

Python para Síntesis de Redes Activas

Lucas Heraldo Duarte
Dr. Ing. Pablo Alejandro Ferreyra

Laboratorio de Circuitos y Sistemas Robustos
Facultad de Ciencias Exactas Físicas y Naturales - Universidad Nacional de Córdoba



Universidad
Nacional
de Córdoba

Ag. 2021

Contenidos

1 Introducción

- Objetivos
- ¿Qué es Python?
- ¿Por qué Python?
- ¿Cómo programar en Python?

2 Instalación

- Python 3.8 - en Linux (Huayra)
- Python 3.8 - win. (64-bit)
- Jupyter

3 Practicas

- Reutilizando código
- Ecuaciones simbólicas de redes

objetivos

Objetivos

- Instalar Python3.8 y Jupyter.
- Instalar paquetes numpy, sympy, matplotlib.
- Presentar documentación específica de Python.
- Practicas:
 - Reutilización de código.
 - Ecuaciones simbólicas de redes.

¿Qué es Python?

Python es un **lenguaje de programación potente y fácil** de aprender. Tiene **estructuras de datos de alto nivel eficientes** y un simple pero **efectivo sistema de programación orientado a objetos**. La elegante sintaxis de Python y su tipado dinámico, junto a su naturaleza interpretada lo convierten en **un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en muchas áreas**, para la mayoría de plataformas.



¿Por qué Python?

La primera y más importante razón es la versatilidad, que le permite funcionar perfectamente en proyectos con **tecnologías de punta**.

El lenguaje tiene **muchas bibliotecas y paquetes**, lo que lo convierte en una gran alternativa para iniciar proyectos con **bases sólidas**.

Otra razón crucial de su eterna popularidad es su **facilidad de uso**. Lo fácil que es para los principiantes comenzar a programar con Python, y es extremadamente poderoso y útil para los ingenieros veteranos.

Ventaja

Un lenguaje para autodidactas, su pagina oficial^a tiene toda la documentación necesaria para aprender a programar.

^a<https://docs.python.org/es/3.8/tutorial/index.html>

¿Cómo programar en Python?

La comunidad Python esta bastante consolidada y madura, debido a esa particularidad, en internet se encuentran innumerables tutoriales, cursos, vídeos, y más donde se enseña a programar.

En este aspecto se recomienda ver los siguientes enlaces:

- <https://claudiovz.github.io/scipy-lecture-notes-ES/index.html>
- <https://docs.python.org/3.8/>
- <https://www.tutorialspoint.com/python/index.htm>
- <https://www.tutorialspoint.com/sympy/index.htm>

Instalación en Linux

Huayra

Para instalar la versión 3.8 python en Linux ejecutamos:

```
1 wget https://www.python.org/ftp/python/3.8.2/Python-3.8.2.tgz
2 tar xzf Python-3.8.2.tgz
3
4 sudo apt install build-essential libreadline-gplv2-dev
   libncursesw5-dev libssl-dev libsqlite3-dev tk-dev
   libgdbm-dev libc6-dev libbz2-dev libffi-dev zlib1g-dev
5 cd Python-3.8.2/
6 sudo ./configure --enable-optimizations
7 sudo make altinstall
```

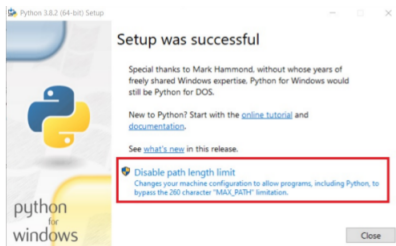
Instalación para win.)

64-bit

Descargar e instalar desde el **sitio oficial** de Python:

<https://www.python.org/>.

Instalar teniendo en cuenta las **opciones marcadas** en las capturas siguientes:



Gestor de paquetes pip

Instalación

Para actualizar el gestor de paquetes en linux:

```
1 sudo pip3.8 install --upgrade pip
```

El gestor pip nos permitirá manejar los paquetes y herramientas.

```
1 sudo pip3.8 install nom_paquete // instala paq.  
2 sudo pip3.8 uninstall nom_paquete // desinstala el paq.  
3 sudo pip3.8 install - -upgrade nom_paquete // actualiza  
4 sudo pip3.8 show nom_paquete // muestra la version instalada  
5 sudo pip3.8 list // muestra todos los paquetes instalados
```

Vamos a usar:

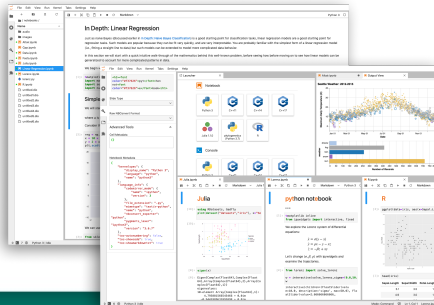
```
1 sudo pip3.8 install numpy  
2 sudo pip3.8 install matplotlib  
3 sudo pip3.8 install sympy
```

Jupyter

Instalación

¿Qué es Jupyter?

Jupyter es un entorno web de desarrollo interactivo que permite desarrollar código Python de manera dinámica. Integra en un mismo documento tanto bloques de código como texto, gráficas o imágenes. Utiliza la extensión .ipynb
Jupyter incluye a Jupyter Notebook y JupyterLab.



Jupyter

Instalación

Instalación:

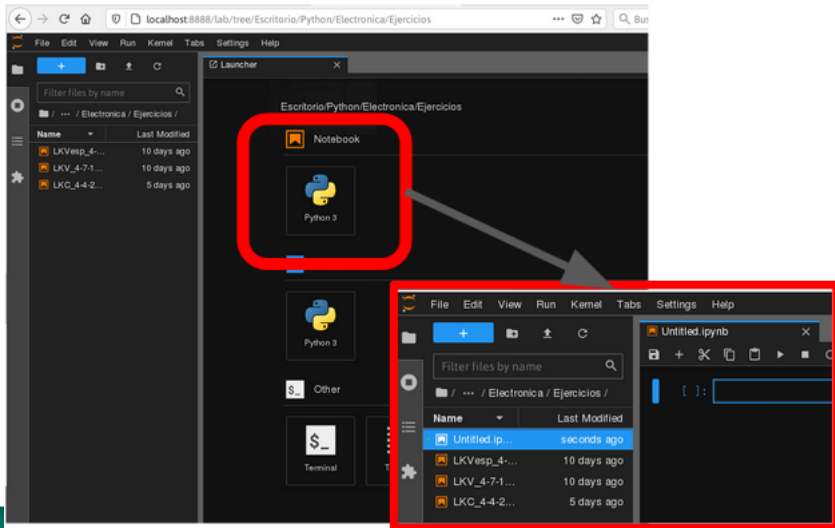
```
1 sudo pip3.8 install jupyterlab // instala Jupyter
2 sudo pip3.8 install PyQt5 // opcional para usar %matplotlib
   qt5 en Jupyter
```

Ejecución:

```
1 jupyter lab // Ejecuta JupyterLab en el navegador por
   defecto.
2 jupyter notebook - -ip ip_del_servidor - -no-browser - -
   port=8080 // Ejecuta un servidor de Jupyter con la ip y
   el puerto dados. Devuelve un enlace para acceder al
   mismo remotamente.
```

Usando Jupyter

Python3.8



PRACTICAS

Reutilizando Código

¿Qué se quiere abordar?

- Observar sintaxis de Python.
- Utilizar a la comunidad Python.
- Reutilizar funciones y adaptarlas.
- Crear una clase, guardarla e importarla.
- Subir código a GitHub
- Mostrar caso de aplicación en Redes Activas.

Lista de alumnos

Reutilizando código

Se quiere crear un 'diccionario' $\{[llave] : [valor], \dots, len(diccionario)_{llaves}\}$
el [valor] puede ser un vector de valores `valor[0]` `valor[1]` ...

Funciones:

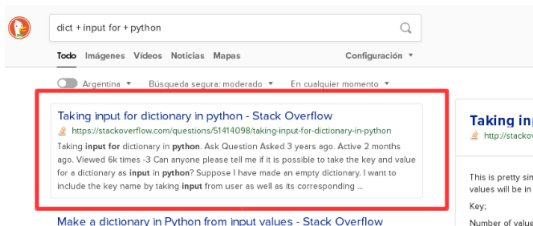
- Crear 'diccionario' con bucle for e input.
- Abrir tabla de datos en .txt
- Crear una lista de personas y elegir al azar una.
- Crear un registro de puntaje.
- Crear grupos random de personas.

Reutilizando Código

Lista de alumnos

La ventaja de python es la gran comunidad que hay en internet. Si se **busca en la web** se encuentra en **stack overflow** un código que nos un punto de inició para el propio.

<https://stackoverflow.com/questions/51414098/taking-input-for-dictionary-in-python>



The screenshot shows a search bar with the text "dict + input for + python". Below the search bar, there are tabs for "Todo", "Imágenes", "Videos", "Noticias", and "Mapas", along with a "Configuración" link. Below the tabs, there are filters for "Argentina", "Búsqueda segura: moderado", and "En cualquier momento". A red box highlights a search result titled "Taking input for dictionary in python - Stack Overflow". The result includes a link to the question, a brief description of the question, and a snippet of the answer. To the right of the red box, there is a partial view of another search result titled "Taking input for dictionary in python - Stack Overflow".

dict + input for + python

Todo Imágenes Videos Noticias Mapas Configuración

Argentina Búsqueda segura: moderado En cualquier momento

Taking input for dictionary in python - Stack Overflow
🔗 <https://stackoverflow.com/questions/51414098/taking-input-for-dictionary-in-python>
Taking **input** for dictionary in **python**. Ask Question Asked 3 years ago. Active 2 months ago. Viewed 6k times -3 Can anyone please tell me if it is possible to take the key and value for a dictionary as **input** in **python**? Suppose I have made an empty dictionary. I want to include the key name by taking **input** from user as well as its corresponding ...

Taking input for dictionary in python - Stack Overflow
🔗 <http://stackoverflow.com/questions/51414098/taking-input-for-dictionary-in-python>
This is pretty simple. The values will be in a dictionary.
Key;
Number of values;

Python

Librerías

El código que se encuentra es el siguiente:

```
1 def inputDict(numKeys=10):  
2     #if you want you could make numKeys an input too(numKeys  
   = int(input("Numkeys?")))  
3     desiredDict = {}  
4     #ask key and value for the numKeys  
5     for nameValuePair in range(numKeys):  
6         key = input("NextKey:\n")  
7         numVals = int(input("How many values for that key?\n  
   "))  
8         valList = []  
9         for value in range(numVals):  
10             valList.append(input("Next value\n"))  
11             desiredDict[key] = valList  
12     return desiredDict
```


En Jupyter

Ejecutar en terminal: `jupyter lab`

Crear archivo `.ipynb`

Pegar el código en una celda.

Para crear **nueva celda 'a'**, para **eliminarla 'd,d'**

Ejecutar con **shift+Enter**

Llama a la función.

Introducir datos.

```
[1]: def inputDict(numKeys=10):  
    #if you want you could make numKeys an input too(numKeys = 10)  
    desiredDict = {}  
    #ask key and value for the numKeys  
    for nameValuePair in range(numKeys):  
        key = input("NextKey:\n")  
        numVals = int(input("How many values for that key?\n"))  
        valList = []  
        for value in range(numVals):  
            valList.append(input("Next value\n"))  
        desiredDict[key] = valList  
    return desiredDict  
  
[*]: sintesis=inputDict(15)  
NextKey:  
Anibal Troilo
```

Abrir archivo .txt

Modificar Código

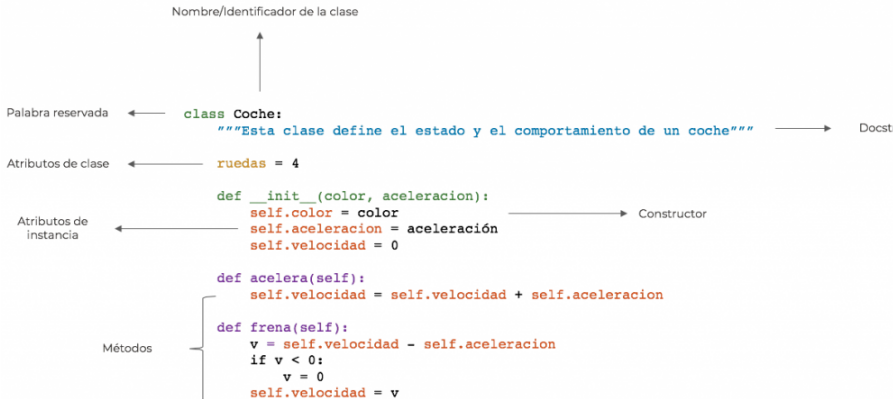
Del Código encontrado, **interpretándolo y viendo que métodos se utilizan**, se modifica el código a:

```
1
2 def abrir_datos(): # nombre, alias, correo
3     "Abrir un archivo que tiene guardada la info [key], [Val
4     [0]], [Val[1] ... ]"
5     vec = []
6     dicc = {}
7     f = open(input("nombre")+'.txt', 'rt')
8     for line in f:
9         vec=line.strip('\n').split(', ')
10        dicc[str(vec[0])]=list(vec[1:])
11    f.close()
12    print(dicc)
13    Planilla_=dict(dicc)
14    return Planilla_
```

Crear clases

Reutilizando Código

La sintaxis de una clase viene dada por:



Crear una Clase

La clase a crear debe tener como **atributo el diccionario**, y como **métodos todas las funciones** que se quería realizar.

Suele ser práctico crear **primero todas las funciones y luego crear la clase**.

Para utilizar la **clase desde otro archivo** hay que **guardar** el código como "nombre".py

Para **importarlo** hay que ejecutar: `import "nombre"`.

El "nombre" es **distinto al nombre de la clase**.

Código en GitHub

repositorio

El código se subió al repositorio de GitHub.

link del Código

El código .py donde se guarda la clase: link
Input y output de Jupyter importando y utilizando la clase: link

Aprender a usar GitHub

Es muy recomendable tener GitHub y usarlo. Una guía rápida:
<https://docs.github.com/en/get-started>

Ejemplo: Amplificador RF

Trabajo Practico de Analógica III

Un ejemplo donde se utilice la programación para abrir y manipular archivos es en el caso del **diseño de un amplificador RF**.

Como primer paso para comenzar el diseño de un amplificador de radiofrecuencia de 1,8 GHz es **utilizar el datasheet del fabricante para determinar el funcionamiento** deseado del dispositivo amplificador, siendo el transistor BFP640 (de Infineon).

Dado que este transistor tiende a oscilar fácilmente se elige cuidadosamente la **polarización para obtener un punto seguro de estabilidad**. Para ello se usan las **librerías .s2p** de varias polarizaciones recomendadas por el fabricante y, **con los parámetros S a distintas frecuencias**. Se puede **calcular los coeficientes de Rollet** y luego **descartar los coeficientes menores a 1** y se elige uno que nos garantizara una cómoda estabilidad.

Ejemplo: Amplificador RF

Trabajo Practico de Analógica III

A esto se lo hace a través de un código simple de programación para facilitar el manejo de datos, por ejemplo:

```
data=importdata(v1,"\t");
%importa la data de .s2p del modelo
%data tiene el mismo formato que V1

for(j=9:(length(data)-9))
%Solo la informacion numerica necesaria
datanum=str2num(char(data(j)));
%le cambia el formato de char a double

if (datanum(1)==frec)
%a la frecuencia seleccionada calcula los parametros S
S11= datanum(2)*exp(i*datanum(3)*C);
S21= datanum(4)*exp(i*datanum(5)*C);
S12= datanum(6)*exp(i*datanum(7)*C);
S22= datanum(8)*exp(i*datanum(9)*C);
%con los parametros calcula delta y K para la estabilidad
Delta = S11*S22-S12*S21;
K=(1-abs(S11)^2-abs(S22)^2+abs(Delta)^2)/(2*abs(S12*S21));
bandera=1;

FIN(n++,:)= [K,iv];
% FIN guarda el K y el valor del for, es decir, la fila de V1 "todos.txt"
%donde esta el nombre del .s2p
endif
```

Ejemplo: Amplificador RF

Trabajo Practico de Analógica III

Facilitando la elección de la polarización del transistor y calculando los parámetros de diseño.

Resultados

Frecuencia de trabajo [GHz]: 1.8

```
1) nombre: BFP640_VCE_1.0V_IC_50mA.s2p  valor de k: 1.807159e+00
2) nombre: BFP640_VCE_1.0V_IC_48mA.s2p  valor de k: 1.782708e+00
3) nombre: BFP640_VCE_1.0V_IC_42mA.s2p  valor de k: 1.576110e+00
4) nombre: BFP640_VCE_1.5V_IC_50mA.s2p  valor de k: 1.487335e+00
5) nombre: BFP640_VCE_1.5V_IC_48mA.s2p  valor de k: 1.403544e+00
6) nombre: BFP640_VCE_1.0V_IC_37mA.s2p  valor de k: 1.260597e+00
....
```

Elija el num de polarizacionLa polarizacion numero: 6

Delta = 0.1387 - 0.1822i

K = 1.2606

Delta 2.289562e-01 < -5.271403e+01 y Coeficiente de Rollet: 1.260597e+00

Es incondicionalmente estable

Maxima ganancia estable MSG: 1.978994e+01

parametros de diseno:

PRACTICAS

Ecuaciones simbólicas de redes

¿Qué se quiere abordar?

- Utilizar Librería SymPy.
- Representar redes eléctricas.
- Resolver Ejercicios Prácticos.

Librería SymPy

<https://www.sympy.org/es/index.html>

SymPy es una **biblioteca de Python** para **matemáticas simbólicas**. Su propósito es llegar a ser un **sistema de álgebra por computadora (CAS) completo** manteniendo el **código tan simple como sea posible** para poder ser legible y extensible de manera fácil.

SymPy tiene una amplia gama de características aplicables en el campo de la aritmética simbólica básica, cálculo, álgebra, matemáticas discretas, física cuántica, etc.

SymPy es capaz de formatear los resultados en una **variedad de formatos incluyendo LaTeX**, MathML, etc.

Redes eléctricas

Códigos en GitHub

Se resuelven distintos ejercicios de circuitos eléctricos, los códigos se encuentran comentados y se muestran figuras de las curvas y circ. esquemáticos.

- Problema de diseño. ^a
- Puente completo celda de carga. ^b
- Amplificador de Puente ^c
- Ec Dif. Lineales ^d
- Laplace ^e

^a<https://github.com/WayraLHD/SRA21/blob/main/python/0/ProblemadeDiseño.ipynb>

^b<https://github.com/WayraLHD/SRA21/blob/main/python/0/Puente.ipynb>

^chttps://github.com/WayraLHD/SRA21/blob/main/python/0/Amp_Puente.ipynb

^d<https://github.com/WayraLHD/SRA21/blob/main/python/0/EDO.ipynb>

^e<https://github.com/WayraLHD/SRA21/blob/main/python/0/Laplace.ipynb>

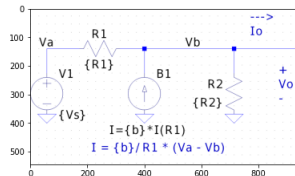
Problema de diseño

Captura de pantalla GitHub

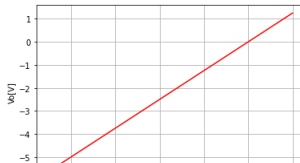
```
In [2]: # Problema de diseño: Circuitos Eléctricos Dorf-Svoboda P5.4
# El circuito que se muestra en la figura tiene cuatro parámetros de circuitos
# no especificados: Vs R1 R2 d
# Para diseñar este circuito
# debemos especificar los valores de estos cuatro parámetros.
# la gráfica que se muestra describe una relación entre Io y Vo

esq= mpimg.imread('Lineal.png') #sube la imagen del circuito esq.
plt.imshow(esq)
plt.show()

Image(filename='Recta.png') #Muestra la imagen
```



Out[2]:

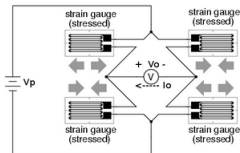


Puente completo celda de carga

Captura de pantalla GitHub

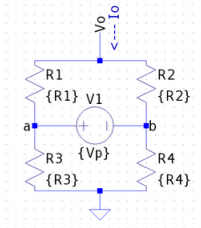
```
In [2]: # Puente completo celda de carga es como se observa en la imagen.
Image(filename='straingauge.png',width=300)
```

Out[2]: Full-bridge strain gauge circuit



```
In [3]: Image(filename='esq_sg.png',width=250) #Circuito esquemático cambiando el enfoque
```

Out[3]:



```
In [10]: #Se busca el equivalente thevenin
#Se van a plantear ecuaciones nodales
#Vth es el voltaje en Vo cuando Io=0 => CA
```

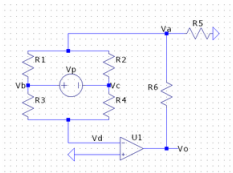
Amplificador de Puente

Captura de pantalla GitHub

In [2]: *# El amplificador puente es como se observa en la imagen.*

```
import numpy as np
import sympy as sym
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from IPython.display import Image
sym.init_printing()
##matplotlib widget
%matplotlib inline
Image(filename='amp_puente.png',width=300)
```

Out[2]:



In [11]: *# VERIFICAR QUE Vo RESPONDE A LA SIGUIENTE ECUACION*

```
sym.var('Va, Vb, Vc, Vd, Vo, Vp')
sym.var('R1, R2, R3, R4, R5, R6')
sym.var('Vo_')
display(sym.Eq(Vo_, sym.fu(((1+R6/R5)*(R2/(R1+R2)-R4/(R3+R4))*Vp))))
Vo_=sym.fu(((1+R6/R5)*(R2/(R1+R2)-R4/(R3+R4))*Vp))
```

$$V_o = V_p \left(1 + \frac{R_6}{R_5} \right) \left(\frac{R_2}{R_1 + R_2} - \frac{R_4}{R_3 + R_4} \right)$$

In [12]: *find=sym.Matrix([[Va],[Vb],[Vc],[Vd],[Vo]]) #Incognitas*

```
#Se escriben tantas ecuaciones como nodos haya
ec_nodo_0=sym.Eq(Vd,0)
ec_nodo_1=sym.Eq(Vb-Vc,Vp)
ec_nodo_2=sym.Eq((Vb-Vd)/R3+(Vc-Vd)/R4,0)
```