

Python developer – 1ste jaar

Examen module “Basis Python”

CURSIST

Naam en voornaam: _____

Datum: _____

INTRO

Begin eerst met je naam, voornaam en datum in te vullen hierboven, zo wordt dit niet vergeten.

Het volledige examen is een open boek examen. Je mag online opzoeken doen, indien nodig. Je mag alle materiaal gebruiken dat ter jouwe beschikking staat. Enkel raad aan elkaar vragen, is niet toegestaan.

Aangezien de antwoorden allemaal kleine programmaatjes zijn, moet alles ingediend worden via Git. Je maakt 1 bestand, genaamd examen.py. Voor elke oefening maak je een nieuwe functie genaamd oefening_x(), waarbij x uiteraard het nummer van de oefening is. De benodigde bestanden om de oefeningen te kunnen maken, staan ook online klaar in de Git repo.

Hiervoor heb je toegang nodig tot je Git repo die gebruikt is voor de vorige examens (examen basis Python). Wanneer je gedaan hebt, verwittig je de leraar zodat hij de push naar de repo kan nakijken.

Ter info: de repo van iedere cursist staat online op:

<https://github.com/syntra-vindevoy/python1-2024-25-voornaam>

Wanneer je inlogt op GitHub (online), zal de repo in je lijst van repositories staan. Ze zijn niet meer “archived” tijdens dit examen. Indien de repo toch niet zichtbaar is, vraag je raad aan de lesgever.

OEFENINGEN

1) Maak een functie die als parameter een lijst krijgt van willekeurige getallen tussen 0 en 100 (het kunnen komma getallen zijn). Het aantal elementen in de lijst is niet op voorhand bekend, maar je mag ervan uitgaan dat alle elementen wel degelijk getallen zijn, dus je moet geen verificatie inbouwen.

De functie moet hetvolgende retourneren:

- het aantal getallen in de lijst
- de som van de getallen
- het kleinste getal uit de lijst
- het grootste getal uit de lijst
- het gemiddelde berekend als volgt: de som van de getallen gedeeld door het aantal getallen
- het gemiddelde berekend als volgt: de som van het kleinste en grootste getal gedeeld door 2

Het type van de retour moet een tuple zijn. Zorg er uiteraard voor dat je functie optimaal is en niets dubbel berekent.

(10 punten)

2) Maak een functie die als parameters (2!) je naam en voornaam krijgt. De functie moet de letters van beiden in alfabetische volgorde terug sturen. De functie moet hoofdletter ongevoelig zijn.

Vb: Yves Vindevogel → deeeegilnosvvvy

(10 punten)

Je krijgt 2 punten als je dit in een one-liner schrijft. Deze one-liner indicatie kan je ook zien als tip over hoe je de functie schrijft.

3) Maak een functie die als input de lijst met voornamen van alle cursisten krijgt. De functie moet een dictionary retourneren met als key de voornaam en als value het email adres van de cursist. Het email adres van de cursist is "voornaam@syntra.be". Je moet geen rekening houden dat in bepaalde gevallen er cursisten zouden kunnen zijn met dezelfde voornaam.

(10 punten)

4) Lees het bestand "namen.txt" in. Deze lijst bevat per lijn een naam. Schrijf een nieuw bestand weg, genaamd "uniek.txt" met daarin de namen zonder dat er nog dubbels in de lijst staan. Met andere woorden, als je in het oorspronkelijke bestand een naam tegen komt die al voorkwam, mag je die niet wegschrijven. Opgelet: de namen in het nieuwe bestand moeten in dezelfde volgorde staan als in het oorspronkelijke bestand. En hou rekening met hoofdletters. Een naam die begint met kleine letter of hoofdletter is een dubbel.

(20 punten)

5) Het buitenlands bedrijf Gaspromo wil een overzicht van zijn export naar bepaalde Europese landen. Daarvoor hebben de data analisten 2 bestanden aangeleverd:

Het eerste bestand: landen.txt bevat per lijn de land code (32 voor België) en de naam van het land. Daarenboven bevat het een derde element: het minimum dat dit land moet afnemen per maand. Dit minimum zal van belang zijn voor de uiteindelijke output.

Het tweede bestand, verbruik.txt, bevat per land een verbruik per maand. Het eerste element is de land code, het tweede element is het jaar van het verbruik, het derde element is de maand van het verbruik (januari: 1, december: 12), het vierde element is het effectieve verbruik van die maand voor dit land.

Als output moet je nieuw bestand schrijven "2024.txt", met als inhoud per land één lijn met daarin de land code als eerste element, de naam van het land als tweede element en de totale omzet van dat land voor het jaar 2024 als derde element. Als vierde getal moet je het verbruik bepalen dat aangerekend wordt, maar niet verbruikt is. Dit vierde getal ontstaat wanneer een land voor een bepaalde maand (zie ook verder) zijn minimum afname niet verbruikt heeft. Het land wordt dan het minimum aangerekend, maar kreeg minder geleverd. De som van deze verschillen is dit vierde getal. Het kan dus best zijn dat dit getal 0 is voor een bepaald land.

De data analisten stonden onder grote druk en hebben een bestand aangeleverd dat mogelijks ook al de data van 2025 bevat. Deze data mag uiteraard niet meegeteld worden.

En, inderdertijd werd afgesproken dat er elke maand een minimum afname zou zijn. Dit minimum is voor alle maanden gelijk, maar per land bepaald. Je kan dit minimum terug vinden in het eerste bestand: landen.txt. Indien een land voor een bepaalde maand een verbruik heeft dat lager ligt dan de minimum afname, dan moet je voor het totaal te berekenen de minimum afname gebruiken en niet het effectieve verbruik. Het verschil tussen beide getallen zal ook gebruikt worden voor het vierde getal in de output per land.

Je moet de output sorteren op naam van het land, niet op land code.

Tip 1: je kan heel gemakkelijk informatie over een land zoeken als je die informatie eerst in een dictionary steekt.

Tip 2: gestructureerde output (4 elementen per land) vraagt om een dictionary

(50 punten in totaal)

(10 punten indien de output de correcte combinatie bevat van land code met de naam van het land)

(10 punten voor het berekenen per land van het verbruik)

(8 punten voor het rekening houden met de minimum afname per maand)

(5 punten voor het filteren van data die niet in 2024 is)

(7 punten voor berekening 4de element)

(10 punten voor de correcte sortering in de output)