

Python Programming - Labo 1

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.

Oefening 1: van start gaan met Python

Installeer Python 3.10.2 via [Python.org](https://python.org) en Visual Studio Code via <https://code.visualstudio.com/>. Je installeert tot slot ook de Python en Pylance extensies in VSCode. Zorg voor een degelijke folderstructuur voor dit vak en maak je eerste script: hallowereld.py. Met de print()-functie schrijf je een boodschap aan de wereld en voert ze uit via VSCode.

Oefening 2: een visitekaartje afgeven

Maak een programma dat een virtuele business card van jezelf weergeeft. Je gebruikt daarvoor het print-commando. De informatie, die ook uit wat interesses mag bestaan, bepaal je zelf. Je mag hiervoor enkele variabelen creëren en gebruik maken van f-strings en niet-afdrukbare tekens als "\n" om ze te formatteren. Dit programma verwacht geen input van de gebruiker.

Oefening 3: Hallo!

Schrijf een programma dat de gebruiker naar zijn of haar naam vraagt. Daarna vraagt het programma naar de interesses van de gebruiker. Het programma antwoordt vervolgens met een hallo-boodschap waarin de gevraagde gegevens verwerkt zitten.

Oefening 4: oppervlakte van een kamer

Schrijf een programma dat de gebruiker vraagt om een lengte en breedte van een kamer in te geven (twee afzonderlijke variabelen). Eens deze waarden ingelezen zijn berekent je programma de oppervlakte van de kamer. De lengte en de breedte zullen ingegeven worden als komma-getallen. Voeg eenheden toe in de output-

boodschap (meter). Gebruik de float()-functie om de ingegeven waarden om te zetten van string naar float.

Oefening 5: statiegeld berekenen

Je schrijft een programma dat berekent hoeveel statiegeld moet worden uitbetaald aan iemand die flessen komt inleveren. Er kunnen kleine flesjes worden teruggegeven en grote flessen. De kleine leveren 0,12eur statiegeld op, de grote 0,25eur. Je maakt van deze waarden constanten. Het programma vraagt hoeveel kleine en grote flessen worden ingeleverd en berekent dan hoeveel de gebruiker zal ontvangen. Je werkt met floats en rondt het eindgetal af tot op 2 cijfers na de komma.

Tip: Voor floating point getallen is het in de praktijk handig om de round() functie te gebruiken. De round() functie zorgt ervoor dat de ingevoerde getallen afgerond worden naar de door jouw gedefinieerde positie achter de komma. De functie vereist twee ingevulde parameters: `round(getal, aantal decimalen)`. De standaard waarde voor een lege bijgevoegde waarde is een compleet afgerond getal. Dit houdt in dat wanneer de tweede parameter leeg blijft dan gaat de functie er van uit dat er afgerond dient te worden op hele getallen.

Oefening 6: Som van de eerste n positieve gehele getallen

Schrijf een programma dat een positief geheel getal als input vraagt en dan als resultaat de som weergeeft van alle gehele getallen van 1 tot n . Je kan dit berekenen met de volgende formule:

$$som = (n)(n + 1)/2$$

Oefening 7: samengestelde rente

Beeld je in dat je net een nieuwe bankrekening hebt geopend die je 1,2% rente per jaar geeft. Deze rente wordt uitbetaald op het einde van het jaar en toegevoegd aan je bankrekening. Schrijf een programma dat begint met het lezen van het bedrag dat je bij het begin van het jaar op je rekening hebt gezet. Het programma rekent dan uit (en toont) de stand van de rekening na 1, 2 en 3 jaar. Toon elk bedrag met wat uitleg en afgerond tot op 2 cijfers na de komma.

Oefening 8: volume van een cilinder

Het volume van een cilinder kan berekend worden door de oppervlak van de cirkelvormige basis te vermenigvuldigen met zijn hoogte. Maak een programma die de straal en de hoogte van de cilinder opvraagt aan de gebruiker en vervolgens het volume berekent. Toon het resultaat afgerond tot op 1 cijfer na de komma.

Oefening 9: eenheden van tijd

Maak een programma dat een lengte van tijd opvraagt van een gebruiker als een reeks dagen, uren, minuten en seconden (afzonderlijk opgevraagd). Bereken en toon het totale aantal seconden dat deze tijd vertegenwoordigt.

Oefening 10: Celsius naar Fahrenheit en Kelvin

Schrijf een programma dat begint met het opvragen van een temperatuur in Celsius. Vervolgens toont je programma deze temperatuur in graden Fahrenheit en Kelvin. De berekeningen die je hiervoor nodig hebt vind je gemakkelijk op het internet.