2)

Hey there! Do you have any questions? I can help get you connected to your dedicated account rep.



```python
# -*- coding: utf-8 -*-
"""
Spyder Editor

This is a temporary script file.
"""

def doble(n):
        return 2*n

def factorial(n):
    # pre.: n: >= 0
    if n == 0:
        return 1
    else:
        return n * factorial(n-1)
```

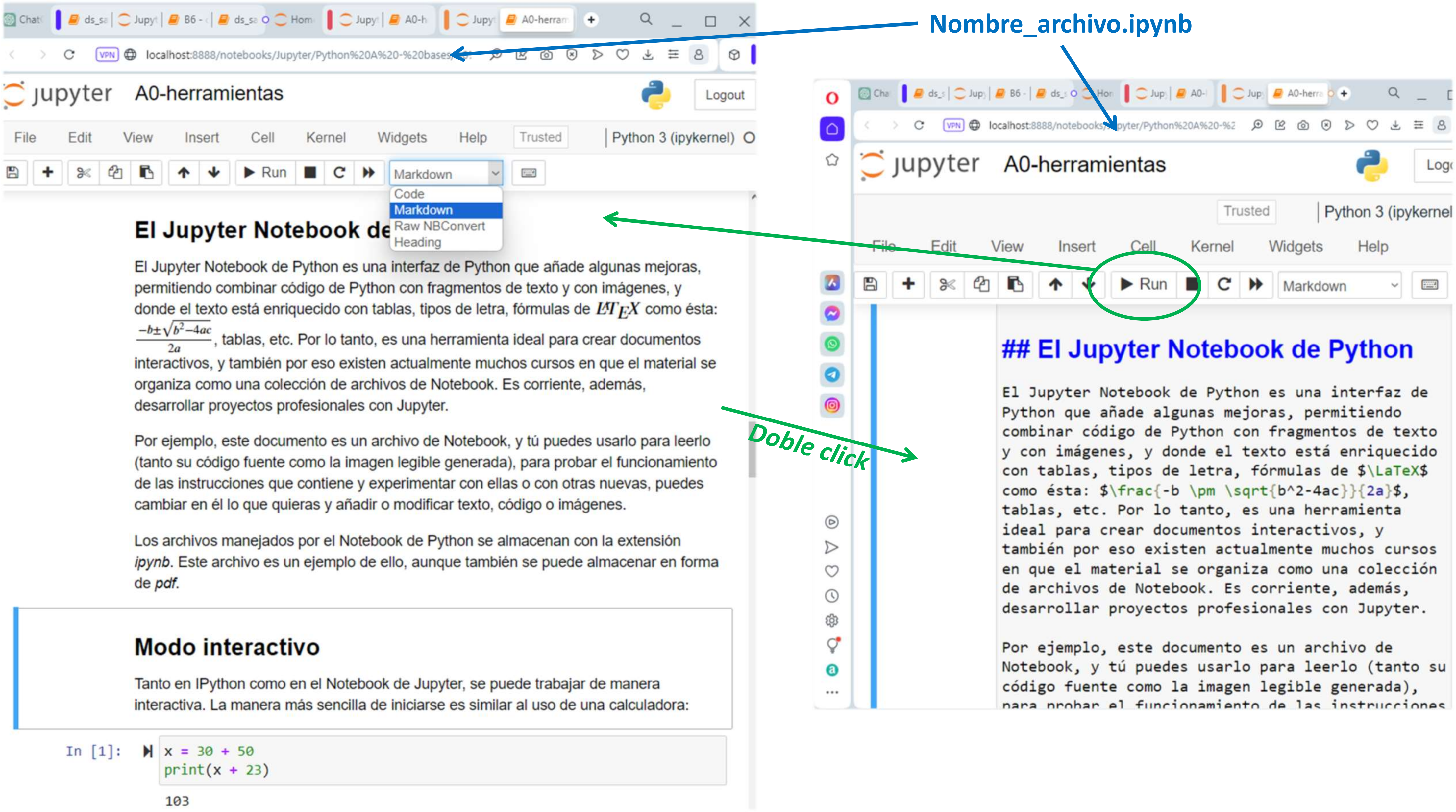
```
Python 3.9.12 (main, Apr  4 2022, 05:22:27) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.2.0 -- An enhanced Interactive Python.

In [1]: 3+8
Out[1]: 11

In [2]: x = 3+12

In [3]: print(x)
15

In [4]: print(type(x))
<class 'int'>

In [5]: x = (2.3 + 3.45) / 3

In [6]: x
Out[6]: 1.9166666666666667

In [7]: type(x)
Out[7]: float

In [8]:
```


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Cristobal
C:\Users\Cristobal\.spyder-py3\temp.py
temp.py* veh_el_autonom.py tipos_pruebas.py A3b - codificar.py complejoBin.py
Source Console Object
Usage
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.
Help Variable Explorer Plots Files
Console 1/A
Anaconda3 (64-bit)
Anaconda Navigator
Anaconda Powershell Pro
Anaconda Prompt
Escritorio
Jupyter
Jupyter Notebook
Jupyter Notebook (2)
Reset Spyder Settings
Spyder
Spyder (2)

```python
# -*- coding: utf-8 -*-
"""
Spyder Editor

This is a temporary script file.
"""

def doble(n):
        return 2*n

def factorial(n):
    # pre.: n: >= 0
    if n == 0:
        return 1
    else:
        return n * factorial(n-1)
```

```
Python 3.9.12 (main, Apr  4 2022, 05:22:27) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.2.0 -- An enhanced Interactive Python.

In [1]: 3+8
Out[1]: 11

In [2]: x = 3+12

In [3]: print(x)
15

In [4]: print(type(x))
<class 'int'>

In [5]: x = (2.3 + 3.45) / 3

In [6]: x
Out[6]: 1.9166666666666667

In [7]: type(x)
Out[7]: float

In [8]:
```

```
In [8]: doble(17)
Traceback (most recent call last):

  Input In [8] in <cell line: 1>
    doble(17)

NameError: name 'doble' is not defined


In [9]: runfile('C:/Users/Cristobal/.spyde

In [10]: doble(17)
Out[10]: 34

In [11]: factorial(17)
Out[11]: 355687428096000

In [12]:
```


Nombre_archivo.ipynb
Doble click
localhost:8888/notebooks/Jupyter/Python%20A%20-%20bases
jupyter A0-herramientas
Logout
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)
Run
Markdown
Code
Markdown
Raw NBConvert
Heading

El Jupyter Notebook de

El Jupyter Notebook de Python es una interfaz de Python que añade algunas mejoras, permitiendo combinar código de Python con fragmentos de texto y con imágenes, y donde el texto está enriquecido con tablas, tipos de letra, fórmulas de LaTeX como ésta: $\frac{-b \pm \sqrt{b^2-4ac}}{2a}$, tablas, etc. Por lo tanto, es una herramienta ideal para crear documentos interactivos, y también por eso existen actualmente muchos cursos en que el material se organiza como una colección de archivos de Notebook. Es corriente, además, desarrollar proyectos profesionales con Jupyter.

Por ejemplo, este documento es un archivo de Notebook, y tú puedes usarlo para leerlo (tanto su código fuente como la imagen legible generada), para probar el funcionamiento de las instrucciones que contiene y experimentar con ellas o con otras nuevas, puedes cambiar en él lo que quieras y añadir o modificar texto, código o imágenes.

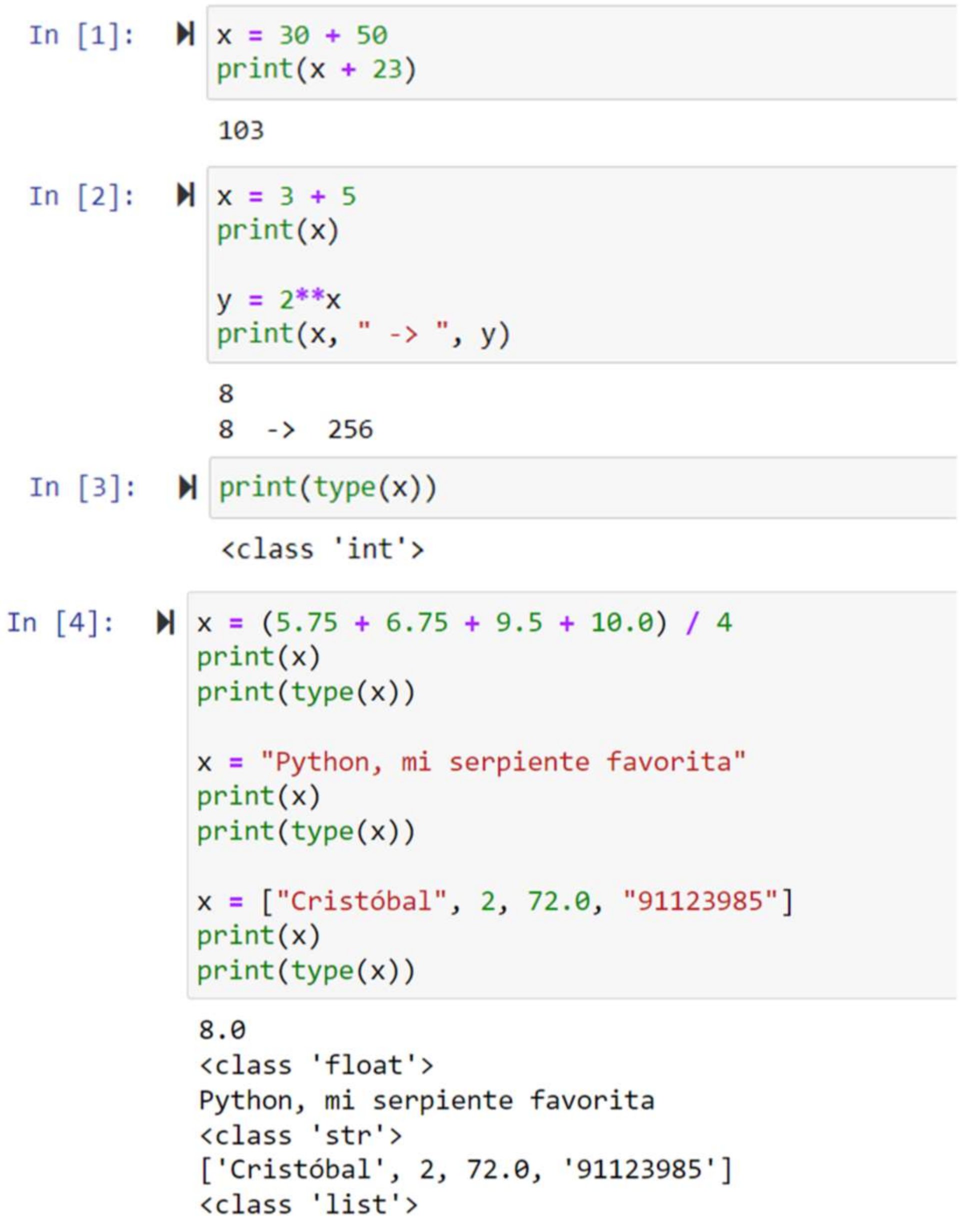
Los archivos manejados por el Notebook de Python se almacenan con la extensión *ipynb*. Este archivo es un ejemplo de ello, aunque también se puede almacenar en forma de *pdf*.

Modo interactivo

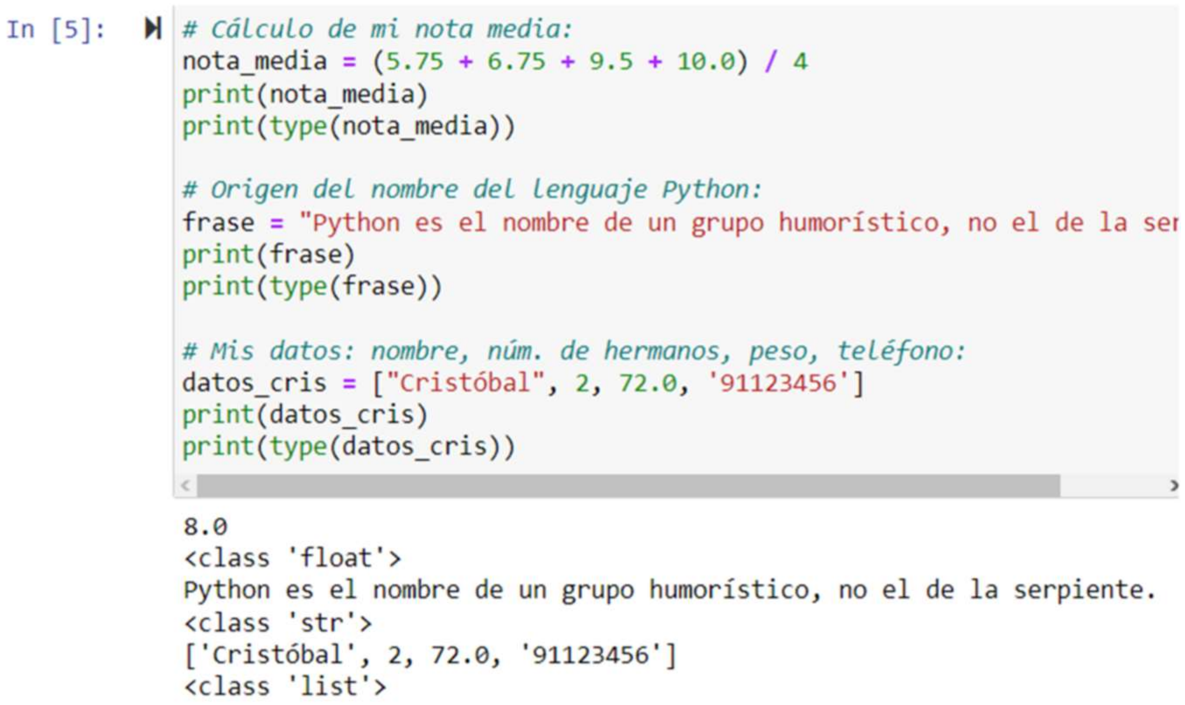
Tanto en IPython como en el Notebook de Jupyter, se puede trabajar de manera interactiva. La manera más sencilla de iniciarse es similar al uso de una calculadora:

```
In [1]:  x = 30 + 50
         print(x + 23)

         103
```

```
## El Jupyter Notebook de Python

El Jupyter Notebook de Python es una interfaz de
Python que añade algunas mejoras, permitiendo
combinar código de Python con fragmentos de texto
y con imágenes, y donde el texto está enriquecido
con tablas, tipos de letra, fórmulas de $\LaTeX$
como ésta: $\frac{-b \pm \sqrt{b^2-4ac}}{2a}$,
tablas, etc. Por lo tanto, es una herramienta
ideal para crear documentos interactivos, y
también por eso existen actualmente muchos cursos
en que el material se organiza como una colección
de archivos de Notebook. Es corriente, además,
desarrollar proyectos profesionales con Jupyter.

Por ejemplo, este documento es un archivo de
Notebook, y tú puedes usarlo para leerlo (tanto su
código fuente como la imagen legible generada),
```

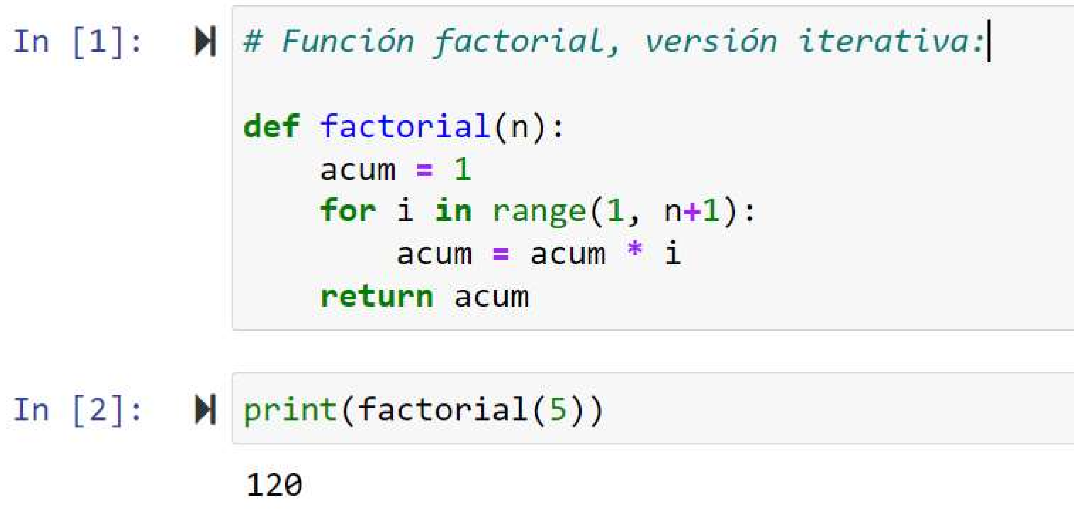


```
In [1]:  x = 30 + 50
         print(x + 23)

         103

In [2]:  x = 3 + 5
         print(x)

         y = 2**x
         print(x, " -> ", y)

         8
         8  ->  256

In [3]:  print(type(x))

         <class 'int'>

In [4]:  x = (5.75 + 6.75 + 9.5 + 10.0) / 4
         print(x)
         print(type(x))

         x = "Python, mi serpiente favorita"
         print(x)
         print(type(x))

         x = ["Cristóbal", 2, 72.0, "91123985"]
         print(x)
         print(type(x))

         8.0
         <class 'float'>
         Python, mi serpiente favorita
         <class 'str'>
         ['Cristóbal', 2, 72.0, '91123985']
         <class 'list'>
```

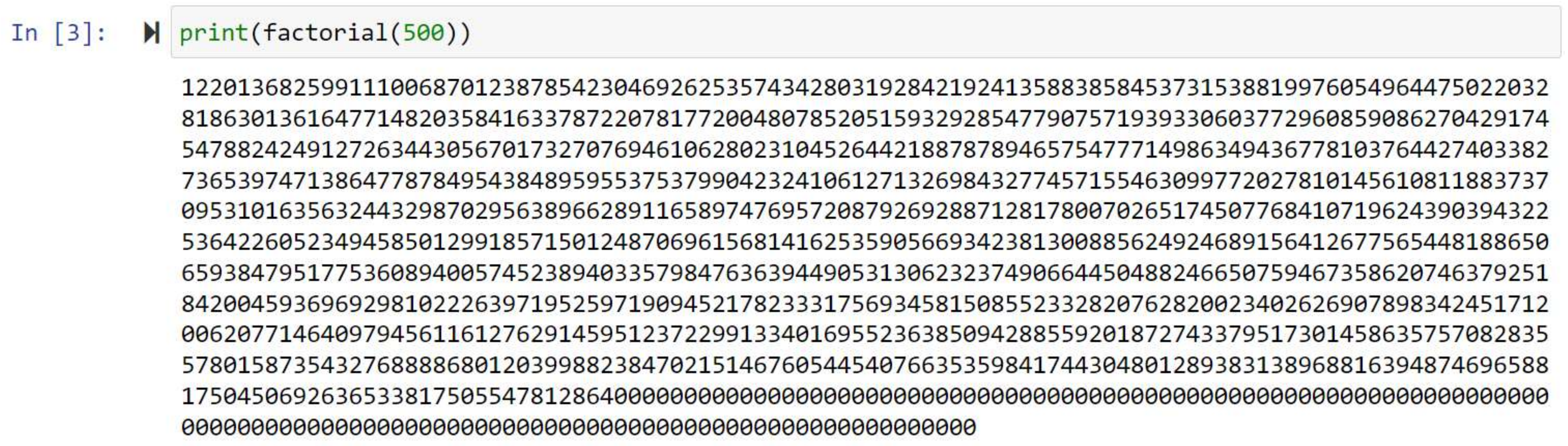


```
In [5]:  # Cálculo de mi nota media:
         nota_media = (5.75 + 6.75 + 9.5 + 10.0) / 4
         print(nota_media)
         print(type(nota_media))

         # Origen del nombre del lenguaje Python:
         frase = "Python es el nombre de un grupo humorístico, no el de la ser
         print(frase)
         print(type(frase))

         # Mis datos: nombre, núm. de hermanos, peso, teléfono:
         datos_cris = ["Cristóbal", 2, 72.0, '91123456']
         print(datos_cris)
         print(type(datos_cris))

         8.0
         <class 'float'>
         Python es el nombre de un grupo humorístico, no el de la serpiente.
         <class 'str'>
         ['Cristóbal', 2, 72.0, '91123456']
         <class 'list'>
```

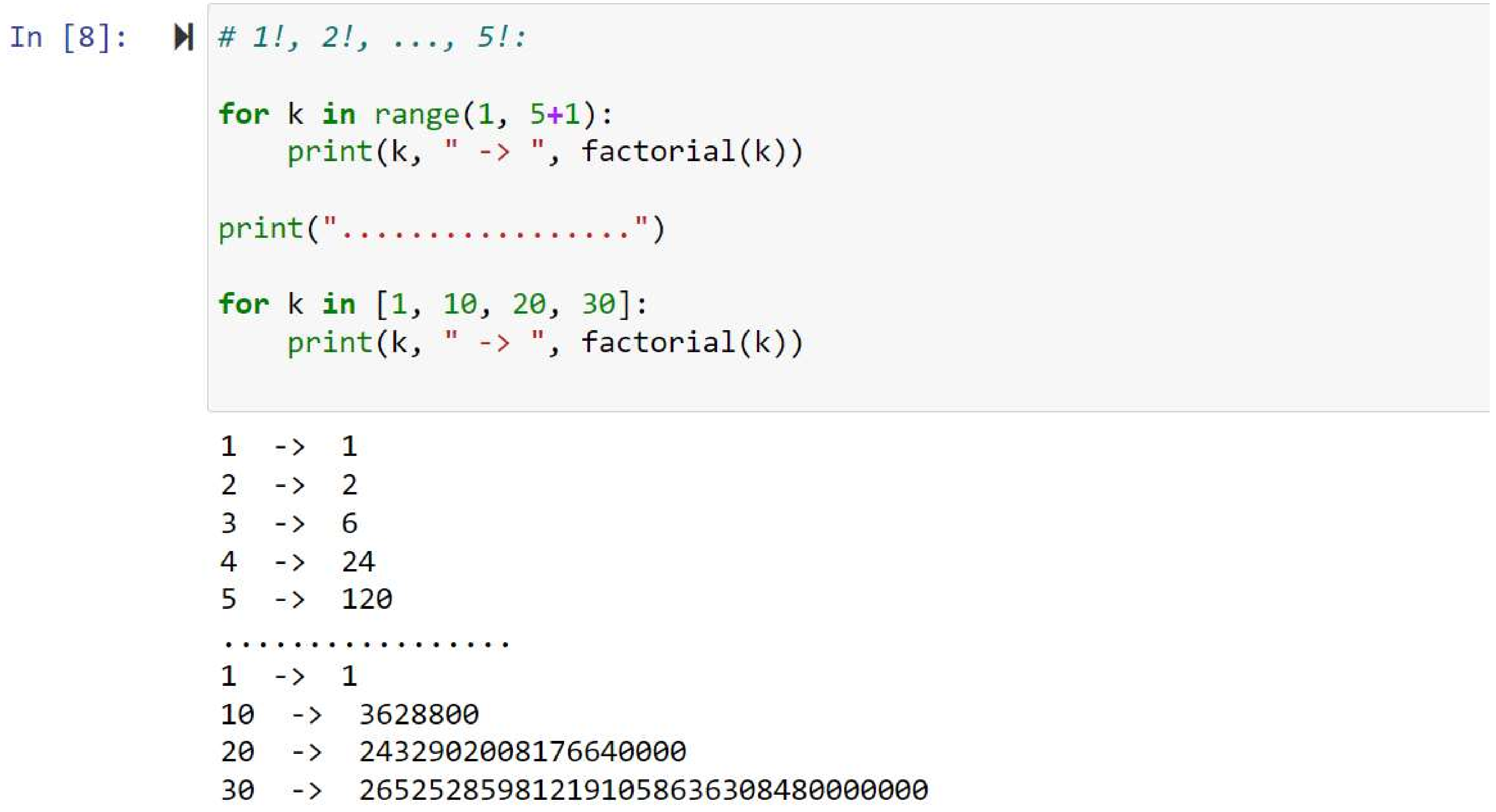
```
range(1, 10)
Out[15]: range(1, 10)

list(range (1, 10))
Out[16]: [1, 2, 3, 4, 5, 6, 7, 8, 9]

n = 10

list(range(1, n+1))
Out[18]: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```



```
In [1]:   # Función factorial, versión iterativa:

          def factorial(n):
              acum = 1
              for i in range(1, n+1):
                  acum = acum * i
              return acum

In [2]:   print(factorial(5))

          120
```

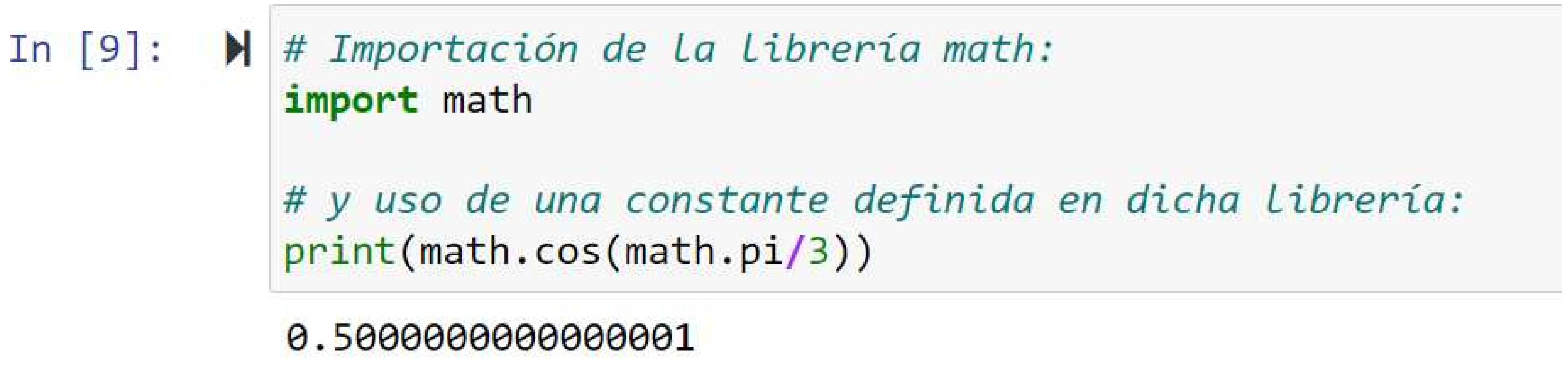


```
In [3]:   print(factorial(500))

          12201368259911100687012387854230469262535743428031928421924135883858453731538819976054964475022032
          81863013616477148203584163378722078177200480785205159329285477907571939330603772960859086270429174
          54788242491272634430567017327076946106280231045264421887878946575477714986349436778103764427403382
          73653974713864778784954384895955375379904232410612713269843277457155463099772027810145610811883737
          09531016356324432987029563896628911658974769572087926928871281780070265174507768410719624390394322
          53642260523494585012991857150124870696156814162535905669342381300885624924689156412677565448188650
          65938479517753608940057452389403357984763639449053130623237490664450488246650759467358620746379251
          84200459369692981022263971952597190945217823331756934581508552332820762820023402626907898342451712
          00620771464097945611612762914595123722991334016955236385094288559201872743379517301458635757082835
          57801587354327688886801203998823847021514676054454076635359841744304801289383138968816394874696588
          17504506926365338175055478128640000000000000000000000000000000000000000000000000000000000000000000
          000000000000000000000000000000000000000000000000000000000
```

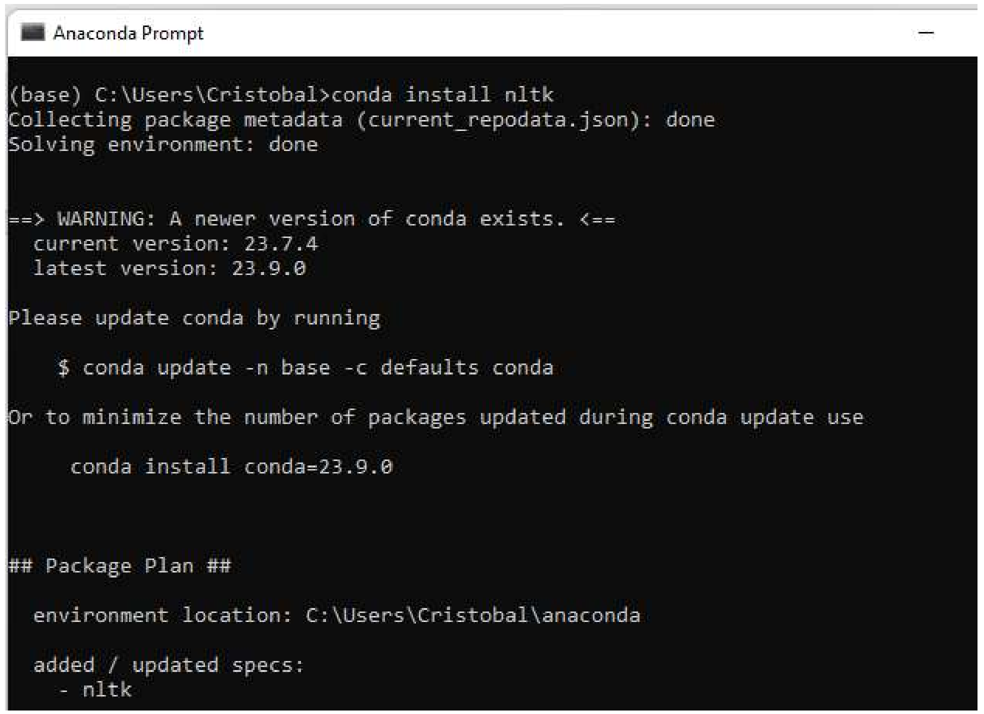


```
In [8]:   # 1!, 2!, ..., 5!:

          for k in range(1, 5+1):
              print(k, " -> ", factorial(k))

          print(".................")

          for k in [1, 10, 20, 30]:
              print(k, " -> ", factorial(k))
```

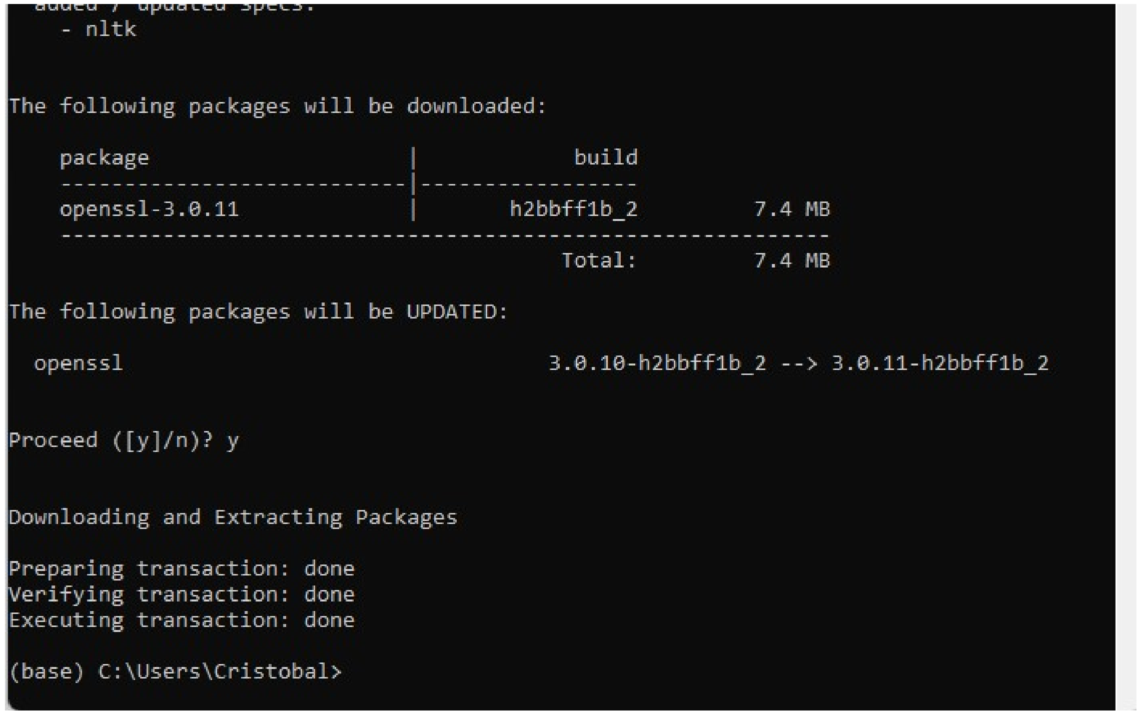
```
1  ->  1
2  ->  2
3  ->  6
4  ->  24
5  ->  120
.................
1  ->  1
10  ->  3628800
20  ->  2432902008176640000
30  ->  265252859812191058636308480000000
```



```
In [9]:  # Importación de la librería math:
         import math

         # y uso de una constante definida en dicha librería:
         print(math.cos(math.pi/3))

         0.5000000000000001
```



```
(base) C:\Users\Cristobal>conda install nltk
Collecting package metadata (current_repodata.json): done
Solving environment: done


==> WARNING: A newer version of conda exists. <==
  current version: 23.7.4
  latest version: 23.9.0

Please update conda by running

    $ conda update -n base -c defaults conda

Or to minimize the number of packages updated during conda update use

     conda install conda=23.9.0



## Package Plan ##

  environment location: C:\Users\Cristobal\anaconda

  added / updated specs:
    - nltk
```



```
  added / updated specs:
    - nltk


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    openssl-3.0.11             |       h2bbff1b_2         7.4 MB
    ------------------------------------------------------------
                                           Total:         7.4 MB

The following packages will be UPDATED:

  openssl                                 3.0.10-h2bbff1b_2 --> 3.0.11-h2bbff1b_2


Proceed ([y]/n)? y


Downloading and Extracting Packages

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

(base) C:\Users\Cristobal>
```

https://www.python.org/

https://ipython.org/

https://www.anaconda.com/

https://jupyter.org/

ntic master
revolucionamos la comunicación

UNIVERSIDAD COMPLUTENSE DE MADRID

Programación. Python

Las herramientas de trabajo

Cristóbal Pareja Flores