Lektion 3 : Programmbeispiele — **Übung 3 : Iteration und Rekursion**

In dieser dritten Lektion von Einheit 3 verwenden Sie Funktionen, um sowohl eine iterative als auch eine rekursive Programmierung durchzuführen.

Lernziele :

- Anwendung einer Funktion
- Kennenlernen und Anwenden der iterativen und der rekursiven Programmierung

---

Berechnung des ggT durch Iteration

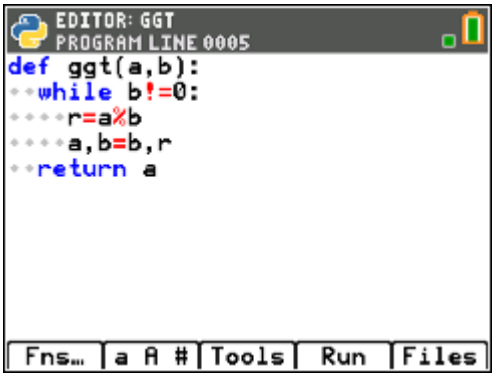
Um den größten gemeinsamen Teiler zweier Zahlen (**ggT**) mit **a > b** zu berechnen, wird der **Euklid-Algorithmus** verwendet, der schnell zum Ergebnis führt:

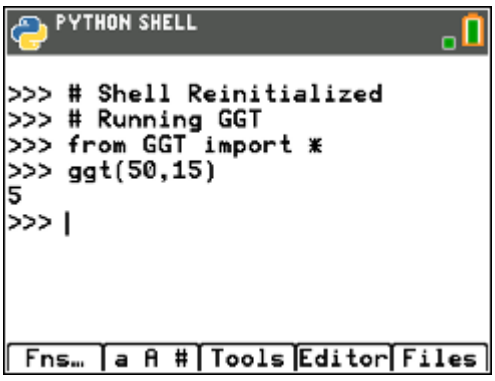
- Bei der Division a/b verbleibt ein Rest r < b.
- Man setzt nun a = b und b = r .
- Die Division a/b wird nun wiederholt.
- Der Prozess endet, wenn r = 0 ist.
- Der ggT ist dann der vorherige Rest.
- Beim oben abgebildeten Beispiel ist **ggT(96,81) = 3**.

a		*b*		*r*
96	= 1 *	81	+	15
81	= 5 *	15	+	6
15	= 2 *	6	+	3
6	= 2 *	3	+	0

Das Programm

- Erstellen Sie ein neues Skript „**GGT**“.
- Definieren Sie die Funktion **ggt(a,b)**.
- „**!=**“ steht für „**≠**“ und „**%**“ für die Bestimmung des **Restes** bei der euklidischen Division.
- **a,b=b,r** ersetzt die Anweisungen **a=b** und **b=r**.
- Testen Sie das Skript mit verschiedenen Beispielen.





Rekursive Programmierung

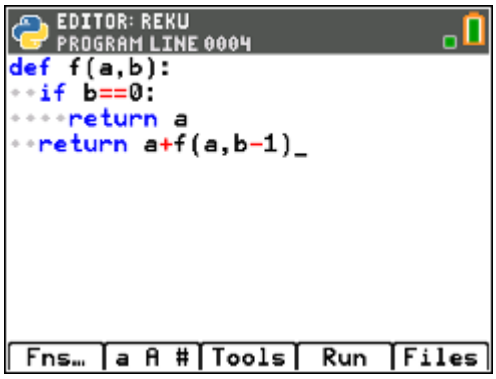
Ein Algorithmus wird als **rekursiv** bezeichnet, wenn er sich irgendwann **selbst aufruft**.

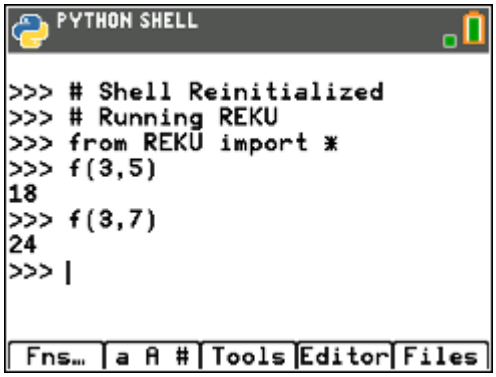
Rekursion kann in einem Algorithmus viele Vorteile haben. Erstens löst sie Probleme, die durch die Verwendung einfacher Schleifen u.U. unlösbar sind. Sie kann einen Algorithmus aber auch lesbarer und kürzer machen, ermöglicht aber vor allem in bestimmten Fällen einen kolossalen Zeitgewinn, wie dies bei den Sortieralgorithmen der Fall ist.

© 2020 Texas Instruments

Ein erstes Beispiel für eine Rekursion

- Erstellen Sie ein neues Skript „**REKU**“.
- Machen Sie sich klar, an welcher Stelle **f(a,b)** von sich selbst aufgerufen wird!
- Überlegen Sie die Rolle der **if** – Anweisung und was passieren würde ohne diese Anweisung!
- Wie entsteht das Ergebnis ?





Rekursive Bestimmung des ggT

- Die beiden Bilder zeigen das rekursive Skript für den ggT sowie eine Berechnung.
- Hier wie bei jeder Rekursion ganz wichtig : es muss eine Abbruchbedingung angegeben werden, die auch sicher im Programmablauf erreicht wird, da sonst das Programm ewig läuft (d.h. bis die Kapazität des Rechners erschöpft ist).