

# Python Programming - Labo 5

## Doel van dit labo

Deze oefeningen zorgen ervoor dat je van start kan gaan met het ontwikkelen in Python en maken je vertrouwd met de basisbeginselen. In dit labo nemen we de kennis uit de voorgaande labo's mee en introduceren we werken met bestanden, command-line argumenten en foutafhandeling.

OPMERKING: bij bepaalde oefeningen voorzie je zelf een tekstbestand, voorzien van data waarmee je je oefening tot een goed einde kan brengen. In bepaalde oefeningen, 10 tot en met 14, is een dataset voorzien!

## Oefening 1

Op Unix gebaseerde besturingssystemen bevatten een tool met de naam head. Het toont de eerste 10 regels van een bestand waarvan de naam is opgegeven als een command-line argument. Schrijf een Python-programma met hetzelfde gedrag. Geef een passend foutbericht weer als het door de gebruiker gevraagde bestand niet bestaat, of als het command-line argument wordt weggelaten.

## Oefening 2

Op Unix gebaseerde besturingssystemen bevatten doorgaans ook een tool met de naam tail. Het toont de laatste 10 regels van een bestand waarvan de naam is opgegeven als een command-line argument. Schrijf een Python-programma met hetzelfde gedrag. Geef een passend foutbericht weer als het door de gebruiker gevraagde bestand niet bestaat of als het command-line argument wordt weggelaten. Er zijn verschillende benaderingen die kunnen worden gevolgd om dit probleem op te lossen.

Een optie is om de volledige inhoud van het bestand in een lijst te laden en vervolgens de laatste 10 elementen weer te geven. Een andere optie is om de inhoud van het bestand twee keer te lezen, een keer om de regels te tellen en een tweede keer om de laatste 10 regels weer te geven. Beide oplossingen zijn echter ongewenst bij het werken met grote bestanden. Er bestaat een andere oplossing

waarbij u het bestand slechts één keer hoeft te lezen en slechts 10 regels uit het bestand tegelijk hoeft op te slaan. Voor een extra uitdaging, ontwikkel een dergelijke oplossing.

## Oefening 3

Op Unix gebaseerde besturingssystemen bevatten doorgaans ook een tool genaamd cat, een afkorting voor concatenate of samenvoegen. Het doel is om de aaneenschakeling van een of meer bestanden weer te geven waarvan de namen worden meegegeven als command-line argumenten. De bestanden worden weergegeven in dezelfde volgorde als waarin ze zijn meegegeven. Maak een Python-programma dat deze taak uitvoert. Je programma genereert ook een passende foutmelding voor elk bestand dat niet kan worden weergegeven en gaat vervolgens verder naar het volgende bestand. Geef een passend foutbericht weer als het programma wordt gestart zonder het command-line argument(en).

## Oefening 4

Maak een programma dat regels uit een bestand leest, er regelnummers aan toevoegt en de genummerde regels vervolgens opslaat in een nieuw bestand. De naam van het input-bestand wordt gevraagd aan de gebruiker, evenals de naam van het nieuwe bestand dat je programma zal aanmaken. Elke regel in het output-bestand moet beginnen met het regelnummer, gevolgd door een dubbele punt en een spatie, gevolgd door de regel uit het input-bestand.

## Oefening 5

In deze oefening maak je een Python-programma dat de langste woorden in een bestand identificeert. Jouw programma moet een passend bericht weergeven dat de lengte van het langste woord bevat, samen met alle woorden van die lengte die in het bestand voorkwamen. Behandel elke groep karakters die geen witruimte zijn als een woord, zelfs als deze cijfers of leestekens bevat.

## Oefening 6

Schrijf een programma dat het woord (of de woorden) weergeeft die het vaakst in een bestand voorkomen. Je programma begint met het vragen van de naam van het bestand dat moet ingelezen worden aan de gebruiker. Dan zou het elke regel in het

bestand moeten verwerken. Elke regel moet worden opgesplitst in woorden en alle leestekens vóór of achter moeten uit elk woord worden verwijderd. Je programma negeert ook de hoofdletters bij het tellen hoe vaak elk woord voorkomt: 'Favoriet' en 'favoriet' bvb zijn dus hetzelfde woord.

## Oefening 7

Maak een programma dat alle door de gebruiker ingevoerde getallen bij elkaar optelt. Het negeert daarbij alle invoer die niet uit een geldig getal bestaat (bvb een woord). Je programma geeft de som weer nadat elk nummer is ingevoerd. Het geeft een passend bericht weer telkens de gebruiker iets anders dan een getal heeft ingevoerd. Verlaat het programma wanneer de gebruiker een lege regel invoert. Zorg ervoor dat je programma correct werkt voor zowel integers als floats.

## Oefening 8

Python gebruikt een hashtag om het begin van een commentaarregel aan te geven. Deze regel bestaat vervolgens volledig uit commentaar. In deze oefening maak je een programma dat alle opmerkingen uit een Python-bestand verwijdert. Controleer elke regel in het bestand om te bepalen of een #-teken aanwezig is. Als dit het geval is verwijdert het programma alle tekens van het #-teken tot het einde van de regel (we negeren de situatie waarin het commentaarteken ergens in het midden van een regel voorkomt). Sla het gewijzigde bestand op met een nieuwe naam. Zowel de naam van het invoerbestand als de naam van het uitvoerbestand moeten aan de gebruiker worden gevraagd. Zorg ervoor dat er een passend foutbericht wordt weergegeven als zich een probleem voordoet bij het openen van één van de bestanden.

## Oefening 9

Schrijf een programma dat een bestand leest dat een lijst met woorden bevat, er willekeurig twee selecteert en ze samenvoegt om een wachtwoord te produceren. Zorg er bij het maken van het wachtwoord voor dat de totale lengte tussen 12 en 15 tekens is en dat elk gebruikt woord ten minste drie letters lang is. Zet elk woord in het wachtwoord met een hoofdletter, zodat de gebruiker gemakkelijk kan zien waar het ene woord eindigt en het volgende begint. Tot slot geeft je programma het wachtwoord weer.