Procesamiento Digital de Señales

Introducción a Python

Facultad de Ingenieria, Universidad de Antioquia

Juan Camilo Vásquez Correa

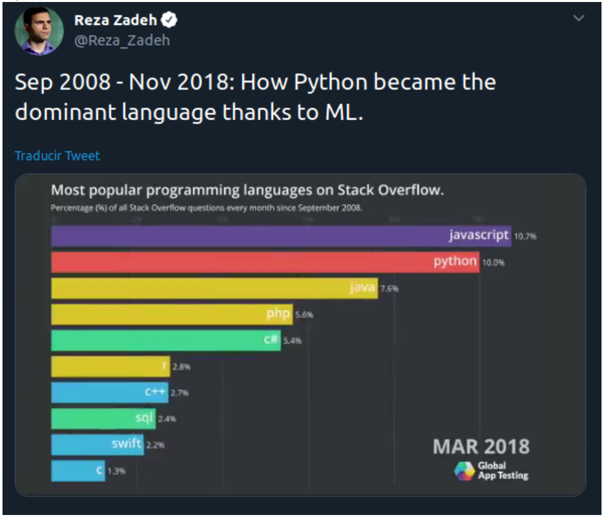
4 de diciembre de 2019

Contenido

Próxima sección

UNIVERSIDAD DE ANTIOQUIA 1 8 0 3

Why Python


Reza Zadeh
@Reza_Zadeh

Sep 2008 - Nov 2018: How Python became the dominant language thanks to ML.

Traducir Tweet

Most popular programming languages on Stack Overflow.
Percentage (%) of all Stack Overflow questions every month since September 2008.
javascript 10.7%
python 10.0%
java 7.6%
php 5.6%
c# 5.4%
r 2.8%
c++ 2.7%
sql 2.4%
swift 2.2%
c 1.3%
MAR 2018
Global App Testing

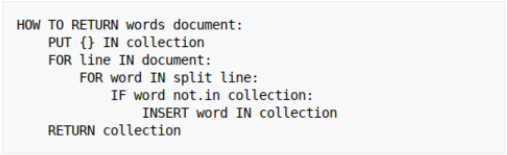
https://twitter.com/i/status/11214522792891351O

Why Python

Guido Van Rossum

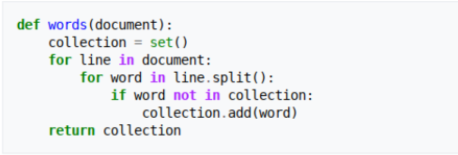
Why Python

Recuperar las palabras de un documento en ABC



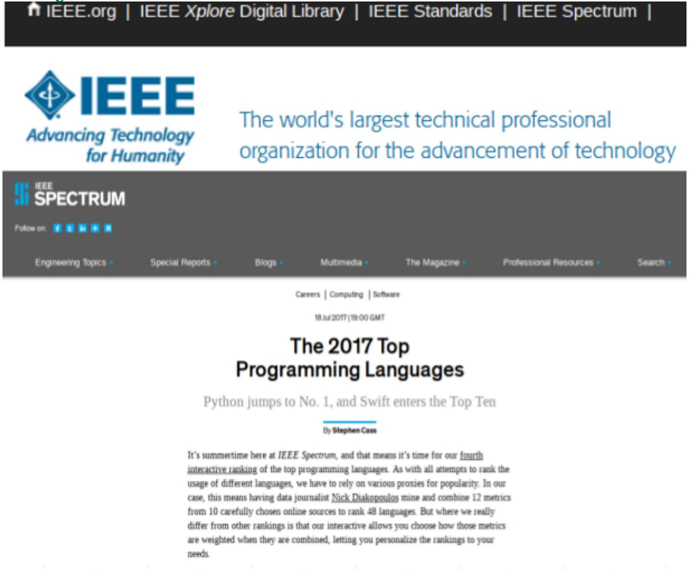
```
HOW TO RETURN words document:
    PUT {} IN collection
    FOR line IN document:
        FOR word IN split line:
            IF word not.in collection:
                INSERT word IN collection
    RETURN collection
```

Recuperar las palabras de un documento en Python



```python
def words(document):
    collection = set()
    for line in document:
        for word in line.split():
            if word not in collection:
                collection.add(word)
    return collection
```

Why Python



IEEE.org | IEEE *Xplore* Digital Library | IEEE Standards | IEEE Spectrum |

IEEE
Advancing Technology for Humanity
The world's largest technical professional organization for the advancement of technology
IEEE SPECTRUM
Follow on:
Engineering Topics
Special Reports
Blogs
Multimedia
The Magazine
Professional Resources
Search

Careers | Computing | Software

18 Jul 2017 | 19:00 GMT

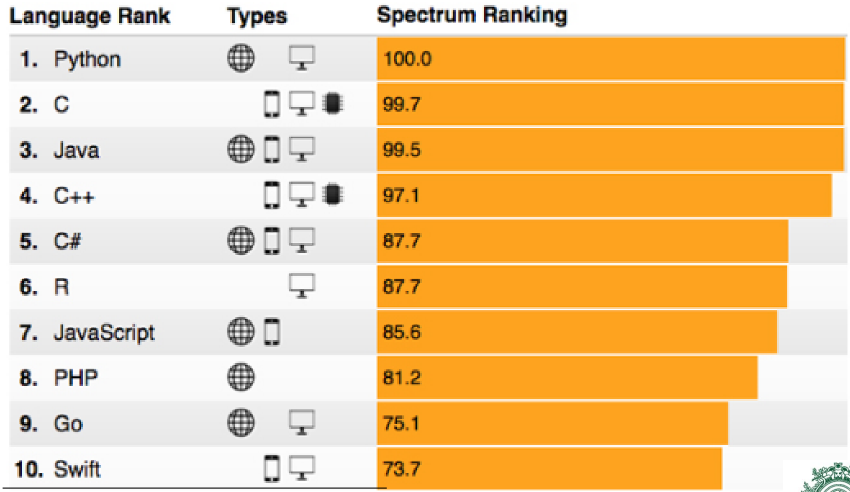
The 2017 Top Programming Languages

Python jumps to No. 1, and Swift enters the Top Ten

By Stephen Cass

It's summertime here at *IEEE Spectrum*, and that means it's time for our fourth interactive ranking of the top programming languages. As with all attempts to rank the usage of different languages, we have to rely on various proxies for popularity. In our case, this means having data journalist Nick Diakopoulos mine and combine 12 metrics from 10 carefully chosen online sources to rank 48 languages. But where we really differ from other rankings is that our interactive allows you choose how those metrics are weighted when they are combined, letting you personalize the rankings to your needs.

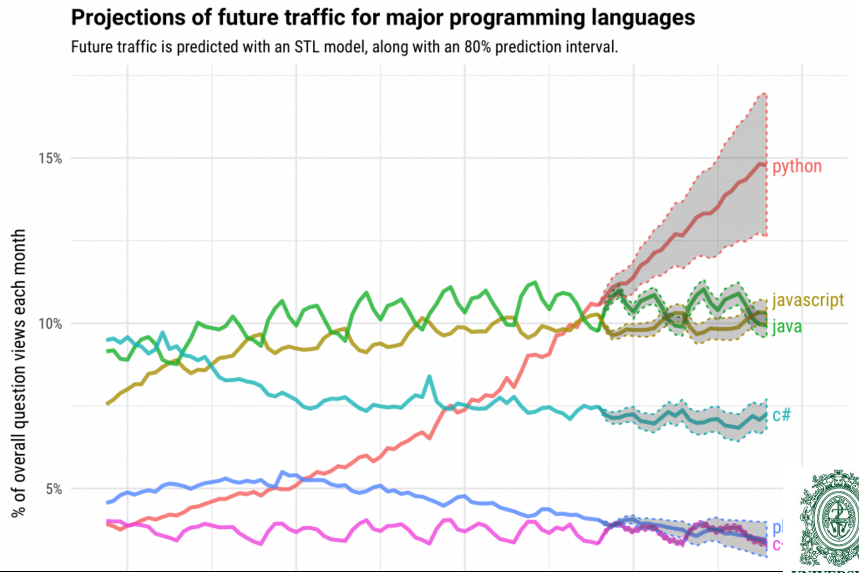
Why Python



Language Rank	Types	Spectrum Ranking
1. Python	🌐 🖥	100.0
2. C	📱 🖥 ■	99.7
3. Java	🌐 📱 🖥	99.5
4. C++	📱 🖥 ■	97.1
5. C#	🌐 📱 🖥	87.7
6. R	🖥	87.7
7. JavaScript	🌐 📱	85.6
8. PHP	🌐	81.2
9. Go	🌐 🖥	75.1
10. Swift	📱 🖥	73.7

[1] Source: IEEE Spectrum 2017:
http://spectrum.ieee.org/computing/software/the-2017-top-programming-

Why python


Projections of future traffic for major programming languages
Future traffic is predicted with an STL model, along with an 80% prediction interval.
% of overall question views each month
15%
10%
5%
python
javascript
java
c#
p
c

Why python

python
powered

```
print("Hello, world!")
```

ANACONDA®

UNIVERSIDAD DE ANTIOQUIA 1803

Python3

A (very incomplete) list of properties Python

- Multi-paradigm programming language (e.g. object-oriented)
- Garbage collection for memory management
- The intention is to have highly readable code
- Indentation instead of (curly) braces
- Dynamic typing
- Data types / structures

Lists	`a = []`	Tuple	`b = ()`
Dictionaries	`d = {'one':1, 'two':2}`	Strings	`str = "hi"`
Integer	`i = 3`	Float	`j = 1.5`

  Type verification: `type(i)`
- Due to Python's popularity there are numerous libraries available for everything one can think of.

UNIVERSIDAD DE ANTIOQUIA 1803

Why python

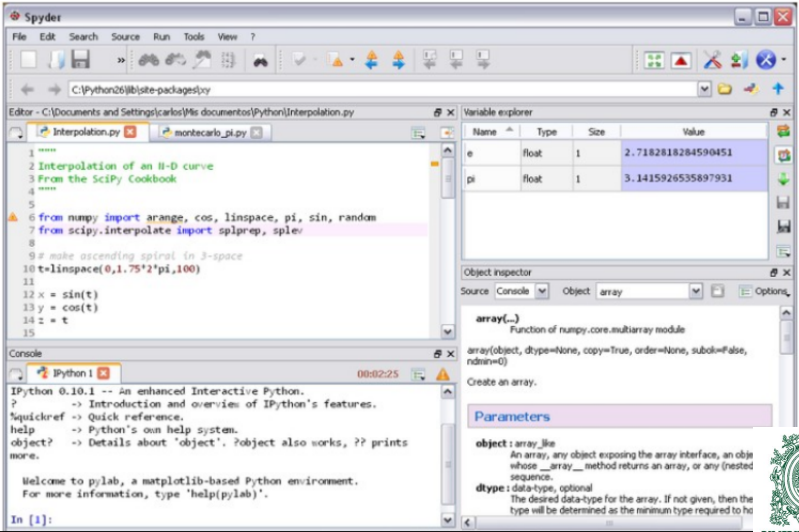
IDE

Integrated Development Environment

UNIVERSIDAD DE ANTIOQUIA
1 8 0 3

Why python

Spyder


Spyder
File Edit Search Source Run Tools View ?
C:\Python26\lib\site-packages\xy
Editor - C:\Documents and Settings\carlos\Mis documentos\Python\Interpolation.py
Interpolation.py
montecarlo_pi.py
Variable explorer
Object inspector
Source Console Object array Options
Console
IPython 1
00:02:25

```
"""
Interpolation of an N-D curve
From the SciPy Cookbook
"""

from numpy import arange, cos, linspace, pi, sin, random
from scipy.interpolate import splprep, splev

# make ascending spiral in 3-space
t=linspace(0,1.75*2*pi,100)

x = sin(t)
y = cos(t)
z = t

```

Name	Type	Size	Value
e	float	1	2.718281828459045 1
pi	float	1	3.1415926535897931

array(...)
Function of numpy.core.multiarray module

array(object, dtype=None, copy=True, order=None, subok=False, ndmin=0)

Create an array.

Parameters

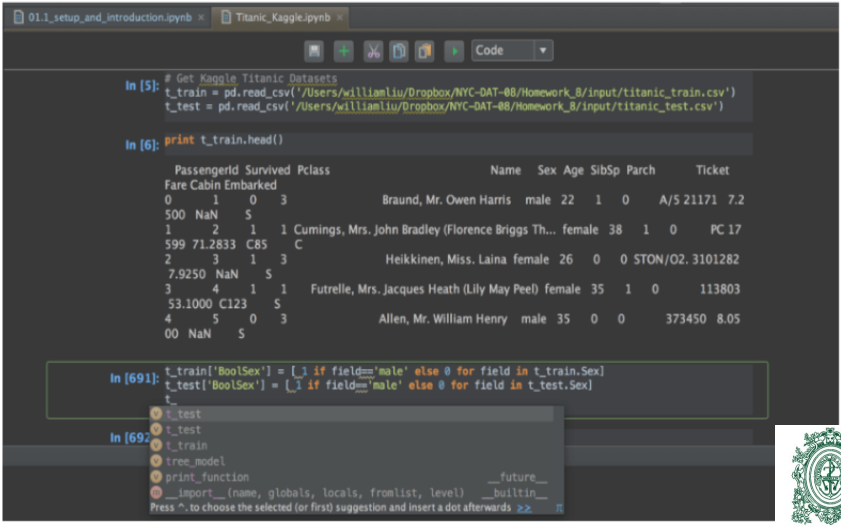
object : array_like
An array, any object exposing the array interface, an obje
whose __array__ method returns an array, or any (nested
sequence.
dtype : data-type, optional
The desired data-type for the array. If not given, then the
type will be determined as the minimum type required to ho

```
IPython 0.10.1 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object'. ?object also works, ?? prints
more.

  Welcome to pylab, a matplotlib-based Python environment.
  For more information, type 'help(pylab)'.

In [1]:
```

UNIVERSIDAD DE ANTIOQUIA
1 8 0 3

Why python

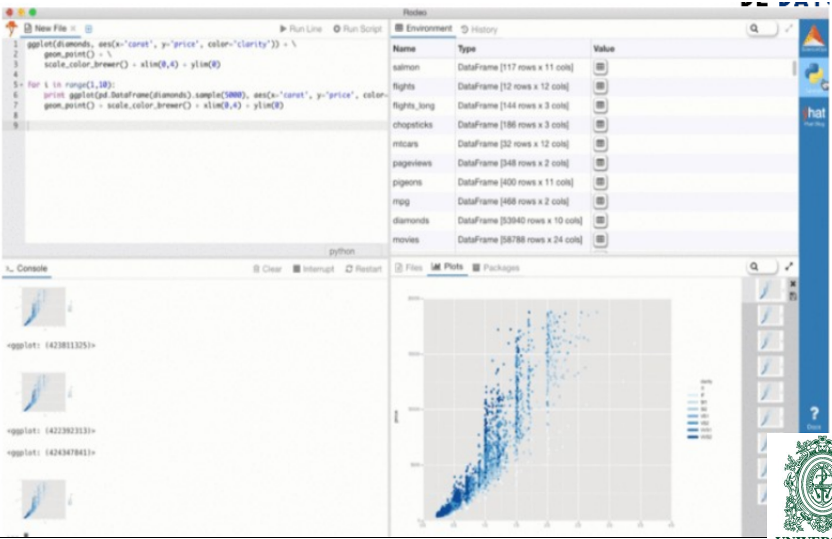
Pycharm



```
In [5]: # Get Kaggle Titanic Datasets
        t_train = pd.read_csv('/Users/williamliu/Dropbox/NYC-DAT-08/Homework_8/input/titanic_train.csv')
        t_test = pd.read_csv('/Users/williamliu/Dropbox/NYC-DAT-08/Homework_8/input/titanic_test.csv')

In [6]: print t_train.head()

   PassengerId  Survived  Pclass                                      Name     Sex  Age  SibSp  Parch           Ticket
Fare Cabin Embarked
0            1         0       3                   Braund, Mr. Owen Harris    male   22      1      0        A/5 21171  7.2
500   NaN        S
1            2         1       1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female   38      1      0         PC 17
599  71.2833   C85        C
2            3         1       3                    Heikkinen, Miss. Laina  female   26      0      0  STON/O2. 3101282
 7.9250   NaN        S
3            4         1       1      Futrelle, Mrs. Jacques Heath (Lily May Peel)  female   35      1      0           113803
 53.1000  C123        S
4            5         0       3                  Allen, Mr. William Henry    male   35      0      0           373450  8.05
00   NaN        S

In [691]: t_train['BoolSex'] = [ 1 if field=='male' else 0 for field in t_train.Sex]
          t_test['BoolSex'] = [ 1 if field=='male' else 0 for field in t_test.Sex]
          t_
          t_test
          t_test
          t_train
          tree_model
          print_function                                   __future__
          __import__(name, globals, locals, fromlist, level)   __builtin__
          Press ^. to choose the selected (or first) suggestion and insert a dot afterwards >>

In [692
```

Why python

Rodeo



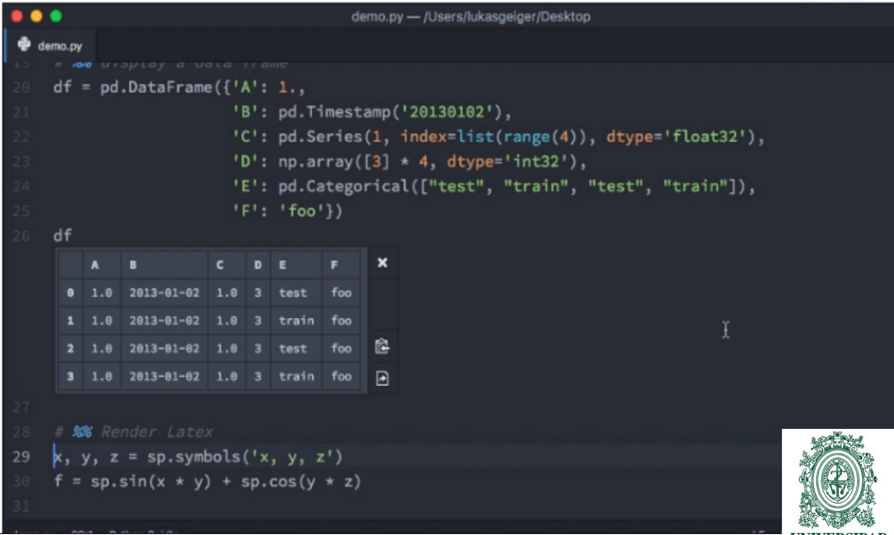
```
ggplot(diamonds, aes(x='carat', y='price', color='clarity')) + \
    geom_point() + \
    scale_color_brewer() + xlim(0,4) + ylim(0)

for i in range(1,10):
    print ggplot(pd.DataFrame(diamonds).sample(5000), aes(x='carat', y='price', color-
    geom_point() + scale_color_brewer() + xlim(0,4) + ylim(0)
```

Name	Type	Value
salmon	DataFrame [117 rows x 11 cols]	▦
flights	DataFrame [12 rows x 12 cols]	▦
flights_long	DataFrame [144 rows x 3 cols]	▦
chopsticks	DataFrame [186 rows x 3 cols]	▦
mtcars	DataFrame [32 rows x 12 cols]	▦
pageviews	DataFrame [348 rows x 2 cols]	▦
pigeons	DataFrame [400 rows x 11 cols]	▦
mpg	DataFrame [468 rows x 2 cols]	▦
diamonds	DataFrame [53940 rows x 10 cols]	▦
movies	DataFrame [58788 rows x 24 cols]	▦

```
<ggplot: (423811325)>

<ggplot: (422392313)>
<ggplot: (424347841)>
```

Why python

Atom

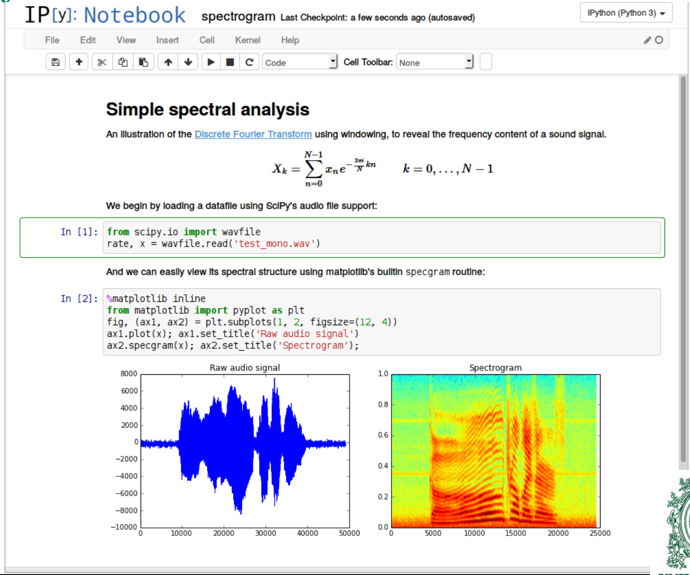


```
# %% display a data frame
df = pd.DataFrame({'A': 1.,
                   'B': pd.Timestamp('20130102'),
                   'C': pd.Series(1, index=list(range(4)), dtype='float32'),
                   'D': np.array([3] * 4, dtype='int32'),
                   'E': pd.Categorical(["test", "train", "test", "train"]),
                   'F': 'foo'})
df
```

	A	B	C	D	E	F
0	1.0	2013-01-02	1.0	3	test	foo
1	1.0	2013-01-02	1.0	3	train	foo
2	1.0	2013-01-02	1.0	3	test	foo
3	1.0	2013-01-02	1.0	3	train	foo

```

# %% Render Latex
x, y, z = sp.symbols('x, y, z')
f = sp.sin(x * y) + sp.cos(y * z)

```

Why python


IP[y]: Notebook
spectrogram Last Checkpoint: a few seconds ago (autosaved)
IPython (Python 3)
File Edit View Insert Cell Kernel Help
Code
Cell Toolbar: None

Simple spectral analysis

An illustration of the Discrete Fourier Transform using windowing, to reveal the frequency content of a sound signal.

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \qquad k = 0, \dots, N-1$$

We begin by loading a datafile using SciPy's audio file support:

```
In [1]: from scipy.io import wavfile
        rate, x = wavfile.read('test_mono.wav')
```

And we can easily view its spectral structure using matplotlib's builtin `specgram` routine:

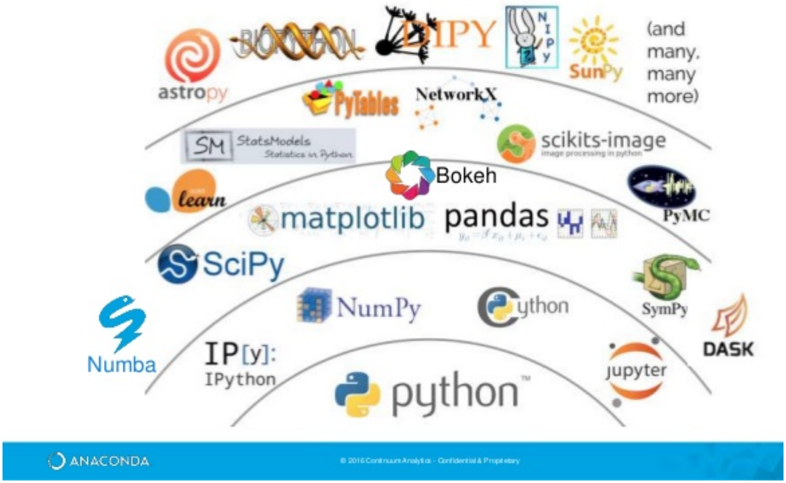
```
In [2]: %matplotlib inline
        from matplotlib import pyplot as plt
        fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))
        ax1.plot(x); ax1.set_title('Raw audio signal')
        ax2.specgram(x); ax2.set_title('Spectrogram');
```

Jupyter

Próxima sección

UNIVERSIDAD DE ANTIOQUIA 1803

Tools


astropy
BIOPYTHON
DIPY
NIPY
SunPy
(and many, many more)
PyTables
NetworkX
StatsModels Statistics in Python
scikits-image image processing in python
learn
Bokeh
PyMC
matplotlib
pandas
SciPy
Numba
NumPy
Cython
SymPy
DASK
IP[y]: IPython
python
jupyter
ANACONDA
© 2016 Continuum Analytics - Confidential & Proprietary

Tools

- Jupyter
- Numpy
- Pandas
- Matplotlib
- Scipy

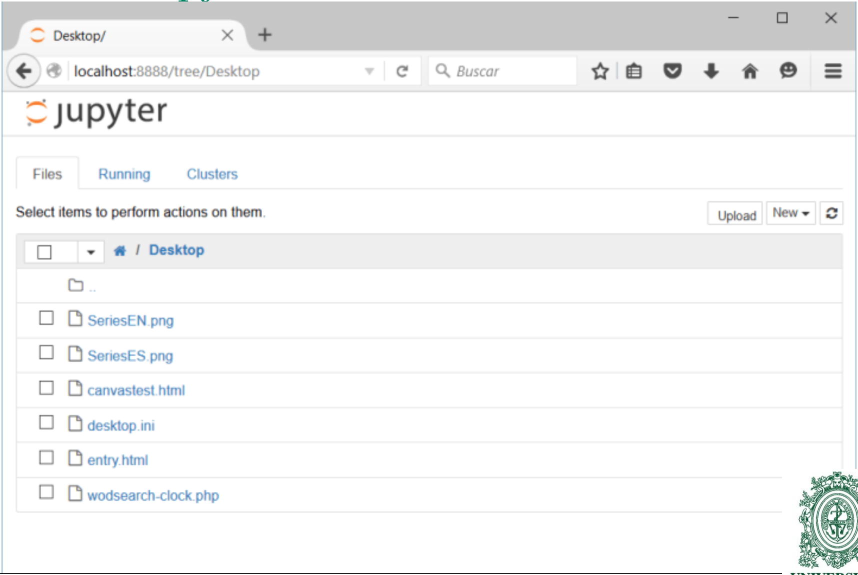
Tools: Jupyter

Interactive environment to create documents, code, interactive widgets, graphs, texts and equations.

[2]

[2] `http://jupyter.org/`

Tools: Jupyter


Desktop/
localhost:8888/tree/Desktop
Buscar
jupyter
Files Running Clusters

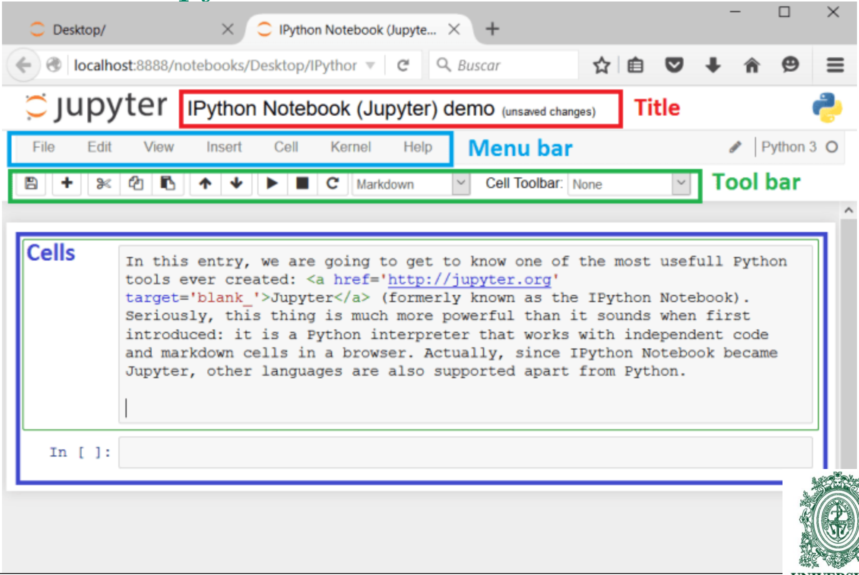
Select items to perform actions on them.

Upload New

/ Desktop

..
SeriesEN.png
SeriesES.png
canvastest.html
desktop.ini
entry.html
wodsearch-clock.php

Tools: Jupyter

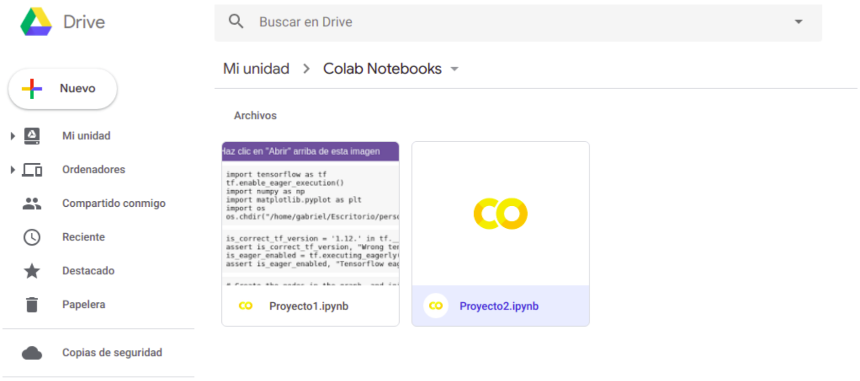

Desktop/
IPython Notebook (Jupyte...
localhost:8888/notebooks/Desktop/IPythor
Buscar
jupyter
IPython Notebook (Jupyter) demo (unsaved changes)
Title
File
Edit
View
Insert
Cell
Kernel
Help
Menu bar
Python 3
Markdown
Cell Toolbar: None
Tool bar
Cells

```
In this entry, we are going to get to know one of the most usefull Python
tools ever created: <a href='http://jupyter.org'
target='blank_'>Jupyter</a> (formerly known as the IPython Notebook).
Seriously, this thing is much more powerful than it sounds when first
introduced: it is a Python interpreter that works with independent code
and markdown cells in a browser. Actually, since IPython Notebook became
Jupyter, other languages are also supported apart from Python.
```

```
In [ ]:
```

UNIVERSIDAD DE ANTIOQUIA 1803

Tools: Google Colab

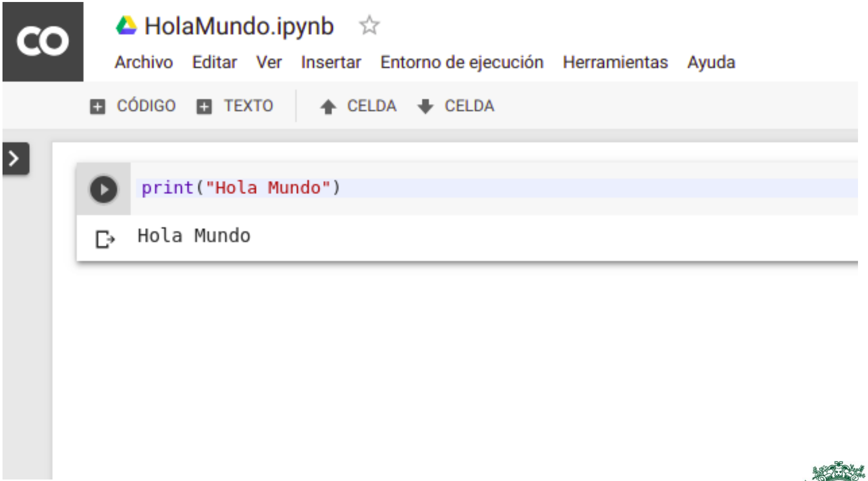


```
import tensorflow as tf
tf.enable_eager_execution()
import numpy as np
import matplotlib.pyplot as plt
import os
os.chdir("/home/gabriel/Escritorio/pers

is_correct_tf_version = '1.12.' in tf._
assert is_correct_tf_version, "Wrong ter
is_eager_enabled = tf.executing_eagerly(
assert is_eager_enabled, "Tensorflow eag
```

[5]

[5] https://colab.research.google.com/notebooks/welcome.ipynb

Tools: Google Colab


co
HolaMundo.ipynb
Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda
CÓDIGO TEXTO CELDA CELDA

```
print("Hola Mundo")
```

```
Hola Mundo
```

6

[6] https://colab.research.google.com/notebooks/welcome.ipynb

Tools: List processing

- Lists are just like the arrays, declared in other languages.
- Lists need not be homogeneous always which makes it a most powerful tool in Python.
- A single list may contain DataTypes like Integers, Strings, as well as Objects.
- Lists are also very useful for implementing stacks and queues.

UNIVERSIDAD DE ANTIOQUIA 1803

Tools: List processing

```python
# Creating a List
List = []
print("Intial blank List: ")
print(List)

# Creating a List with
# the use of a String
List = ['GeeksForGeeks']
print("\nList with the use of String: ")
print(List)

# Creating a List with
# the use of multiple values
List = ["Geeks", "For", "Geeks"]
print("\nList containing multiple values: ")
print(List[0])
print(List[2])
```

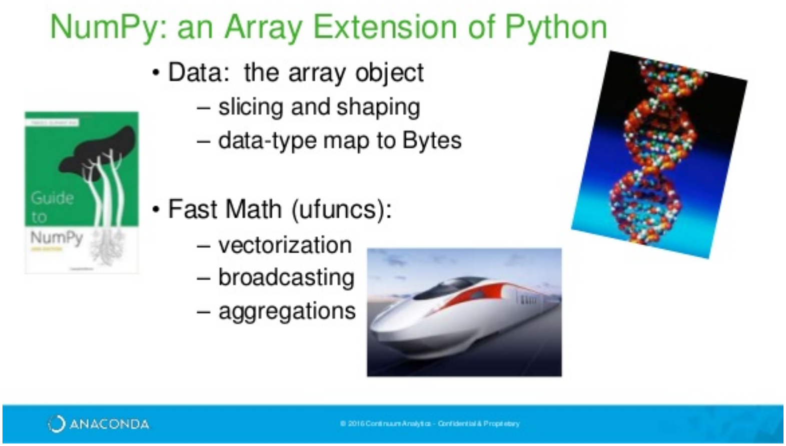
UNIVERSIDAD DE ANTIOQUIA 1803

Tools: List processing

```python
# Creating a List with
# the use of Numbers
# (Having duplicate values)
List = [1, 2, 4, 4, 3, 3, 3, 6, 5]
print("\nList with the use of Numbers: ")
print(List)

# Creating a List with
# mixed type of values
# (Having numbers and strings)
List = [1, 2, 'Geeks', 4, 'For', 6, 'Geeks']
print("\nList with the use of Mixed Values: ")
print(List)
```

UNIVERSIDAD DE ANTIOQUIA 1803

Tools: Numpy



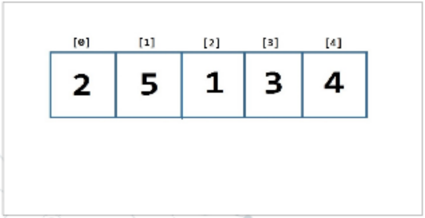
NumPy: an Array Extension of Python

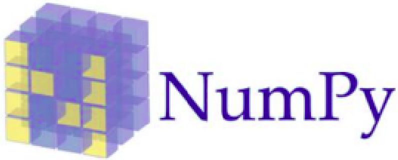
- Data: the array object
  - slicing and shaping
  - data-type map to Bytes
- Fast Math (ufuncs):
  - vectorization
  - broadcasting
  - aggregations

Guide to NumPy
ANACONDA
© 2016 Continuum Analytics - Confidential & Proprietary

Tools: Numpy

- Array, matrix, vectorial operations.
- Import numpy as np

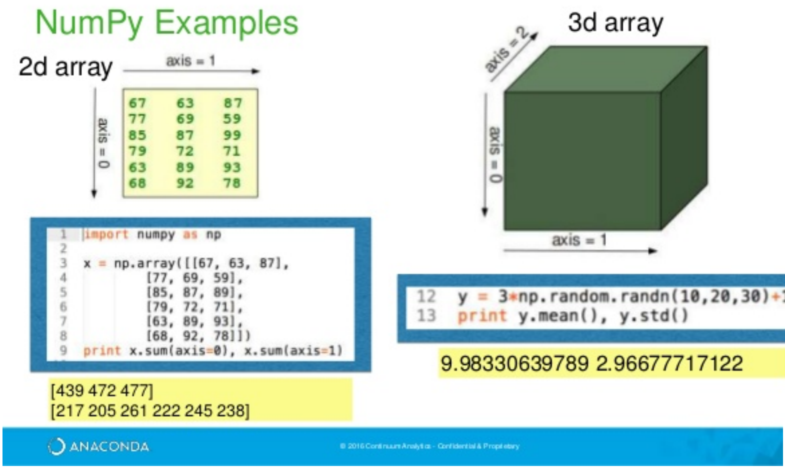




[7]

[7] http://www.numpy.org/

Tools: Numpy



NumPy Examples

```
import numpy as np

x = np.array([[67, 63, 87],
              [77, 69, 59],
              [85, 87, 89],
              [79, 72, 71],
              [63, 89, 93],
              [68, 92, 78]])
print x.sum(axis=0), x.sum(axis=1)
```

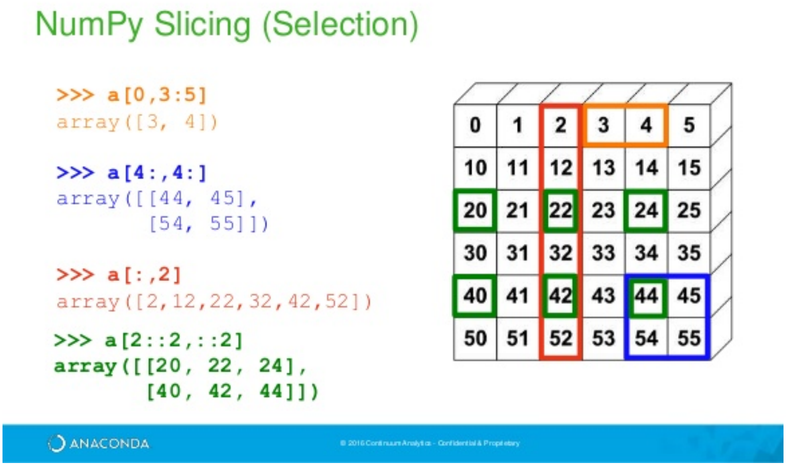
```
[439 472 477]
[217 205 261 222 245 238]
```

```
y = 3*np.random.randn(10,20,30)+10
print y.mean(), y.std()
```

```
9.98330639789 2.96677717122
```

ANACONDA

Tools: Numpy



NumPy Slicing (Selection)

```
>>> a[0,3:5]
array([3, 4])

>>> a[4:,4:]
array([[44, 45],
       [54, 55]])

>>> a[:,2]
array([2,12,22,32,42,52])

>>> a[2::2,::2]
array([[20, 22, 24],
       [40, 42, 44]])
```

0	1	2	3	4	5
10	11	12	13	14	15
20	21	22	23	24	25
30	31	32	33	34	35
40	41	42	43	44	45
50	51	52	53	54	55

ANACONDA

UNIVERSIDAD DE ANTIOQUIA 1803

Tools: Numpy

```
# manejo de arreglos

import numpy as np #libreria estandar para manejo de datos y senales en Python

a = np.zeros((2,2))  # Crea un arreglo tipo matriz de ceros
print(a)               # Prints "[[ 0.  0.]
                      #          [ 0.  0.]]"
b = np.ones((1,2))    # Crea un arreglo tipo vector fila de unos
print(b)              # Prints "[[ 1.  1.]]"

c = np.full((2,2), 7) # Crea un arreglo tipo matriz de un valor constante
print(c)                # Prints "[[ 7.  7.]
                       #          [ 7.  7.]]"
c = 7*np.ones((2,2)) # Otra forma
print (c)

d = np.eye(2)         # Crea un arreglo con la matriz
```

Tools: Numpy

```
e = np.random.random((2,2)) # Crea un arreglo tipo matriz de numeros aleatorios entre 0 y 1
print (e)        # print "[[ 0.91940167  0.08143941]
                 #         [ 0.68744134  0.87236687]]"
print (e[0,1])                        # Seleccionar un elemento
print (e[0,:])                       # Seleccionar la primera fila
print (e[:,1])                        # Seleccionar la segunda columna

xrang=np.arange(20) # Crea un arreglo de numeros consecutivos entre 0 y 19
print (xrang)           # print "[0 1 2 3 4 5... 19]"
# Para seleccionar partes del arreglo use los siguientes comandos
print (xrang[0:5])      # print "[0 1 2 3 4]"
print (xrang[10])       # print [10]

A=np.random.random(100)
media=np.mean(A) # calcula la media de un arreglo A
maxA=np.max(A) # calcula el maximo valor de un arreglo
```

Tools: Pandas

Python has long been great for data munging and preparation, but less so for data analysis and modeling. pandas helps fill this gap, enabling you to carry out your entire data analysis workflow in Python

pandas

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$ [8]

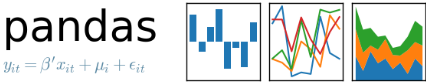
[8] `https://pandas.pydata.org/`

UNIVERSIDAD DE ANTIOQUIA 1803

Tools: Pandas

Data structure and tools for data analysis.

- DataFrames
- GroupBy, merge, join
- Data reading and writing
- Import pandas as pd



[9]

[9] https://pandas.pydata.org/

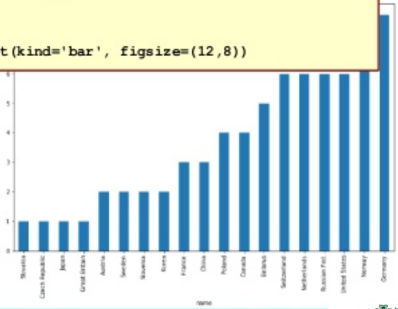
Tools: Pandas

pandas $y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

Easy Data Wrangling

```
medals = pd.read_csv('data/medals.csv', index_col='name')
medals.head()
gold = medals['medal'] == 'gold'
won = medals['count'] > 0
medals.loc[gold & won, 'count'].sort_values().plot(kind='bar', figsize=(12,8))
```

	count	medal	country
name			
Australia	1	bronze	AUS
Australia	2	silver	AUS
Australia	0	gold	AUS
Austria	1	bronze	AUT
Austria	6	silver	AUT



Tools: Matplotlib

Graphics in Python

- Frequency diagram
- Histograms
- Scatter-plots
- 3D graphics
- Surface

Import matplotlib

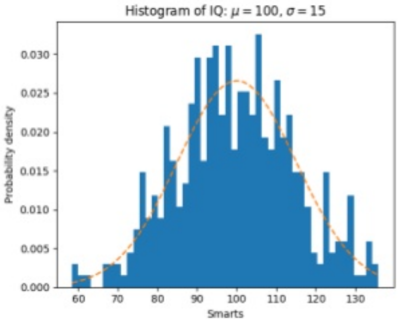
[10]

[10] https://matplotlib.org/

Tools: Matplotlib

matplotlib a powerful plotting engine

```
import numpy as np
import matplotlib.mlab as mlab
import matplotlib.pyplot as plt

np.random.seed(0)

# example data
mu = 100  # mean of distribution
sigma = 15  # standard deviation of distribution
x = mu + sigma * np.random.randn(437)

num_bins = 50

fig, ax = plt.subplots()

# the histogram of the data
n, bins, patches = ax.hist(x, num_bins, normed=1)

# add a 'best fit' line
y = mlab.normpdf(bins, mu, sigma)
ax.plot(bins, y, '--')
ax.set_xlabel('Smarts')
ax.set_ylabel('Probability density')
ax.set_title(r'Histogram of IQ: $\mu=100$, $\sigma=15$')

# Tweak spacing to prevent clipping of ylabel
fig.tight_layout()
plt.show()
```



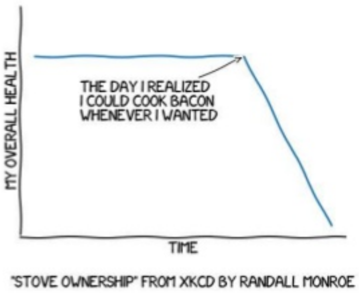
ANACONDA

UNIVERSIDAD DE ANTIOQUIA 1803

Tools: Matplotlib

matplotlib a powerful plotting engine

```
import matplotlib.pyplot as plt
import numpy as np

with plt.xkcd():
    fig = plt.figure()
    ax = fig.add_axes((0.1, 0.2, 0.8, 0.7))
    ax.spines['right'].set_color('none')
    ax.spines['top'].set_color('none')
    plt.xticks([])
    plt.yticks([])
    ax.set_ylim([-30, 10])
    data = np.ones(100)
    data[70:] -= np.arange(30)

    plt.annotate('THE DAY I REALIZED\nI COULD
COOK BACON\nWHENEVER I WANTED',
        xy=(70, 1),
        arrowprops=dict(arrowstyle='->'),
        xytext=(15, -10))

    plt.plot(data)

    plt.xlabel('time')
    plt.ylabel('my overall health')
    fig.text(0.5, 0.05, '"Stove Ownership" from
xkcd by Randall Monroe',
        ha='center')
```

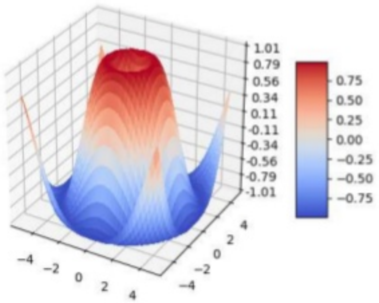


ANACONDA © 2016 Continuum Analytics - Confidential & Proprietary

Tools: Matplotlib

matplotlib a powerful plotting engine

```
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
from matplotlib import cm
from matplotlib.ticker import LinearLocator,
FormatStrFormatter
import numpy as np

fig = plt.figure()
ax = fig.gca(projection='3d')
# Make data.
X = np.arange(-5, 5, 0.25)
Y = np.arange(-5, 5, 0.25)
X, Y = np.meshgrid(X, Y)
R = np.sqrt(X**2 + Y**2)
Z = np.sin(R)
# Plot the surface.
surf = ax.plot_surface(X, Y, Z, cmap=cm.coolwarm,
linewidth=0, antialiased=False)

# Customize the z axis.
ax.set_zlim(-1.01, 1.01)
ax.zaxis.set_major_locator(LinearLocator(10))
ax.zaxis.set_major_formatter(FormatStrFormatter('%
.02f'))

# Add a color bar which maps values to colors.
fig.colorbar(surf, shrink=0.5, aspect=5)
plt.show()
```

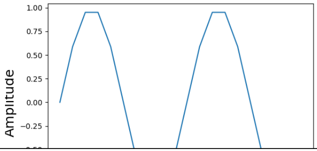


ANACONDA © 2016 Continuum Analytics - Confidential & Proprietary

UNIVERSIDAD DE ANTIOQUIA 1803

Tools: Matplotlib

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
f=1.0 # Signal frequency
fs=10.0 # Sampling frequency
t=np.arange(0, 2.0, 1.0/fs) # Vector de tiempo
x = np.sin(2*np.pi*f*t)
plt.plot(t,x)
plt.xlabel('Time',fontsize=18)
plt.ylabel('Amplitude',fontsize=18)
plt.show()
```



File managment

Load Text files containing vector or matrices data.

```python
import numpy as np
data=np.loadtxt("file.txt")
```

Write a Text file with vector or matrices data.

```python
Data=np.random.rand((100,20))
np.savetxt("file.txt", Data)
```

File managment

Loading and plotting a Text file that contains a signal.

```python
data=np.loadtxt("PrecioDolar.txt")
signal=data[:,3]
t=np.arange(len(signal))
plt.plot(t,signal)
plt.xlabel("samples")
plt.ylabel("TRM COP -US")
plt.show()
```

```python
Ts=1.0/365
t=np.arange(1992, 2016+244*Ts+5*Ts, Ts)
plt.plot(t,signal)
plt.xlabel("Fecha")
plt.ylabel("TRM COP-US")
plt.show()
```

File managment

Load data from excel files (.csv)

```python
import pandas as pd
import matplotlib.pyplot as plt
data=pd.read_csv("DataSiata.csv")
signal=data["pm10"]
t=data["index"]
plt.plot(t,signal)
plt.xlabel("samples")
plt.ylabel("Concentracion particulas PM10")
plt.show()
```

File managment

Load audio signals (.wav)

```
from scipy.io.wavfile import read
import matplotlib.pyplot as plt
fs,signal=read("098_readtext_PCGITA.wav")
t=np.arange(0,len(signal)/fs, 1./fs)
plt.plot(t,signal)
plt.xlabel("Time [s]")
plt.ylabel("Amplitude")
plt.show()
```

File managment

Write signal into .wav filee

```python
from scipy.io.wavfile import write
import numpy as np

fs=44100
t=np.arange(0,10,1.0/fs)
signal=np.sin(2*np.pi*fs*(t**3))

signal2=np.asarray(signal*2**15, dtype=np.int16)
write("signal.wav", fs, signal2)
```

Functions

If a sequence of commands is used more often, a function is helpful:

```python
def say_hello():
    print("hello")
```

```python
def say_hello_to(name):
    print("hello " + name)
```

Call the function like this:

```python
say_hello()
say_hello_to("Donald")
```

Functions

```python
def energy(signal):
    s2=signal**2
    energy=sum(s2)
    return(energy)
```

Call the function like this:

```python
signal=[2,4,6,0,0,1,2,4,0]
E=energy(signal)
print("Energy="+str(E))
```

Important: Make sure you get the indentation right, especially when you do cut and paste (so your code works).

Próxima sección

Let's code

Terminal: **jupyter notebook**

UNIVERSIDAD DE ANTIOQUIA
1 8 0 3

Preguntas

Thanks!

jcamilo.vasquez@udea.edu.co

UNIVERSIDAD DE ANTIOQUIA
1 8 0 3