

Introduction to Python for Psychological Research

MSc. Renato Paredes Venero
Laboratorio de Neurociencia Cognitiva Computacional
Departamento de Psicología

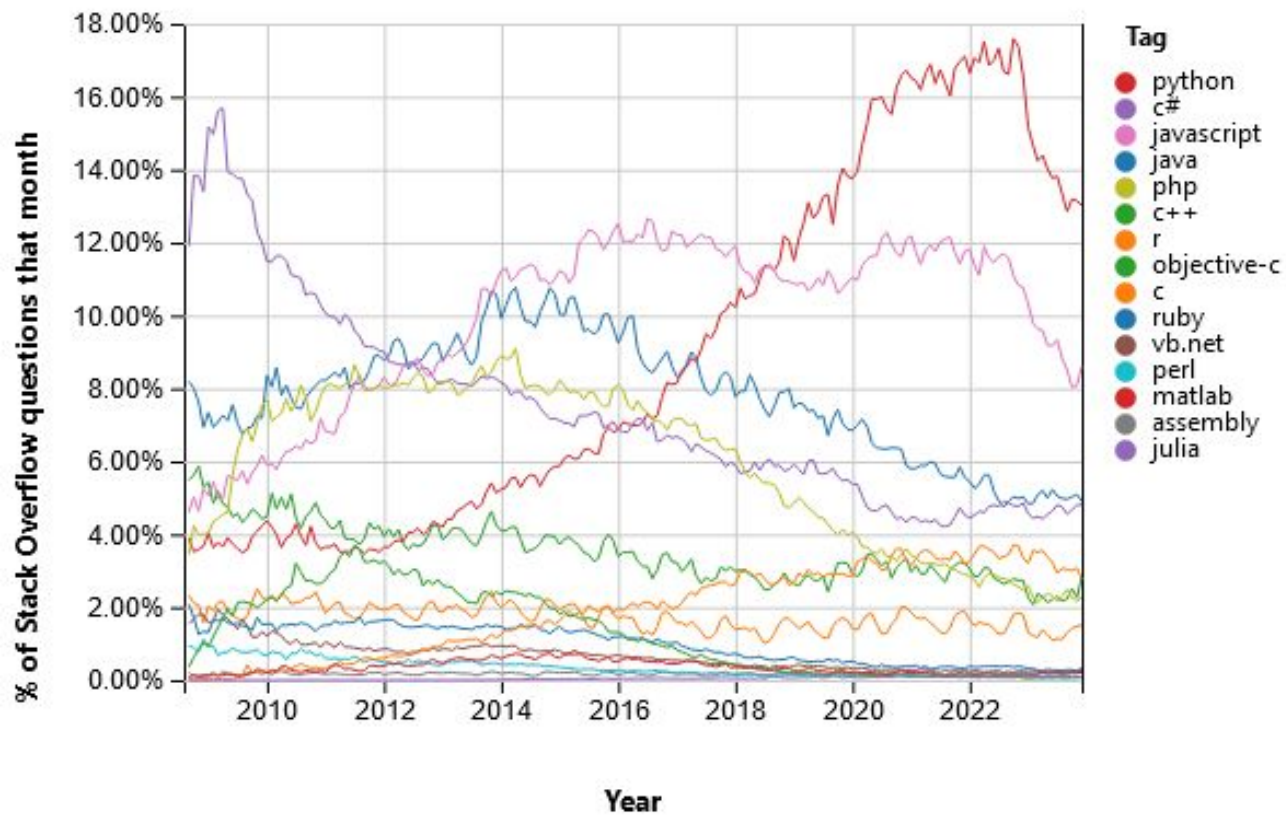


Why learn to program?

- Programming is an essential skill for modern psychological research.
- Programming allows researchers to quickly and efficiently complete specific tasks that would otherwise require hours.
- It is highly demanded in today's working world.



Tomado de: <https://www.bbva.com/en/computer-programming-moving-toward-a-gender-neutral-domain/>



Why is Python so popular?

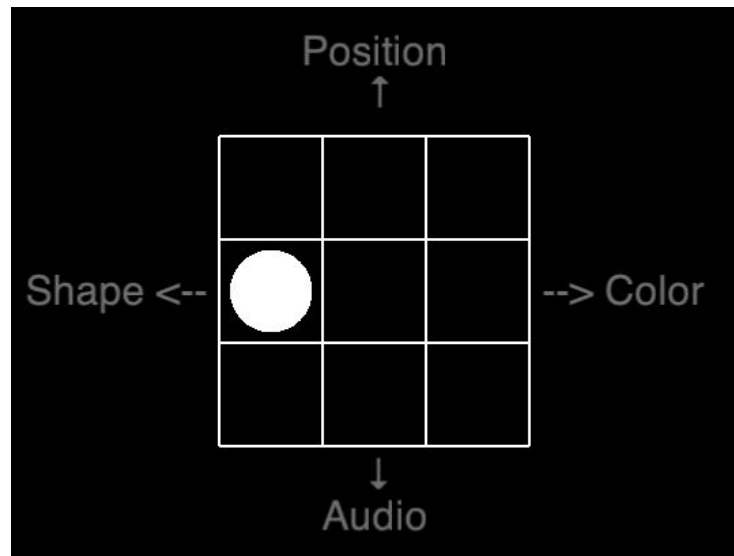
- Versatile general purpose language
- Easy access and learning
- Various software libraries
- Great support and community on the Internet
- Promotes open and reproducible science



**What can you do with Python
scientific stack?**

Data collection

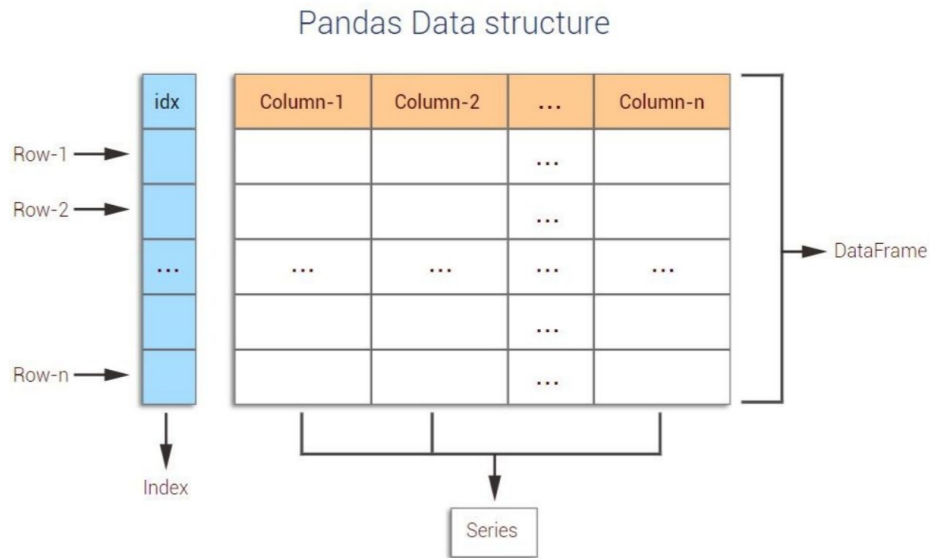
- It gives full control over all aspects of an experimental task.
- Allows the creation of online experiments.
- Packages: Psychopy, Pavlovia, oTree



Tomado de: <https://lindeloev.net/>

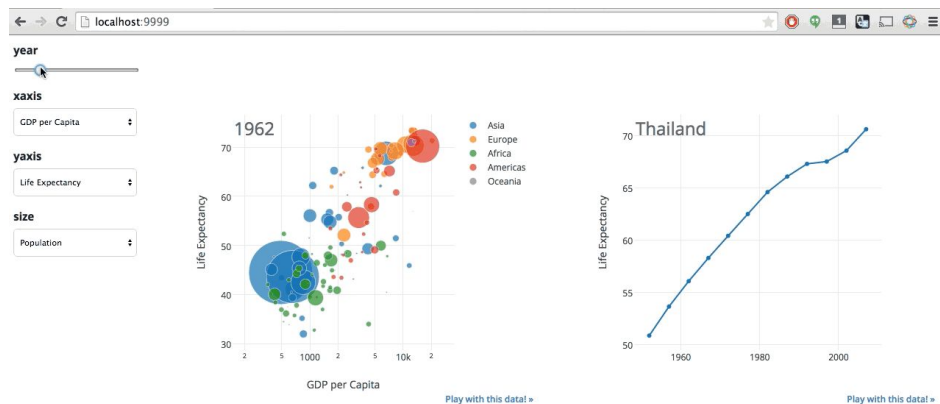
Data processing and organization

- It allows you to clean and organize data iteratively.
- It favors the automation of pre-processing of large amounts of data.
- Packages: Pandas



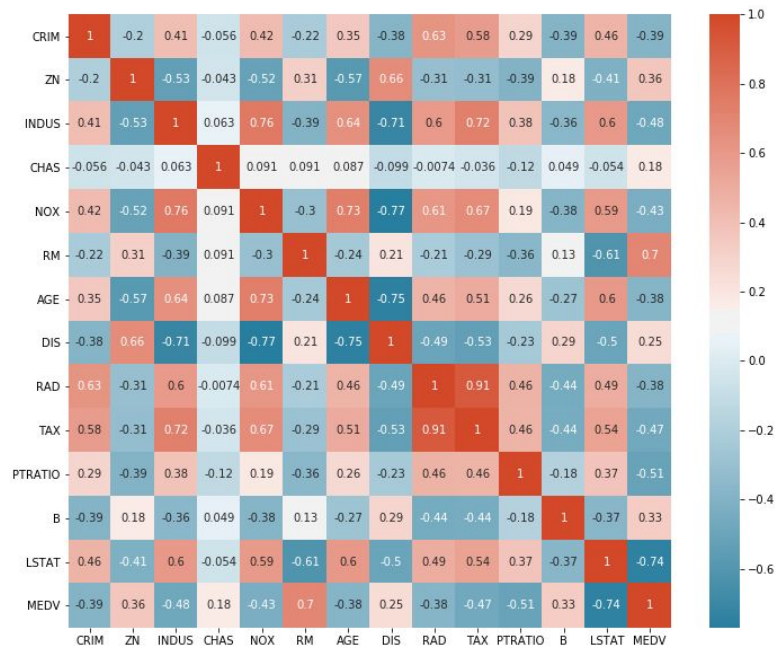
Visualización de datos

- It allows you to create personalized data visualizations aimed at different audiences.
- Visualizations can be linked directly to websites or other IT developments.
- Packages: Matplotlib, Seaborn, Bokeh



Data Analysis

- It has tools for descriptive and inferential statistical analysis.
- Enables the use of advanced analytics techniques such as machine learning.
- Packages: Numpy, Scipy, Statsmodels, **Scikit-learn**



Tomado de: <https://tinyurl.com/fypjuu5f>

Quantitative research in Psychology

- It has tools for the most used statistical analysis in psychology: Group comparisons, regression models, factor analysis, structural equation models.
- Packages: Pingouin, FactorAnalyzer, Semopy

pingouin

Textual data processing

- It has tools to process and analyze textual data from web pages, social networks and digital files.
- There are specific packages for natural language processing, sentiment analysis, thematic modeling, among others.
- Packages: NLTK, GENSIM, SpaCy

The logo for SpaCy, featuring the word "spaCy" in a blue, lowercase, sans-serif font. The "a" is stylized with a dot above it.

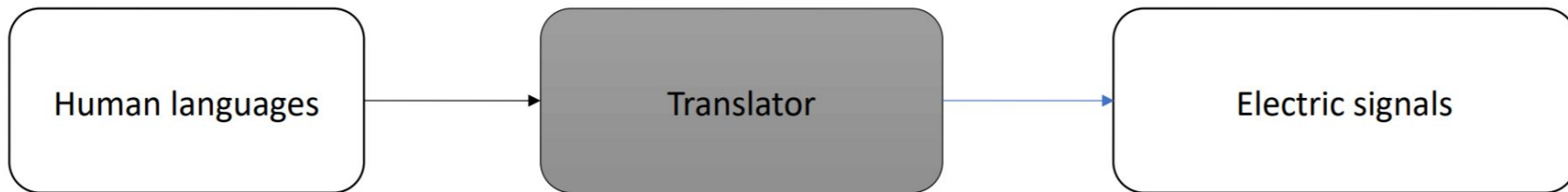
Neurobehavioral data processing

- It has tools to process and analyze neurobehavioral data from experimental studies with humans and animals.
- There are specific packages for electrophysiology, neuroimaging, eye-tracking, facial expressions, among others.
- Packages: MNE, NeuroKit2, Nilearn, BrainIAK, Py-Feat, PyRAT, PyTrack

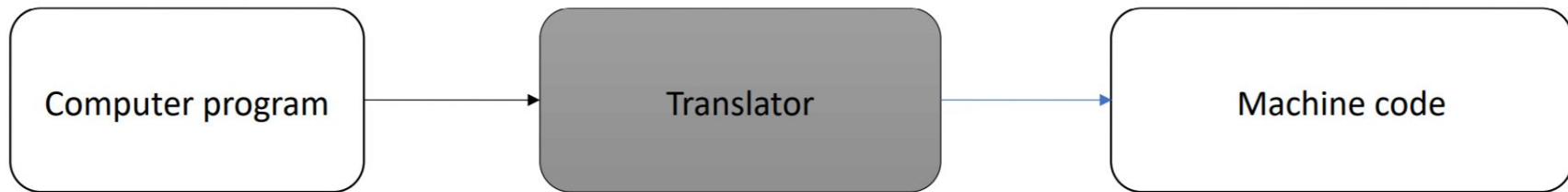


Basic notions of software development

What is a computer program?



What is a computer program?



How do these translators work?

- Compilers
 - They translate the entire program in one go.
 - Optimization is possible.
 - It is not very comfortable for exploration work.
- Interpreters
 - They translate line by line.
 - Much easier to "play with your data".
 - Lack of optimization can affect performance.

What translator does Python use?

- Python is an interpreted language.
- Therefore, speed may be affected.
 - There are optimization techniques and tools - Vectorization, Numba, etc.
 - Also extension modules in C.

How do I run my code?

- In the case of Python, it depends on the style of code you write:
 - Command line interface - CLI (e.g. Bash, CMD, IPython)
 - Scripts (e.g. ThesisCode.py)
 - Web-based execution environments (e.g. Google Colaboratory)

Google Colab

The screenshot shows a Google Colab notebook titled "I01c01_introduction_to_colab_and_python.ipynb". The browser address bar shows the URL: https://colab.research.google.com/github/tensorflow/examples/blob/master/courses/udacity_intro_to_tensorflow/blob/master/I01c01_introduction_to_colab_and_python.ipynb. The notebook interface includes a "Table of contents" sidebar on the left, a "Share" button, and a "Copy to Drive" button. The main area displays a Python code cell with the following code:

```
[ ] def HelloWorldXY(x, y):  
    if (x < 10):  
        print("Hello World, x was < 10")  
    elif (x < 20):  
        print("Hello World, x was >= 10 but < 20")  
    else:  
        print("Hello World, x was >= 20")  
    return x + y  
  
for i in range(8, 25, 5): # i=8, 13, 18, 23 (start, stop, step)  
    print("--- Now running with i: {}".format(i))  
    r = HelloWorldXY(i,i)  
    print("Result from HelloWorld: {}".format(r))  
  
[ ] print(HelloWorldXY(1,2))
```

Below the code cell, the text "Easy, right?" is displayed, followed by the text "If you want a loop starting at 0 to 2 (exclusive) you could do any of the following". A second code cell is partially visible at the bottom:

```
[ ] print("Iterate over the items. `range(2)` is like a list [0,1].")  
for i in range(2):
```

Recursos útiles

1. [Easy-to-read Python guide](#)
2. [Python official documentation](#)
3. [Python – Stack overflow](#)
4. [Common Python pitfalls](#)
5. [Community curated resources](#) – a list of lists
6. [Python for non-programmers](#) (sección ‘Tutorial and Websites’)