

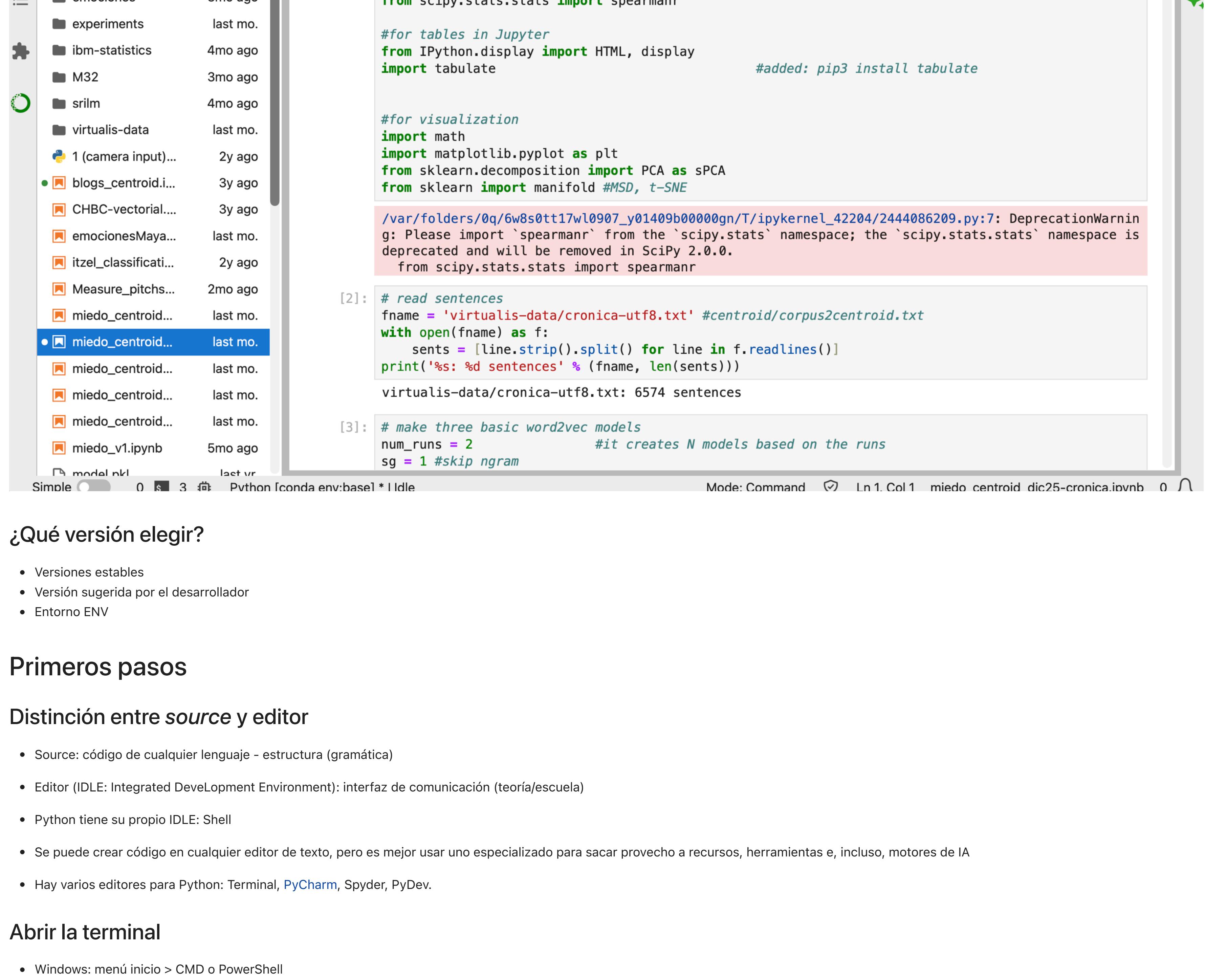
Introducción a Python: Hola mundo 😊

Reflexiones iniciales

- ¿Qué significa programar?
- ¿Cuáles son las características de un lenguaje?
- ¿Cuáles son las diferencias entre lenguaje natural y artificial?

Instalación de Python

- Requisitos
- Download
- Versiones: un gran problema para los desarrolladores



```
from gensim import models
from gensim.models.fasttext import FastText
from scipy.stats.stats import spearmanr

#for tables in Jupyter
from IPython.display import HTML, display
import tabulate

#added: pip3 install tabulate

#for visualization
import math
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA as sPCA
from sklearn import manifold #MDS, t-SNE

/var/folders/0q/6w8s0tt1wl0907_y01409b0000gn/T/ipykernel_42204/2444086209.py:7: DeprecationWarning: Please import `spearmanr` from the `scipy.stats` namespace; the `scipy.stats.stats` namespace is deprecated and will be removed in SciPy 2.0.0.
from scipy.stats.stats import spearmanr

[2]: # read sentences
fname = 'virtualis-data/cronica-utf8.txt' #centroide/corpus2centroide.txt
with open(fname) as f:
    sents = [line.strip() for line in f.readlines()]
print('%s: %d sentences' % (fname, len(sents)))
virtualis-data/cronica-utf8.txt: 6574 sentences

[3]: # make three basic word2vec models
num_runs = 2 #it creates N models based on the runs
sg = 1 #skip ngram
```

¿Qué versión elegir?

- Versiones estables
- Versión sugerida por el desarrollador
- Entorno ENV

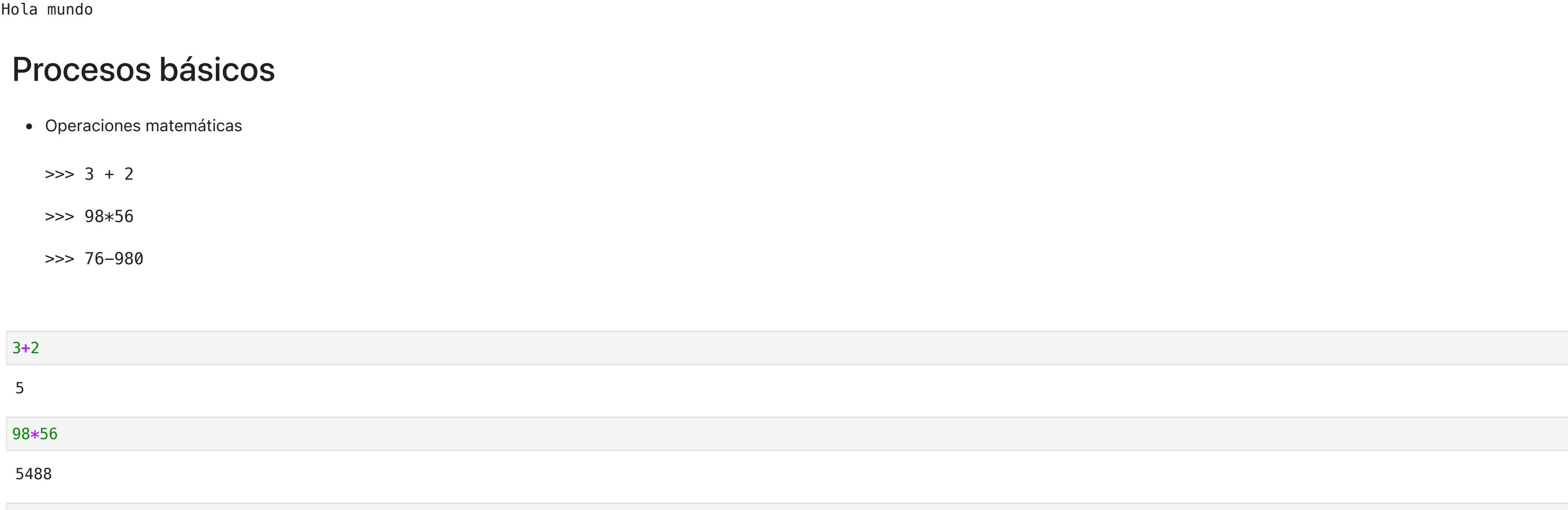
Primeros pasos

Distinción entre source y editor

- Source: código de cualquier lenguaje - estructura (gramática)
- Editor (IDLE: Integrated Development Environment): interfaz de comunicación (teoría/escuela)
- Python tiene su propio IDLE: Shell
- Se puede crear código en cualquier editor de texto, pero es mejor usar uno especializado para sacar provecho a recursos, herramientas e, incluso, motores de IA
- Hay varios editores para Python: Terminal, PyCharm, Spyder, PyDev.

Abrir la terminal

- Windows: menú inicio > CMD o PowerShell
- Mac: Spotlight > Terminal
- Ejecutar Python



```
Last login: Wed Jan 14 11:52:09 on ttys000
(base) arp@MacBook-Air-de-Antonio ~ % python
Python 3.13.5 | packaged by Anaconda, Inc. | (main, Jun 12 2025, 11:23:37) [Clang 14.0.
6 ] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

Empezando a leer los errores

```
>>> print hola mundo
In [14]: print hola mundo
File "/tmp/ipython-input-3859097774.py", line 1
    print hola mundo
    ^
SyntaxError: Missing parentheses in call to 'print'. Did you mean print(...)?
```

```
In [2]: print ('Hola mundo') #Toda secuencia textual debe ir entre comillas simples o dobles
Hola mundo
```

Procesos básicos

- Operaciones matemáticas

```
>>> 3 + 2
>>> 98*56
>>> 76-980
```

```
In [3]: 3*2
```

```
Out[3]: 5
```

```
In [4]: 98*56
```

```
Out[4]: 5488
```

```
In [6]: 76-980
```

```
Out[6]: -904
```

- Orden y secuencias

```
>>> 3 + 2 * 5 - 1
• Por qué se da ese resultado?
```

- Estandar de orden: PEMDAS (Parentheses, Exponentiation, Multiplication, Division, Addition, Subtraction)

```
>>> (3+2) * 5 - 1
```

```
>>> (3+2) * (5*5)
```

```
>>> (3+2) * (5*5) / 2
```

Ejercicio:

- Escribe el código para sumar $3 + 2 - 1 * 3$ y obtener 12 como resultado

```
>>>
```

Ejercicio:

- Explica por qué el resultado del siguiente código es 2

```
>>> (3+2) - 1 * 3
```

Variables

Una analogía lingüística

Las variables se pueden entender como signos que representan algo de la realidad: un dato, un número, una operación, un resultado, un corpus, etc.

1. **fif** = sonido fricativo, labiodental, sordo
2. **-it** = sufijo apreciativo
3. **silla** = objeto sólido con cuatro patas y una base que sirve para sentarse

Toda variable debe estar referenciada.

4. Prefiero tomarla en el verano, luego la coloco **ahí** y de inmediato **le** sucede **lo** que **yo** te había comentado.

5. **la** = cerveza

No debe existir ambigüedad, de lo contrario, el resultado puede ser sobre un dato que no es el que necesitamos.

6. **la** = botella

Hay que tener mucho cuidado con el tratamiento de las variables ya que éstas se pueden sobreescribir y, por lo tanto, representar (guardar) datos que han cambiado.

7. **la** no siempre es anáfora de cerveza.

Definición de variables en Python

Toda variable debe definirse respetando la siguiente sintaxis:

- **VARIABLE = algo**

```
>>> x = 3+2
```

Una vez que se asigna un valor a una variable, éste se puede usar para realizar operaciones sin tener que volver a poner el **algo**.

```
>>> x * 2 - 1
```

El proceso se puede volver recursivo y la variable guardará nuevos datos, sobreescribiendo los previos:

```
>>> x = x * 2 - 1
```

• **con este tipo de procesos por lo que se mencionó anteriormente sobre la reescritura de datos.**

- El código anterior modificó la variable **x**, ¿cómo puedo ver el valor actualizado?

```
>>> print VARIABLE
```

```
>>> print x
```

Ejercicio:

- Define dos variables y asigna a cada una el resultado de una operación matemática en la que incluyas secuencias parentéticas. Posteriormente, obtén un siguiente resultado que represente la división de la variable 1 entre la variable 2. El resultado lo guardas en una variable 3. Finalmente, imprime el dato que tiene la variable 3.

Las variables también pueden almacenar datos textuales. Para ello, Python exige que este tipo de datos cumpla con la sintaxis que se estableció anteriormente: uso de **comillas simples o dobles**.

Ejercicio:

- Toma el código anterior y genera una variable 4 que almacene el siguiente texto: El resultado de la operación es:

- Imprime las variables necesarias para que en pantalla aparezca lo siguiente: El resultado de la operación es: **VALOR**

Nota: Python permite hacer operaciones sobre texto. Prueba sumar o multiplicar una cadena de texto.