

Datos Geoespaciales - 513376

R. Cifuentes Lobos ¹ N. Soto Delis ¹ Franco Retamal ¹

¹Departamento de Geofísica
Universidad de Concepción
Concepción, Chile

Introducción a Python.

Temario.

Introducción.

Lenguaje Python

- Lenguaje de programación de código abierto.
- Centrado en la legibilidad y transparencia.
- Fácil de aprender.
- Funciona a base de módulos.
- Para hacer mapas existen los módulos Basemap y Cartopy.

Introducción.

Lenguaje Python

- Lenguaje de programación de código abierto.
- Centrado en la legibilidad y transparencia.
- Fácil de aprender.
- Funciona a base de módulos.
- Para hacer mapas existen los módulos Basemap y Cartopy.

Introducción.

Lenguaje Python

- Lenguaje de programación de código abierto.
- Centrado en la legibilidad y transparencia.
- Fácil de aprender.
- Funciona a base de módulos.
- Para hacer mapas existen los módulos Basemap y Cartopy.

Introducción.

Lenguaje Python

- Lenguaje de programación de código abierto.
- Centrado en la legibilidad y transparencia.
- Fácil de aprender.
- Funciona a base de módulos.
- Para hacer mapas existen los módulos Basemap y Cartopy.

Introducción.

Lenguaje Python

- Lenguaje de programación de código abierto.
- Centrado en la legibilidad y transparencia.
- Fácil de aprender.
- Funciona a base de módulos.
- Para hacer mapas existen los módulos Basemap y Cartopy.

Instalación.

Prerrequisitos.

```
sudo apt-get install libreadline-dev
sudo apt-get install libsqlite3-dev
sudo apt-get install libbz2-dev
sudo apt-get install libssl-dev
```

Instalación.

Descarga.

```
mkdir /ruta/a/directorio
cd /ruta/a/directorio
sudo wget https://www.python.org/ftp/python/2.7.14/Python-2.7.14.tgz
sudo tar -xzf Python-2.7.14.tgz
cd Python-2.7.14
sudo ./configure --enable-optimizations
sudo make altinstall
```

Instalación.

Instalar IPython y basemap

```
pip install ipython  
sudo apt-get install python-matplotlib  
sudo apt-get install python-mpltoolkits.basemap
```

Generalidades.

Aspectos generales.

- A diferencia de Matlab, no todo es una matriz.
- Se parte contando desde el 0.
- Existen Listas, Tuplas y Diccionarios entre otros.
- Se debe cargar los módulos al principio del script para ser usados posteriormente.
- Los loops y sentencias requieren de indentación.

Generalidades.

Aspectos generales.

- A diferencia de Matlab, no todo es una matriz.
- Se parte contando desde el 0.
- Existen Listas, Tuplas y Diccionarios entre otros.
- Se debe cargar los módulos al principio del script para ser usados posteriormente.
- Los loops y sentencias requieren de indentación.

Generalidades.

Aspectos generales.

- A diferencia de Matlab, no todo es una matriz.
- Se parte contando desde el 0.
- Existen Listas, Tuplas y Diccionarios entre otros.
- Se debe cargar los módulos al principio del script para ser usados posteriormente.
- Los loops y sentencias requieren de indentación.

Generalidades.

Aspectos generales.

- A diferencia de Matlab, no todo es una matriz.
- Se parte contando desde el 0.
- Existen Listas, Tuplas y Diccionarios entre otros.
- Se debe cargar los módulos al principio del script para ser usados posteriormente.
- Los loops y sentencias requieren de indentación.

Generalidades.

Aspectos generales.

- A diferencia de Matlab, no todo es una matriz.
- Se parte contando desde el 0.
- Existen Listas, Tuplas y Diccionarios entre otros.
- Se debe cargar los módulos al principio del script para ser usados posteriormente.
- Los loops y sentencias requieren de indentación.

NumPy.

Módulo NumPy.

- Módulo con funciones similares a Matlab.
- `import numpy as np.`
- Para usar una función de numpy es necesario especificar que esta corresponde al módulo :
`a=np.sqrt(2)`

NumPy.

Módulo NumPy.

- Módulo con funciones similares a Matlab.
- `import numpy as np.`
- Para usar una función de numpy es necesario especificar que esta corresponde al módulo :
`a=np.sqrt(2)`

NumPy.

Módulo NumPy.

- Módulo con funciones similares a Matlab.
- `import numpy as np.`
- Para usar una función de numpy es necesario especificar que esta corresponde al módulo :
`a=np.sqrt(2)`

Matplotlib.

Módulo Matplotlib.

- Módulo con funciones para crear gráficos.
- `import matplotlib.pyplot as plt.`

Matplotlib.

Módulo Matplotlib.

- Módulo con funciones para crear gráficos.
- `import matplotlib.pyplot as plt.`

Ejemplos.

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

```
def f(t):
    'Exponencial amortiguado'
    s1 = np.cos(2 * np.pi * t)
    e1 = np.exp(-t)
    return s1 * e1
```

```
t1 = np.arange(0.0, 5.0, .2)
l = plt.plot(t1, f(t1), 'ro')
plt.show()
```

Ejemplos.

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

t = np.arange(0.0, 1.0 + 0.01, 0.01)
s = np.cos(2 * 2*np.pi * t)
t[41:60] = np.nan

plt.subplot(2, 1, 1)
plt.plot(t, s, '--', lw=2)

plt.xlabel('time (s)')
plt.ylabel('voltaje (mV)')
plt.title('Seno con NaNs entre 0.4 y 0.6')
plt.grid(True)

plt.subplot(2, 1, 2)
t[0] = np.nan
t[-1] = np.nan
plt.plot(t, s, '--', lw=2)
plt.title('Seno con NaNs en el primer y final elemento.')

plt.xlabel('tiempo (s)')
plt.ylabel('Voltaje')
plt.grid(True)

plt.tight_layout()
plt.show()
```

Ejercicios.

- 1 Agregue una línea azul al gráfico del primer ejemplo.
- 2 Agregue una grilla, etiquetas en ambos ejes y un título al primer gráfico.
- 3 Utilice la función `plt.legend()` para agregar una leyenda.
- 4 Grafique una función cuadrática con líneas punteadas que contenga grilla, etiqueta y leyendas.
- 5 Cree un gráfico con dos subplots donde el primero sea una función exponencial con los últimos 5 elementos NaN y el segundo una función exponencial amortiguada con marcadores en forma de estrellas rojas.
- 6 Descargue estos datos y gráfíquelos. Calcule sus estadísticos y utilizando la función `plt.text()` anote los resultados en el gráfico.