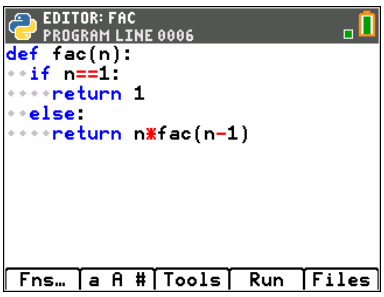
Unit 3 : Beginselen van programmeren in Python | **Oefenblad 1 : Recursieve functie**

In deze Unit gaan we wat dieper in op programmeren in Python.

Doelen :

- Een recursieve functie definiëren.
- Recursief versus rechtstreeks.

Een recursieve functie is een functie die zichzelf aanroept.
Een mooi voorbeeld hiervan is de faculteit functie.
Onder 5! (spreek uit 5 faculteit) verstaan we: $5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 120$.
Algemeen: $n! = n \cdot (n-1) \cdot (n-2) \cdot \cdots \cdot 1$
Dit kun je ook schrijven als: $n! = n \cdot (n-1)!$
Dus als je de functie fac(n) maakt voor n!, dan geldt: fac(n)=n*fac(n-1).



De functie krijgt de vorm:
Als n=1 dan is de uitkomst 1
Anders is de uitkomst n*fac(n-1)

Maak een Python programma waarin je deze functie definieert en bereken daarmee een aantal uitkomsten.

Een nadeel van deze functie is dat voor grote waarden van n het geheugen van de rekenmachine niet toereikend is. Zoek uit wat de grootste waarde van n is zodat de functie nog een uitkomst geeft

Een andere benadering is een functie die gebruik maakt van een for-lus:
Start met het getal 1 en maak een for-lus waarbij elke keer de nieuwe uitkomst wordt vermenigvuldigd met de lus-variabele (i in dit geval).

Definieer deze functie als fac1(n) en ga na dat je hiermee bijvoorbeeld wel 70! kunt berekenen terwijl dat met je gewone rekenmachine functie niet kan.