

## Deel 1 – ascii-figuren

Bekijk eerst de video's functies 1 t.e.m. functies 7 op [www.youtube.be/dve845](http://www.youtube.be/dve845)

Voor deze opgave zijn er, zoals gebruikelijk, twee bestanden. Het bestand **opgave6.py** en het bestand **opgave6tester.py**. Maak een nieuw project en zet beide bestanden in de projectfolder. Hernoem het bestand **opgave6.py** naar **opgave6opl.py**.

Het laatste bestand bevat de reeds gedefinieerde automatische testen voor de functies die moeten geschreven worden in het eerste bestand. Het is de bedoeling dat alle testen slagen. Probeer niet alle testen in één keer aan te pakken. Werk ze test per test (en dus functie per functie) af.

Zelf testen toevoegen mag, maar de bestaande laat je ongemoeid. Bekijk alle testen eens in vogelvlucht om een idee te hebben van welke functionaliteit gevraagd wordt, want het is best mogelijk dat sommige functies nuttig kunnen zijn voor het definiëren van andere. De volgorde van de testen volgt dus niet noodzakelijk de beste volgorde voor implementatie. Verderop volgt een kleine omschrijving van de gevraagde functies. Voor deze opgave heb je geen andere bestaande functies nodig!

**line(length, fill = True, indentation = 0, char = '\*')**

geeft een lijn terug van *length* keer het karakter *char*, al dan niet opgevuld en voorafgegaan door *indentation* spaties

**square(size, fill = True, indentation = 0, char = '\*')**

geeft vierkant terug met zijde *size* als string (andere parameters gedragen zich zoals hierboven)

**rectangle(width, height, fill = True, indentation = 0, char = '\*')**

geeft een rechthoek terug als string van *height* rijen met telkens *width* karakters (andere parameters gedragen zich zoals hierboven)

**parallelogram(width, height, fill = True, indentation = 0, char = '\*', step = -1)**

geeft een parallelogram terug als string van *height* rijen met telkens *width* karakters die wijzigen met *step* per lijn (andere parameters gedragen zich zoals hierboven)

## Deel 2 – voor gevorderden of zij die een uitdaging willen

Sla deze oefening over als nieuw bent met functies. Maak ze in dat geval later (laatste week of zo).

**triangle(width, fill = True, indentation = 0, char = '\*', step = 0, alignRight = False, fromTopToBottom = True, rightAngled = True)**

geeft een driehoek terug met basis *width* (gelijknamige parameters gedragen zich zoals hierboven)

*alignRight* bepaalt of de driehoek al dan niet rechts uitgelijnd is

*fromTopToBottom* bepaalt of de driehoek z'n top al dan niet bovenaan heeft

*rightAngled* bepaalt of het een rechthoekige driehoek is of een gelijkbenige (in het laatste geval wordt *alignRight* genegeerd)