



Manipulación y transformación de datos (Parte III)

Clase sincrónica

Manipula datos utilizando funciones y herramientas de Python y SQL.

- Unidad 1: Conociendo Python
(Parte I)
(Parte II)
- Unidad 2: Tipos y estructura de datos
(Parte I)
(Parte II)
- Unidad 3: Manipulación y transformación de datos
(Parte I)
(Parte II)
(Parte III)



Te encuentras aquí



¿Qué aprenderás en esta sesión?

- A crear funciones en Python, y conectar con bases de datos de SQL

¿Qué es una función, en matemáticas?



¡Manos a la obra!
A crear funciones



A crear funciones

Veremos cómo crear funciones utilizando Jupyter Notebook para que tu profesor pueda mostrarte las principales herramientas. A continuación, aprenderemos:

1. Funciones - Definición básica
 - a. Componentes de una función
 - b. Parámetros y argumentos
 - c. Retornos
 - d. Principio DRY y buenas prácticas
2. Funciones, argumentos y variables
 - a. Parámetros obligatorios y optativos
 - b. Funciones como variables
3. Funciones y DataFrames

Abre tu propio archivo de Jupyter y replica los comandos que irás aprendiendo.



****/Python y SQL/****

Python y SQL

Conectando bases de datos

Python nos permite conectarnos con bases de datos SQL gracias a las bibliotecas `psycopg2` y `sqlalchemy`. Para poder trabajar combinando estas funcionalidades, deberás abrir una terminal en tu equipo y aplicar los siguientes pasos:

- **Instalar las bibliotecas.** Para ello, escribe
 - `pip install psycopg2-binary`
 - `pip install sqlalchemy`
- **Crear la base de datos:**
 - `psql -h localhost -p 5432 -U postgres -c "CREATE DATABASE northwind"`
- **Cargar el archivo a la base de datos:**
 - `psql -h localhost -p 5432 -U postgres -d northwind -f northwinddb.sql`

Python y SQL

Conectando bases de datos

Para trabajar con Python y SQL, se recomienda escribir algunos comandos que permitirán hacer más limpio nuestro trabajo.

- **Creamos un string de conexión.** Se recomienda crear uno diferente para cada base de datos a la que queramos conectar. En nuestro caso escribiremos

```
db_url = "postgresql://postgres:password@localhost:5432/northwind"
```

donde **password** es la clave para el usuario postgres (se debe cambiar para cada usuario)

- **Creamos un objeto de conexión.** De la misma manera, crearemos diferentes “engines” para cada conexión que deseemos realizar,

```
from sqlalchemy import create_engine  
engine = create_engine(db_url)
```

¡Manos a la obra!

Python y SQL



Python y SQL

Veremos ahora cómo conectar y consultar con una base de datos SQL utilizando funciones de Python. Para ello, además, de haber ejecutado los pasos anteriores, deberás descargar el archivo 01 - Python y SQL y ejecutarlo en tu equipo utilizando Jupyter Notebook.

¡A continuación, pondremos en práctica lo aprendido! ¡Sigue las instrucciones de tu profesor y replícalas en tu propio archivo de Jupyter!



Prueba - Python para el análisis de datos



Prueba

"Python para el análisis de datos"

- Descarga el archivo "Prueba"
- Tiempo de desarrollo asincrónico: desde 2 horas.
- Tipo de desafío: individual.

¡AHORA TE TOCA A TI! 💪



Ideas fuerza



Podemos definir **funciones** en Python, que a partir de **parámetros y variables** permiten **retornar** resultados y automatizar tareas



Podemos aplicar funciones a **estructuras de datos**, como listas, funciones o **DataFrames**



Es posible acceder a **bases de datos** con Python, realizando **consultas en SQL** y almacenando resultados en **DataFrames**

¿De qué manera crees que
Python mejora tus aptitudes
como Analista de Datos?



Recursos asincrónicos

¡No olvides revisarlos!

Para esta semana deberás revisar:

- Guía de estudio



{desafío}
latam_

*Academia de
talentos digitales*

