

Woche 8: Datenquellen mit Python lesen

Einleitung

Die folgenden drei Wochen widmen sich einem Spezialthema, bei dem Python effektiv eingesetzt werden kann, der Datenanalyse. Mit Python können Sie sich Datenquellen erschließen, die Sie mit anderen Modulen aufbereiten und präsentieren können. Eine weitere Verwendung von Python im Bereich der Datenanalyse, sind die Verfahren des maschinellen Lernens. Dies würden allerdings den Kursrahmen sprengen und sind daher nicht mehr Teil dieser Weiterbildung.

Die achte Woche der Schulung zeigt unter anderem am Beispiel von verschiedenen Textformaten und der SQL-Datenbanken, wie Sie auf Datenquellen zugreifen können. Sie beginnen also mit der Erkundung verschiedener Datenstrukturen und wie man CSV-Dateien in Python manipuliert. Danach werden die Konzepte der Nutzung relationaler Datenbanken mit SQLite eingeführt und Sie lernen, wie Sie SQL-Syntax in Python anwenden können, um CRUD-Operationen (Erstellen, Lesen, Aktualisieren und Löschen) durchzuführen.

Diese Woche wird Ihre Datenverarbeitungsfähigkeiten erweitern und Ihnen dabei helfen, effizienter mit großen Datenmengen zu arbeiten.

Gesamtüberblick

Hier ein Überblick über die Inhalte und Aktivitäten der aktuellen Woche:

- Selbststudium:
 - Datenstrukturen
 - CSV-Dateien
 - Relationale Datenbanken (SQLite)
 - SQL-Syntax in Python
 - CRUD-Operationen
 - NoSQL Datenbanken (MongoDB)
- Aufgabe:
 - Datenbankoperationen durchführen für relationale Datenbanken
 - Daten transformieren

Inhalte und thematische Abgrenzung

Die folgende Auflistung zeigt detailliert, welche Themen Sie in der Woche behandeln und bearbeiten. Sie sind eine Voraussetzung für die folgenden Wochen und sollten gut verstanden worden sein. Wenn es Verständnisprobleme gibt, machen Sie sich Notizen und fragen Sie am Präsenztage nach, so dass wir gemeinsam zu Lösungen kommen können. Und denken Sie bitte immer daran: es gibt keine „dummen“ Fragen!

1. Datenstrukturen:

- Einführung in Datenstrukturen in Python
- Verschiedene Arten von Datenstrukturen: Listen, Tupel, Mengen, Dictionaries
- Wahl der richtigen Datenstruktur für verschiedene Szenarien

2. CSV-Dateien:

- Verständnis der Struktur von CSV-Dateien
- Verwendung der csv-Bibliothek in Python zum Lesen und Schreiben von CSV-Dateien
- Datentransformation und -manipulation in CSV-Dateien

3. Relationale Datenbanken (SQLite):

- Grundkonzepte relationaler Datenbanken
- Einführung in SQLite: Vorteile und Eigenschaften
- Erstellung und Manipulation von SQLite-Datenbanken in Python

4. SQL-Syntax in Python:

- Grundlagen der SQL-Syntax: SELECT, INSERT, UPDATE, DELETE
- Anwendung der SQL-Syntax in Python zur Interaktion mit Datenbanken
- Verwendung von parametrisierten Abfragen zur Erhöhung der Sicherheit

5. CRUD-Operationen:

- Verständnis der CRUD-Operationen (Create, Read, Update, Delete)
- Implementierung von CRUD-Operationen in Python mit SQLite
- Best Practices für die Durchführung von CRUD-Operationen

Lernpfad

Der Lernpfad ist ein Vorschlag, in welcher Reihenfolge Sie die Inhalte der Woche angehen können. Betrachten Sie ihn gerne als eine Todo-Liste, die Sie von oben nach unten abhaken. So können Sie sicher sein, dass Sie alle wichtigen Themen bearbeitet haben und sind gut vorbereitet für die folgenden Wochen.

1. Beginnen Sie die Woche mit einem vertiefenden Blick auf Python Datenstrukturen.

- Untersuchen Sie spezifische Anwendungsfälle für Listen, Tupel, Mengen und Dictionaries.
- Erforschen Sie die Methoden, die mit jeder Struktur verbunden sind, und lernen Sie, diese in Ihrem Code effektiv einzusetzen.

2. Als nächstes konzentrieren Sie sich auf CSV-Dateien und wie sie in Python gehandhabt werden.

- Nutzen Sie die eingebaute csv Bibliothek von Python zum Einlesen und Schreiben von CSV-Dateien.
- Experimentieren Sie mit der Datentransformation und Manipulation innerhalb dieser Dateien und arbeiten Sie an einigen Übungsaufgaben, um Ihre Fähigkeiten zu festigen.

3. Tauchen Sie dann in die Welt der relationalen Datenbanken ein.

- Beginnen Sie mit den Grundlagen der relationalen Datenbanktheorie und verstehen Sie, was Tabellen, Zeilen, Spalten und Schlüssel sind.
- Installieren Sie SQLite und lernen Sie, wie Sie diese Datenbank in Python verwenden können.

4. Als nächstes konzentrieren Sie sich auf SQL in Python.

- Lernen Sie die Grundlagen der SQL Syntax und wie Sie diese in Python verwenden können.
- Üben Sie die Erstellung von Tabellen, das Einfügen von Daten und das Abfragen von Daten mit den Grundlagen von SQL.

5. Vertiefen Sie Ihr Verständnis für CRUD-Operationen in SQL.

- Üben Sie das Erstellen, Lesen, Aktualisieren und Löschen von Daten in einer SQLite Datenbank.
- Stellen Sie sicher, dass Sie sich mit den Befehlen und Konzepten, die mit jeder dieser Operationen verbunden sind, wohl fühlen.

6. Wenden Sie sich dann NoSQL-Datenbanken zu und lernen Sie die Unterschiede zu relationalen Datenbanken kennen.

Programmieraufgaben

Die folgenden Programmieraufgaben sollen Ihnen eine Anregung geben. Haben Sie eigene Ideen und Themen, die Sie ausprobieren wollen, dann sollten Sie diesen nachgehen. Wichtig ist vor allem, dass Sie „Dinge ausprobieren“. Und auch, dass Sie Fehler machen, sowohl syntaktische als auch semantische. Versuchen Sie diese Fehler zu finden und aufzulösen, dann gerade aus den Fehlern lernen Sie am Ende am meisten.

1. Datenstrukturen:

Entwerfen Sie eine Aufgabe, in der eine Mischung aus Datenstrukturen erforderlich ist. Zum Beispiel, erstellen Sie ein Telefonbuch, das Namen (String) und zugehörige Telefonnummern (Liste von Strings) speichert. Implementieren Sie Funktionen, um Einträge hinzuzufügen, zu entfernen und nach bestimmten Namen zu suchen. Denken Sie an eine Speichern- und Ladenfunktionalität.

2. CSV-Dateien:

Geben Sie eine CSV-Datei mit Daten aus. Die Aufgabe ist es, eine Python-Anwendung zu schreiben, die die CSV-Datei einliest, einige Berechnungen oder Transformationen auf den Daten durchführt (z.B. Durchschnitt einer Spalte berechnen, Daten filtern, etc.) und die Ergebnisse in einer neuen CSV-Datei speichert.

3. Relationale Datenbanken (SQLite):

Entwickeln Sie eine einfache Anwendung, die eine SQLite-Datenbank verwendet, um Daten zu speichern und abzurufen. Beispielsweise könnte die Anwendung ein einfaches Inventarsystem für ein Geschäft sein, das Produkte und ihre Mengen speichert.

4. SQL-Syntax in Python:

Schreiben Sie ein Python-Skript, das eine bestehende Datenbank verwendet und komplexe SQL-Abfragen ausführt, wie z.B. JOIN-Operationen oder gruppierte Abfragen mit HAVING. Die Abfragen sollten dann entsprechend formatiert und ausgegeben werden.

Ressourcen

Hier nun die Verweise auf Lernquellen, die uns für diese Woche und ihre Inhalte geeignet erscheinen. Je nachdem, welcher Lerntyp Sie sind, wählen Sie sich ihre bevorzugte Quelle, es ist nicht zwingend notwendig alle durchgearbeitet zu haben. Allerdings sollten die Inhalte des Lernpfads angesprochen und verstanden worden sein.

- **Buch:**

Foundation Db2 and Python

- **Video:**

Codecademy: Querying Sqlite Databases With Python Cheatsheet

Codecademy: Learn Advanced Python 3: Database Operations

SQL Databases Using SQLAlchemy: Getting Started

Flask-RESTful: Integrating the API Server with a MySQL Database

SQL Databases Using SQLAlchemy: Using the SQLAlchemy