**Debugging: Fehlerbehebung in Programmen:**

Debugging ist der Prozess, Fehler aus einem Programm zu entfernen. Dabei versucht man, mögliche Fehlerquellen zu identifizieren, diese Fehler anzeigen zu lassen und anschließend zu beheben. Hierfür haben wir unter anderem die klassische print()-Funktion verwendet.

Allerdings stößt die alleinige Nutzung von print() an ihre Grenzen, insbesondere wenn das Programm nach einem Fehler abstürzt.

**Erweiterte Fehlerbehandlung mit try: und excpet:**

Um Abstürze zu vermeiden, haben wir häufig die Konstrukte **try:** und **except Exception as e:** eingesetzt. Mit diesen Funktionen kann ein Programmabschnitt (im try-Block) „versucht“ werden. Sollte dabei ein Problem auftreten, wird dieses „abgefangen“ (im except-Block). Im except-Block können wir den Fehler dann mit **print(e)** ausgeben. Der große Vorteil dieser Methode ist, dass das Programm nicht abstürzt. Es gibt lediglich den Fehler aus und überspringt den restlichen Teil der Funktion, sodass das Programm mit anderen Funktionen fortfahren kann.

**Gezieltes Fehlermanagement mit raise Exception(„text“):**

Zusätzlich haben wir das **raise Exception(„text“)** -Statement genutzt. Dies ermöglicht es, Funktionen, bei denen ein bestimmtes, vom Programm nicht unterstütztes Ereignis erwartet wird, abzubrechen und das Problem zu benennen. Dieses Problem kann dann wie oben beschrieben mit try und except erfasst und ausgegeben werden.

**Effektivste Fehleranalyse mit dem Python-Debugger:**

Sollten all diese Methoden nicht ausreichen, können wir auf den **Python-Debugger** zurückgreifen. Die Debugger-Übersicht von Visual Studio Code ist hierbei sehr hilfreich. Sie ermöglicht es uns, neben den Zeilennummern einen **Breakpunkt** zu setzen. An diesem Punkt unterbricht das Programm die Ausführung, und wichtige Daten wie relevante Variablen in der jeweiligen Zeile, können in der Übersicht angezeigt werden.

**Beispiel:**

Das Video demonstriert, **wie der Debugger genutzt werden kann**, um Fehler zu finden, die beispielsweise durch **fehlerhafte Daten** entstehen.

