

In der zweiten Übung in Lektion 2 wirst du lernen, wie man in einem Programm Variable als **lokale** Variable festlegt.

Lernziele:

- Den Nutzen von lokalen Variablen erkennen
- **Lokale** Variable in Programmen einsetzen

Beginne indem du das Dokument öffnest, das das Programm **heron** enthält, das du in Übung 1 geschrieben hast (siehe Abbildung rechts).

Erinnere dich, dass dieses Programm die Variablen **s** und **fl** - unbeabsichtigt - so erzeugt hat, dass sie für das aktuelle Problem im ganzen Dokument Gültigkeit hat. Das ist so nicht erwünscht, und du wirst lernen, wie du das vermeiden kannst.

Füge unter dem **Prgm**-Schlüsselwort die Anweisung

Local s, fl ein, die über **menu> Variable definieren** zu finden ist.

Nachdem das geschehen ist, überprüfe und speichere das Programm wie bereits gewohnt über **menu> Syntax überprüfen und speichern> Syntax überprüfen und speichern** (oder mit **ctrl [B]**).

Die Variablen **s** und **fl** löschst du nun im Calculator mit **DelVar s, fl**.

(**Variable löschen** findet sich in **menu> Aktionen > Variable löschen**.)

Führe das Programm **heron** aus.

Drücke wieder die **var**-Taste und du wirst feststellen können, dass nur mehr das Programm **heron** in der Variablenliste aufscheint.

Was ist geschehen?

Wenn man eine Variable als 'Local' deklariert, sagt man dem TI-Nspire™ CX, dass diese Variable während der Ausführung des Programms erzeugt und nach seiner Beendigung wieder gelöscht wird. Damit gibt es sie im aktuellen Problem nicht. Das löst auch das Problem, wenn die Variable **s** oder **fl** im Problem erzeugt wird und dort im Speicher verbleibt. Wenn dann z.B. **s** im Programm als lokale Variable eingeführt wird, berührt das die 'Problemvariable' **s** nicht, da das Programm seine eigene (temporäre) Variable erzeugt.

Zusammenfassung

Die 'Gültigkeit' einer Variablen ist der Bereich, in dem sie existiert. Im TI-Nspire™ CX gelten die Variablen im jeweils *aktuellen tns-Problem*. Wenn man im Dokument ein neues Problem öffnet, erhält man wieder eine leere Variablenliste. So weiß Problem 2 nichts über die Variablen von Problem 1 und umgekehrt.

Wenn Programme im selben Problem die gleichen Variablen verwenden kann es unbeabsichtigt oder auch beabsichtigt zur Erzeugung oder zum Gebrauch der Variablen in diesem Problem kommen. Bisweilen ist es sinnvoll, Programmvariable und Problemvariable zu verbinden. Darauf ist beim Programmentwurf zu achten.

Variable als **Local** zu definieren vermeidet die Erzeugung und Beeinflussung von Problemvariablen.

Für Funktionen gelten andere Regeln im Umgang mit lokalen Variablen. Das werden wir in der nächsten Übung sehen.
Bleib dran!