

Introducción

Dra. Yessenia Yari Ramos
ydyari@ucsp.edu.pe



Universidad Católica
San Pablo

Departamento de Ciencia
de la Computación

Contenido

❖ Introducción a LP

- Lenguajes de programación

❖ Python

- Características
- Historia
- Aplicaciones
- Evolución
- Instalación
- IDEs

Introducción a LP



Lenguajes de Programación

- Es un lenguaje de programación de alto nivel.
- Los lenguajes de alto nivel necesitan ser traducidos a lenguajes de bajo nivel antes de ser ejecutados.



Lenguajes de Programación

- Es un lenguaje de programación de alto nivel.
- Los lenguajes de alto nivel necesitan ser traducidos a lenguajes de bajo nivel antes de ser ejecutados.
- Los programas de alto nivel son:
 - cortos y fáciles de leer,
 - portables



Lenguajes de Programación

- Hay dos tipos de programas que traducen lenguajes de alto nivel a bajo nivel:



Lenguajes de Programación

- Hay dos tipos de programas que traducen lenguajes de alto nivel a bajo nivel:
 - Intérpretes
 - Compiladores



Python



Python

- Lenguaje de programación de alto nivel
- Lenguaje interpretado
- Fácil de implementar y entender.
- Usado en las aplicaciones web, el desarrollo de software, la ciencia de datos y el machine learning (ML)
-



Introducción

- Python es un lenguaje de programación de propósito general.
- Algunas características:
 - Poderoso
 - Flexible
 - Sencillo
 - Fácil de aprender
- Es un lenguaje de alto nivel (todo tipo de estructuras de datos, tanto numéricos como de texto).



Python

- Sintaxis muy clara y legible.
- Fuerte capacidad de introspección.
- Orientación a objetos intuitiva.
- Expresión del código procedimental.
- Altamente modular, soporta paquetes jerárquicos.
- Enfocado en el uso de excepciones para el manejo de errores.
- Tipos de datos dinámicos de muy alto nivel.
- Extensa biblioteca estándar (STL)
- Extensiones y módulos fácilmente escritos en C, C++ (o Java para Jython, o .NET para IronPython).

Historia

- Creado en 1991 por Guido van Rossum en los Países Bajos.
- El nombre es en honor al programa cómico de la televisión británica “Monty Python Flying Circus”.
- Python se desarrolla bajo una licencia de Open source.
- Es de código abierto, multi-plataforma es decir se adecua a diversos paradigmas de programación.
- Python se ha vuelto muy popular, es usado en algunos proyectos como Openstack, Blender, Google App Engine, Django, Jupyter, etc.



Historia

Top 10 Lenguajes de Programación más utilizados

Este es el ranking de los lenguajes de programación más usados en la actualidad, está basado en los datos del índice **TioBE** el cual (hay que aclarar) **no señala que lenguaje es mejor**, sino, que lenguaje se escribió en **mayor cantidad** durante el último mes del 2022.

#1



Python:



Es un lenguaje muy versátil ya que tiene múltiples áreas de aplicación como inteligencia artificial, Big Data y desarrollo web, es de código abierto y muy fácil de aprender..

#2



Lenguaje C:



Es uno de los primeros lenguajes y forma la base de otros más actuales como C++, C# o Java. Con él podemos desarrollar tanto aplicaciones como sistemas operativos.

#3



Java:



Es un lenguaje de propósito general y su ámbito de aplicación es súper amplio, es orientado a objetos y se utiliza en gran parte para crear aplicaciones y procesos en múltiples dispositivos.

#4



C++:



Surgió como extensión de C para que pudiese manipular objetos, es muy utilizado en bases de datos, navegadores web, compiladores o videojuegos.

#5



C#:



Es un lenguaje muy versátil, creado por Microsoft, similar a C pero orientado a objetos. Es muy usado en la industria de los juegos, robótica, impresión 3D, IoT y desarrollo móvil / web.

#6



Visual Basic:



Es un lenguaje orientado a objetos desarrollado por Microsoft. El uso de Visual Basic agiliza y simplifica la creación de aplicaciones .NET con seguridad de tipos.

#7



JavaScript:



Es uno de los preferidos por ser uno de los más poderosos y flexibles, sirve para todo; aplicaciones web, servidores, aplicaciones móviles, no necesita compilación ninguna

#8



Assembly:



Es un lenguaje de programación de bajo nivel. Consiste en un conjunto de instrucciones básicas para los computadores, microprocesadores y otros circuitos integrados programables

#9



SQL:



Es un lenguaje de programación utilizado para el manejo de información en las bases de datos relacionales, además, permite realizar operaciones de acceso y manipulación de la información.

#10



PHP:



PHP es uno de los clásicos; es un lenguaje de uso general que se adapta especialmente al desarrollo web. PHP se encuentra instalado en más de 20 millones de sitios web.



/Tecsify



@Tecsify



@Tecsify



www.Tecsify.com/blog



/Tecsify



Tecsify



@Tecsify



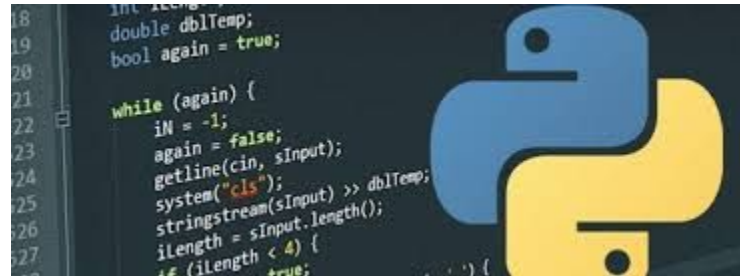
Aplicaciones

- Aplicaciones de escritorio.
- Aplicaciones web.
- Análisis de datos.
- Administración de servidores.
- Seguridad y análisis de penetración.



Aplicaciones

- Cómputo en la nube.
- Cómputo científico.
- Análisis de lenguaje natural.
- Visión artificial.
- Animación, videojuegos e imágenes generadas por computadora.
- Aplicaciones móviles.



Evolución

- Actualmente existen dos versiones de Python cuyo código no es compatible.
- Python 3 es una versión revisada del lenguaje, la cual fue publicada en 2009 y que incluye modificaciones y mejoras que lo hacen incompatible con código de versiones previas; mientras que Python 2 es una versión que es compatible con código antiguo.



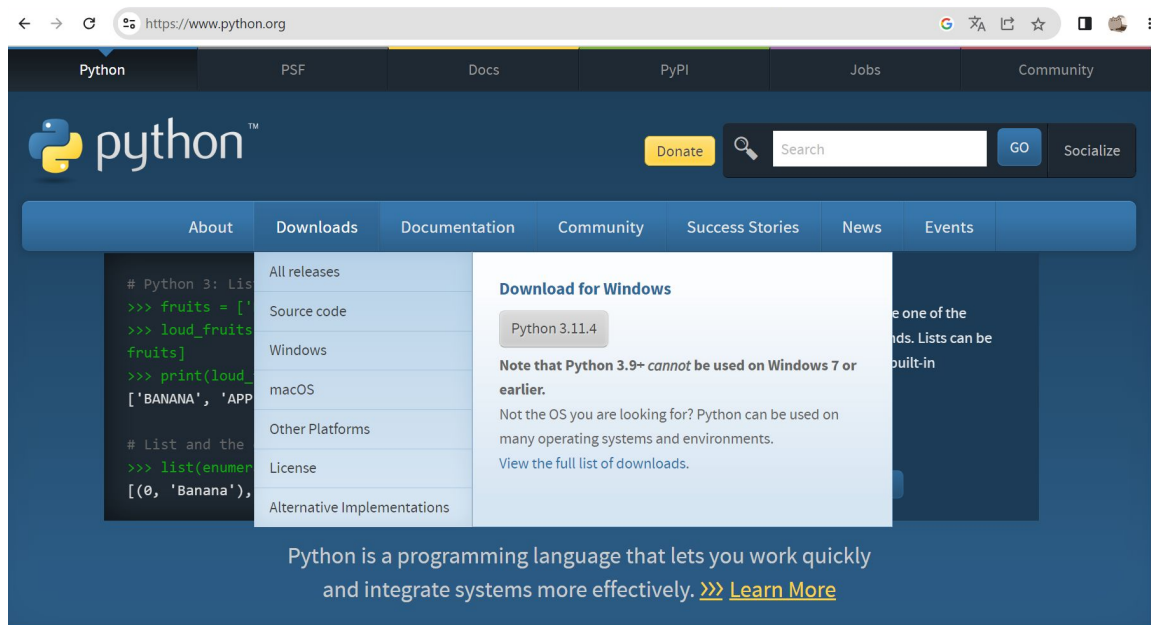
Evolución

- Python 2.7 es una versión "de transición" que permite usar algunos elementos sintácticos tanto de Python 2 como de Python 3.
- Python 3 presenta mejoras notables con respecto a Python 2, sin embargo aún existe mucho código legado compatible con Python 2.
- Se prevé que la publicación y soporte de nuevas versiones de Python 2 cese a partir del año 2020.



Instalación

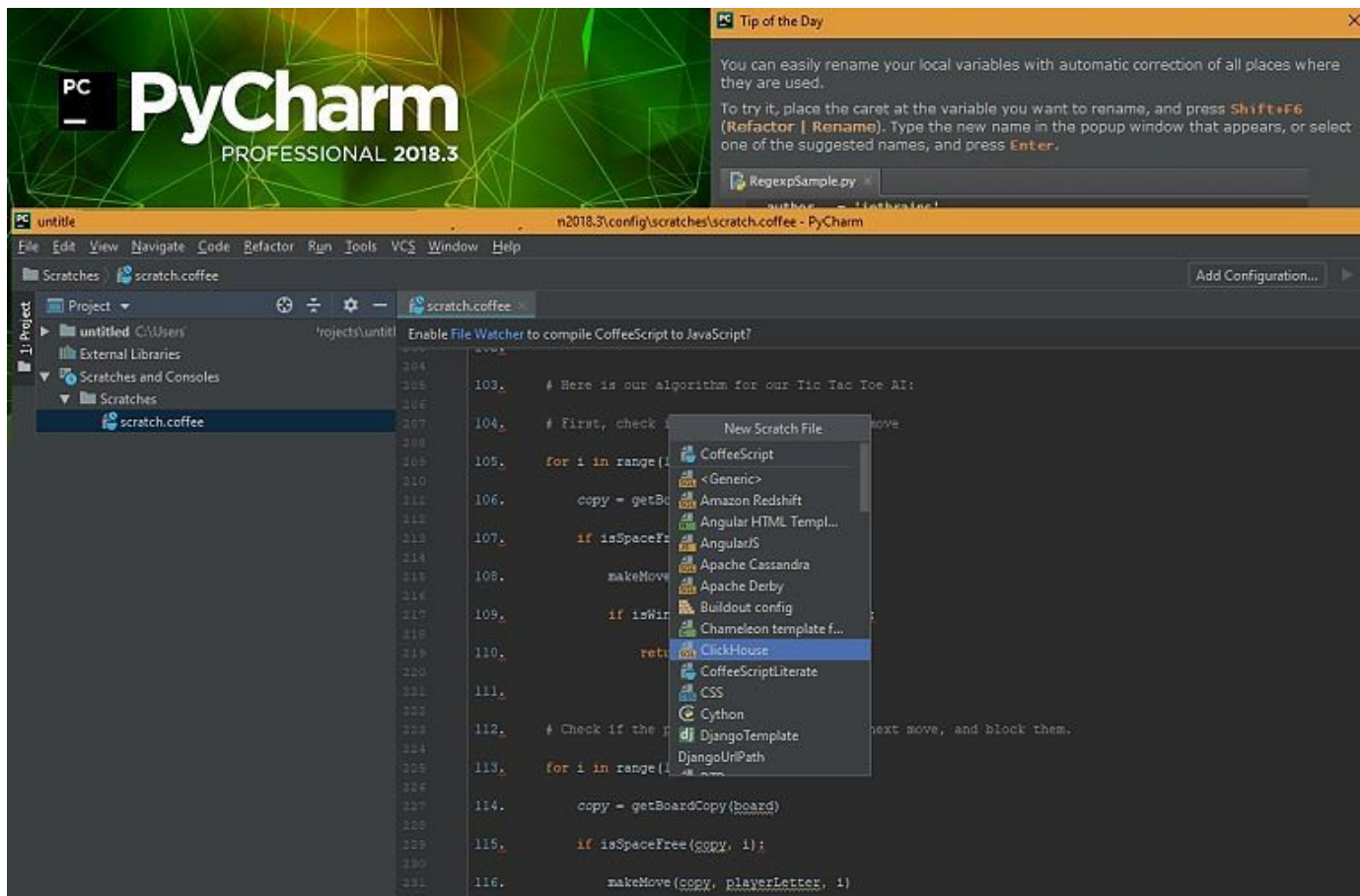
Página: <https://www.python.org/>



Instalación

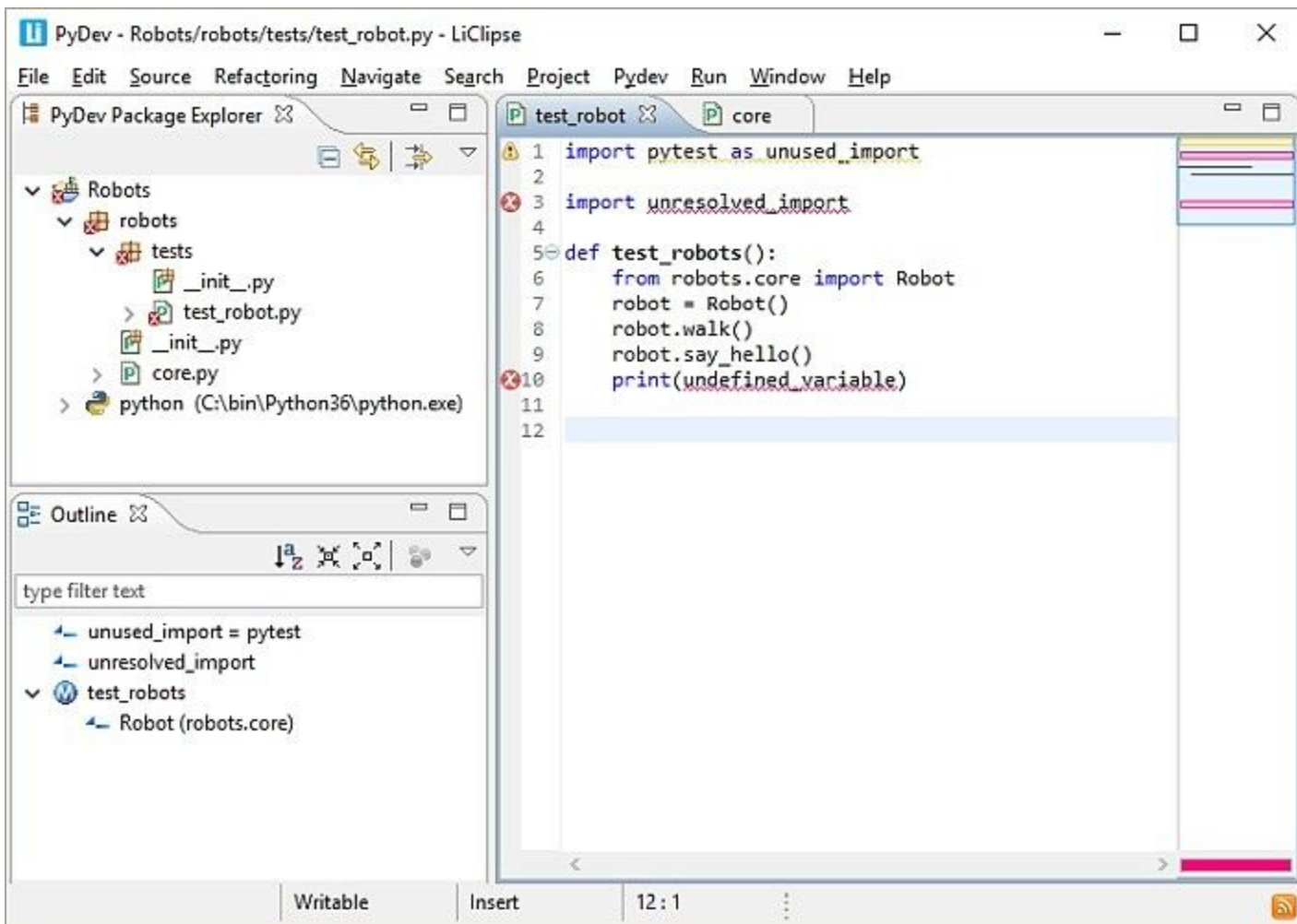
Link de ayuda:

<https://python-para-impacientes.blogspot.com/2017/02/instalar-python-paso-paso.html>



PyCharm

- PyCharm de JetBrains es muy popular.
- Usado por desarrolladores de Python en todos los niveles.
- Editor de código completo (ofrece sugerencias)
- Sangría automática,
- Detectar texto duplicado
- Comprobar si hay errores en los resaltes.
- Desventaja? Pago



PyDev

- PyDev ofrece una interfaz útil.
- Viene con Eclipse, que está disponible por separado.
- Incluyen la finalización automática de código,
- Sangrados inteligentes y sangrados de bloque,
- Resaltando los corchetes coincidentes,
- Búsqueda simple y un depurador.
- Código libre

~/workspace/spyder - Spyder 3.3.2.dev0 (Python 3.6)

File Edit Search Source Run Debug Cerebras PyM Projects Tools View Help

Symbol finder

Project explorer

- spyder
 - continuous_integration
 - img_src
 - requirements
 - rope_profiling
 - scripts
 - spyder
 - __pycache__
 - app
 - __pycache__
 - tests
 - _init_.py
 - cli_options.py
 - mac_stylesheet.qss
 - mainwindow.py
 - restart.py
 - start.py
 - tour.py
 - config
 - defaults
 - fonts
 - images
 - locale
 - plugins

Outline

- mainwindow.py
 - if os.name == 'nt':
 - try:
 - except ImportError:
 - try:
 - except Exception:
 - if hasattr(OL, 'AA_EnableHighDpi')
 - if PYQT5:
 - else:
 - if not running_under_pytest():
 - else:

Editor - /home/pizetta/workspace/spyder/spyder/app/mainwindow.py

```

3080 sys.exit = fake_sys_exit
3081
3082 # ----Monkey patching sys.excepthook to avoid crashes in PyQt 5.5+
3083 if PYQT5:
3084     def spy_excepthook(type_, value, tback):
3085         sys.__excepthook__(type_, value, tback)
3086         sys.excepthook = spy_excepthook
3087
3088 # Removing arguments from sys.argv as in standard Python interpreter
3089 sys.argv = []
3090
3091 # Selecting Qt4 backend for Enthought Tool Suite (if installed)
3092 try:
3093     from enthought.etsconfig.api import ETSConfig
3094     ETSConfig.toolkit = 'qt4'
3095 except ImportError:
3096     pass
3097
3100 import qdarkstyle
3101 app.setStyleSheet(qdarkstyle.load_stylesheet_pyqt5())
3102
3103 return app
3104
3105
3106 class Spy(object):
3107     """
3108     Inspect Spyder internals
3109
3110     Attributes:
3111         app      Reference to main QApplication object
3112         window   Reference to spyder.MainWindow widget
3113     """
3114     def __init__(self, app, window):
3115         self.app = app
3116         self.window = window
3117
3118     def __dir__(self):
3119         return list(self.__dict__.keys()) + \

```

History log

history.py

```

# -*- coding: utf-8 -*-
# *** Spyder Python Console History Log ***

## ---(Thu Sep 13 21:42:42 2018)---
runfile('/home/pizetta/workspace/spectrometer/pymr2/pymr2/gui/manager.py', wdir='/home/pizetta/workspace/spectrometer/pymr2/pymr2/gui')

```

Variable explorer

Name	Type	Size	Value
------	------	------	-------

Static code analysis

enforce-asynchron ☐ Analyze

Results for /home/pizetta/workspace/spectrometer/hardware-api/ci/mag_spec/peripheral/

- Convention (15 messages)
- Warning (0 messages)
- Warning (0 messages)
- Error (0 messages)

Breakpoints

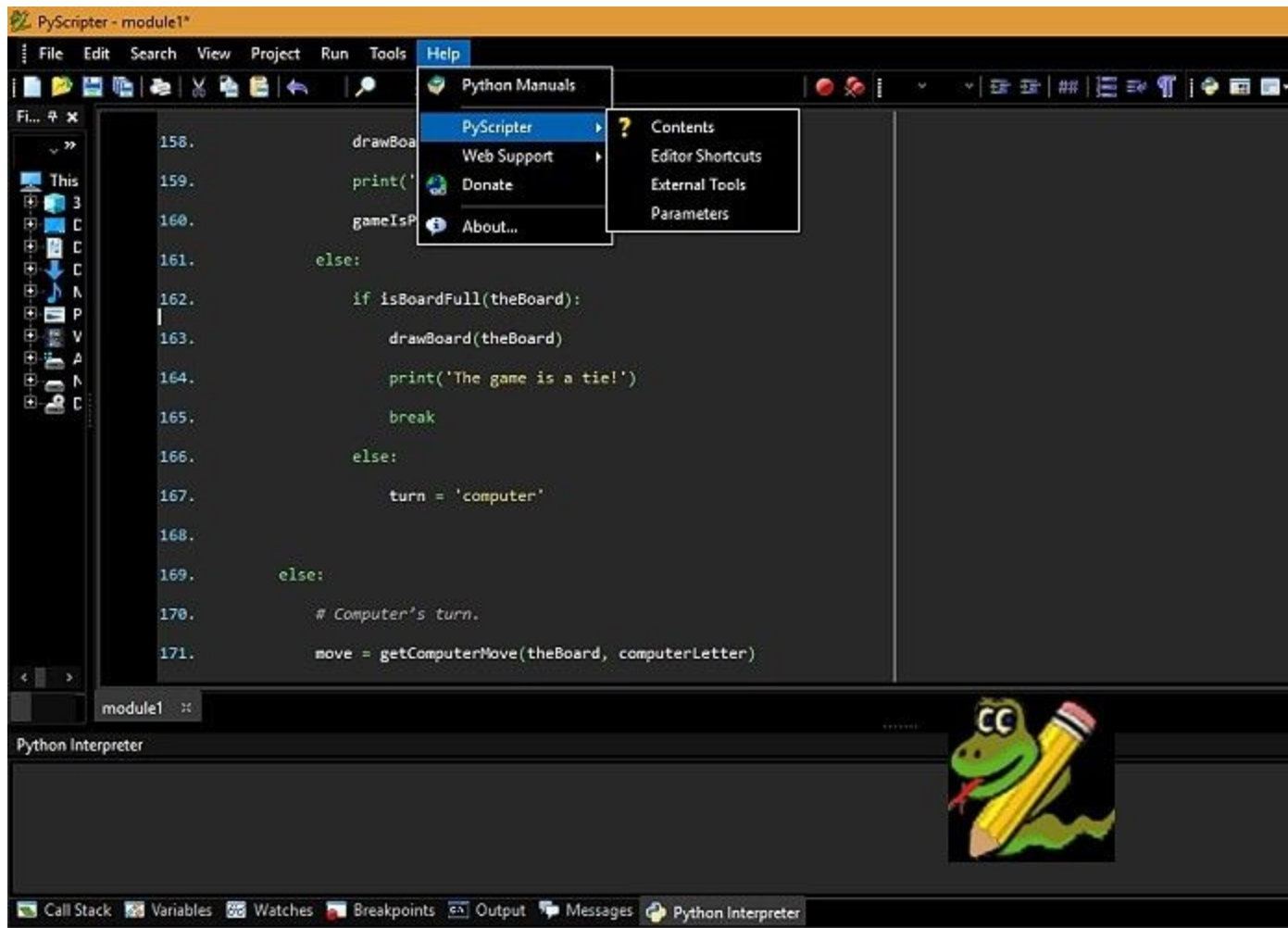
File	Line	Condition
------	------	-----------

Breakpoints Python console

Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 3106 Column: 1 Memory: 47 %

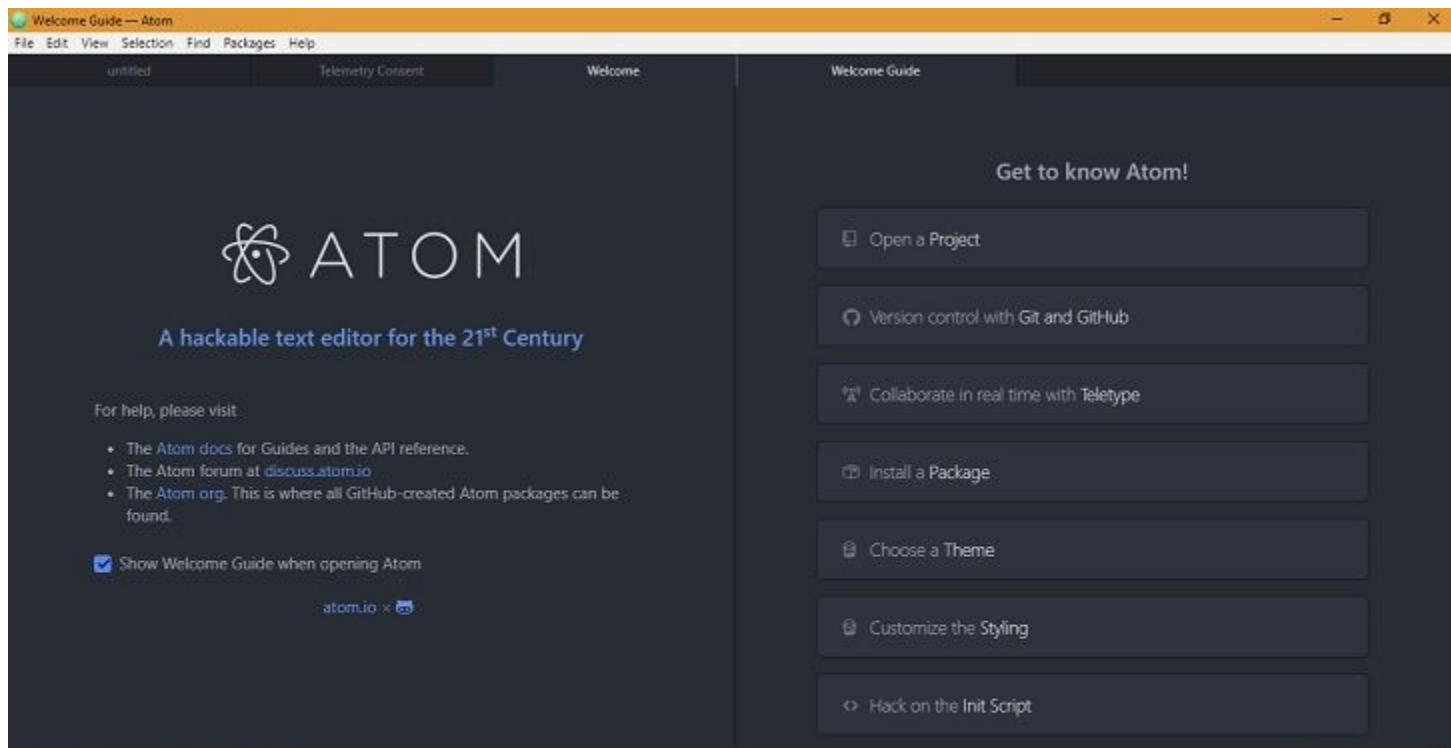
Spyder

- Spyder es un poderoso entorno científico escrito en Python, para Python, y diseñado por y para científicos, ingenieros y analistas de datos.
- Es un software ligero, escrito en Python y disponible de forma gratuita bajo licencia del MIT
- Funcionalidad avanzada de edición, análisis, depuración, etc
- Además, Spyder ofrece integración integrada con muchos paquetes científicos populares, incluyendo NumPy, SciPy, Pandas, IPython, QtConsole, Matplotlib, SymPy, y más.
- Spyder también puede ser utilizado como una biblioteca de extensiones de PyQt5, permitiéndole construir sobre su funcionalidad e integrar sus componentes.



PyScripter

- Es completamente gratuito y de código abierto y extremadamente ligero
- El compilador viene con todas las características básicas de una plataforma de codificación y puede satisfacer las necesidades de principiantes y expertos.
- Todos los programas se ven en la ventana de ejecución, y se puede ejecutar varias veces para comprobar si hay errores.



Atom

- Atom es un editor de texto multiplataforma creado por GitHub.
- Atom soporta la instalación de paquetes que añaden funcionalidades extra al programa
- Alta capacidad de personalización.

File

Edit

Selection

View

Go

Debug

Terminal

Help

Ejercicio.py - Visual Studio Code

EXPLORER

hola.java

aspecto1.java

poa2.java

prueba3.java

cuenta.java

Ejercicio.py

Ejercicio2.py

OPEN EDITORS

localOtsu.py D:\Te...

poa1.java D:\Ejerci...

hola.java D:\Ejerci...

aspecto1.java D:\E...

poa2.java D:\Ejerci...

prueba3.java D:\Ej...

cuenta.java D:\Ejer...

Ejercicio.py D:\Tesi...

Ejercicio2.py D:\Te...

otsu.py D:\TesisIng

NO FOLDER OPENED

You have not yet opened a folder.

Open Folder

D: > TesisIng > Ejercicio.py

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 from matplotlib.ticker import NullFormatter # useful for `logit` scale
5
6 # Fixing random state for reproducibility
7 np.random.seed(19680801)
8
9 # make up some data in the interval ]0, 1[
10 y = np.random.normal(loc=0.5, scale=0.4, size=1000)
11 y = y[(y > 0) & (y < 1)]
12 y.sort()
13 x = np.arange(len(y))
14
15 # plot with various axes scales
16 plt.figure(1)
17
18 # linear
19 plt.subplot(221)
20 plt.plot(x, y)
21 plt.yscale('linear')
22 plt.title('linear')
23 plt.grid(True)
24
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

1: powershell

Visual Studio Code

- Creado a mediados de 2015 por Microsoft para Windows, OS X y Linux .
- Soporta un gran número de lenguajes de programación e incluye muchas herramientas que no son accesibles a través de la interfaz del menú del usuario.
- Gratuito.
- Resaltado de sintaxis, asociaciones de teclas personalizables, snippets para tener un código escrito y pegarlo con una combinación de teclas,
- Bracket matching para comprobar que las llaves estén siempre convenientemente cerradas
- Soporte para Git.


Practica00.ipynb - Colaboratory

+

← → ↻

https://colab.research.google.com/drive/124uqHt6KfDFx4eOuxzsQMqBI_...


🔖 🌐 🏠 ⋮

 Practica00.ipynb ☆

Comentario

Compartir

⚙️



Archivo Editar Ver Insertar Entorno de ejecución Her

⋮

+ Código + Texto

✓ RAM
Disco

🔍 0 s

▶️

1 print("Hola Mundo")

Hola Mundo

↑ ↓ 🔗 💬 ⚙️ 📄 🗑️ ⋮

📁

<>

☰

📄

✓ 0 s completado a las 21:41

🟢 ✕

Google Colab

- Colaboratory, o "Colab"
- Permite a cualquier usuario escribir y ejecutar código arbitrario de Python en el navegador.
- Es especialmente adecuado para tareas de aprendizaje automático, análisis de datos y educación.
- Desde un punto de vista más técnico, Colab es un servicio de cuaderno alojado de Jupyter que no requiere configuración y que ofrece acceso sin coste adicional a recursos informáticos, como GPUs.



Gracias!