Doctorado en Tecnologías de Transformación Digital

Materia: Ingeniería para el procesamiento masivo de datos

Hoja de Trucos de Python

Alumno: Luis Alejandro Santana Valadez

Tutor: Dr. Jonás Velasco Álvarez

Pachuca de Soto, Hgo.
29 de junio de 2025

1. Introducción

Esta hoja de trucos de Python está diseñada como una referencia práctica para estudiantes e ingenieros que trabajan con procesamiento masivo de datos. Resume los comandos, técnicas y estructuras más importantes del lenguaje Python, con ejemplos concretos y organizados por secciones temáticas. La información contenida fue extraída y estructurada a partir de las prácticas y tablas analizadas en los capítulos revisados de los materiales de estudio del Doctorado en Tecnologías de Transformación Digital. Cada sección ofrece ejemplos funcionales y una breve descripción para cada instrucción.

2. Entornos y Ejecución

python — Ejecuta el intérprete interactivo de Python.
exit() — Finaliza la sesión interactiva de Python.
python -m venv [nombre] — Crea un entorno virtual para el proyecto.
[entorno]\Scripts\activate — Activa el entorno en Windows.
deactivate — Desactiva el entorno activo.
pip install -r archivo.txt — Instala dependencias desde un archivo.

```
python3
exit()
python -m venv mi_entorno
mi_entorno\Scripts\activate
pip install -r requirements.txt
deactivate
```

3. Tipos de Datos y Estructuras

int, float, complex — Tipos numéricos: enteros, decimales y complejos.
Decimal(str) — Precisión decimal exacta.
set(), frozenset() — Conjuntos mutables e inmutables.
dict() — Diccionario clave-valor.
defaultdict(tipo) — Diccionario con valores por defecto.

Números y Cadenas

```
a = 42
b = 3.14
c = complex(2, 3)

from decimal import Decimal
Decimal('0.1') + Decimal('0.2')

texto = "Python"
print(texto[0])      # Indexado
print(texto[::2])    # Rebanado
```

Listas

```
lista = [1, 2, 3]
lista.append(4)
lista.sort()

tupla = (10, 20)
a, b = b, a

s = set([1, 2, 2, 3])
fs = frozenset([4, 5, 6])
```

Diccionarios y Colecciones (ancho completo)

```
from collections import defaultdict, namedtuple, ChainMap

# defaultdict
dd = defaultdict(int)
dd['x'] += 1

# namedtuple
Vision = namedtuple('Vision', ['left', 'right'])
v = Vision(left=9.5, right=8.8)
print(v.left)

# ChainMap
dict1 = {'a': 1}
dict2 = {'b': 2}
combined = ChainMap(dict1, dict2)
print(combined['b'])
```

4. Control de Flujo

if/elif/else — Condiciones.
for, while — Bucles de iteración.
break, continue — Control de flujo en ciclos.
enumerate(), zip() — Iteraciones avanzadas.

```
if x > 0:
    print("Positivo")
elif x == 0:
    print("Cero")
else:
    print("Negativo")

for i in range(5):
    print(i)

for i, c in enumerate("abc"):
    print(i, c)

for a, b in zip([1, 2], ['uno', 'dos']):
    print(a, b)
```

5. Funciones y Parámetros

def — Define una función.
***args, ****kwargs** — Argumentos variables.
help() — Muestra la documentación.

```
def saludar(nombre="amigo"):
    print("Hola", nombre)

def mostrar(*args, **kwargs):
    for a in args:
        print(a)
    for k, v in kwargs.items():
        print(k, v)

valores = (1, 2)
param = {'a': 10, 'b': 20}
mostrar(*valores, **param)
help(print)
```

6. Archivos y Persistencia

open(), **with open()** — Lectura y escritura.
pickle, shelve — Serialización de objetos.
json.dumps(), json.loads() — JSON.

Lectura y escritura

```
with open('archivo.txt', 'r') as f:
    print(f.read())

with open('nuevo.txt', 'w') as f:
    f.write("Hola mundo")
```

Persistencia con pickle y shelve

```
import pickle, shelve

data = [1, 2, 3]
with open('data.pkl', 'wb') as f:
    pickle.dump(data, f)

with shelve.open('basedatos') as db:
    db['clave'] = data
```

7. JSON, Solicitudes y Archivos ZIP

requests — Peticiones HTTP.
ZipFile — Manejo de archivos comprimidos.

```
import json, requests
from zipfile import ZipFile

obj = {'clave': 'valor'}
json_str = json.dumps(obj, indent=2)

response = requests.get("https://httpbin.org/ip")
print(response.json())

with ZipFile('backup.zip', 'w') as z:
    z.write('archivo.txt')
```