

Deel 1 – meer stringfuncties

Bekijk eerst de video's functies 1 t.e.m. functies 7 op www.youtube.be/dve845

Voor deze opgave zijn er opnieuw twee bestanden. Het bestand **opgave3.py** en het bestand **opgave3tester.py**. Maak een nieuw project en zet beide bestanden in de projectfolder. Hernoem het bestand **opgave3.py** naar **opgave3opl.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 mag gebruik gemaakt worden van volgende zaken

- de functie **len()**
len(x) geeft de lengte van de string of de array *x* terug
- de functie **append()**
x.append(y) de waarde *y* wordt toegevoegd aan de tabel *x*, waardoor *x* groter wordt
let op de puntnotatie, het wordt volgend semester duidelijk waar dit vandaan komt
- de **[]** operator op strings
x[i] geeft het *i*-de karakter terug van de string *x*

Gebruik van andere functies is niet nodig!

decompose(x)

splitst een getal *x* in *z*'n cijfers en geeft die terug in een array (voor details zie testen)

reverse(s)

geeft de string *s* omgekeerd (achterstevoren) terug

isPalindrome(s)

geeft terug of een string *s* identiek is aan *z*'n omgekeerde, d.m.v. een logische waarde (bool)

find(needle, haystack)

geeft de plaats terug waar de string *needle* kan gevonden worden in de string *haystack* (voor details zie testen)

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).

findAll(needle, haystack)

geeft alle plaatsen (in de vorm van een array) terug waar de string *needle* kan gevonden worden in de string *haystack* (voor details zie testen)

replace(string, byString, inString)

vervangt alle *string* voorkomens door de string *byString* in de string *inString* zonder dat deze laatste wijzigt (voor details zie testen)

Schrijf de functie **inDutch(anInteger)** die het gegeven geheel getal *anInteger* omzet naar een string waarin dat getal voluit wordt geschreven in het Nederlands. Het kan helpen om als tussenstap eerst een functie te maken die dat doet voor getallen van maximaal 3 cijfers. De functie die dat doet heb ik in de testen de werktitel **inDutchThreeDigit(threeDigitInteger)** genoemd.