



Week 5

Doelstellingen

Je bent in staat om te werken met de datastructuur tuple, dictionary, set.
Je bent in staat om aan functies deze datastructuren door te geven of terug te krijgen.

Afspraken

Eindniveau - oefeningen

Ben je een student MCT, dan beheers je de oefeningen tot moeilijkheidsgraad “D”
Ben je een student MIT, dan beheers je de oefeningen tot moeilijkheidsgraad “C”

GitHub

Alle oplossingen van week 3 dienen op Github geplaatst te worden. Zie week 1 voor procedure.

Open hiervoor het project dat je vorige week op GitHub geplaatst hebt. Maak een submap voor week 3. Plaats hierin je oplossingen van deze week. Na elke oefening kan je een 'commit & push' doen zodat jouw versie op GitHub steeds aangepast wordt. Geef telkens een gepaste message mee.

Niet afgewerkte oefeningen werk je thuis verder af: voer regelmatig een 'push & commit'-opdracht uit zodat alle oplossingen op je github-repository beschikbaar zijn.

Bij een programmeertaal zoals Python onder de knie krijgen is veelvuldig oefenen essentieel en een noodzakelijke voorwaarde. Daarom vind je in elk labo-document nog twee extra onderdelen. Deze worden als volgt aangeduid.

Uitbreidingsoefeningen - eigen onderzoek

Dit onderdeel gaat verder dan de geziene leerstof van deze week. Vaak zijn de opdrachten net iets moeilijker dan hetgeen je in het labo deed. Je zal de Python [documentation](#) en Google nodig hebben voor dit onderzoek.

We motiveren iedereen om dit (thuis) iedere week voor te bereiden. Je onderzoekt in dit onderdeel een onderwerp die de volgende weken terugkomt in de theorie of het labo.

Oefeningen voor thuis

In dit onderdeel vind je analoge oefeningen zoals je reeds in het labo maakte. Deze oefeningen hebben dezelfde moeilijkheidsgraden zoals in het labo. Het is pas door de oefeningen thuis “alleen” te maken dat je de leerstof zich eigen maakt. Loop je vast bij een oefening? Herbekijk de theorie, kijk of je een analoge oefening terugvindt die je maakte tijdens het labo. Lukt het nog steeds niet? Kom met je voorbereiding naar het monitoriaat!



Feedback labo week 4

Lees nog even de algemene opmerkingen uit de vorige theorieles na:

https://leho-howest.instructure.com/courses/6938/files/526658?module_item_id=120052

Reminder uit vorig labo: via de sneltoets ' **Shift+Alt+F** ' kan je jouw code onmiddellijk volgens de PEP8-stijl formatteren.

Voor wie fan is van tips & trics bij het gebruik van Visual Studio Code:

<https://code.visualstudio.com/docs/getstarted/tips-and-tricks>

Oefeningen

Vermeld in commentaar telkens de opgave!

Gebruik van Tuples

Oef 01

Maak volgende tuples aan:

- tuple1 met de elementen 1NMCT, 2NMCT, 3NMCT
- tuple2 met de elementen 1Devine, 2Devine, 3Devine
- tuple3 met elementen van verschillende datatypes: "test", True, 3.2 , 1

Geef een antwoord op onderstaande vragen dmv een kort codevoorbeeld op bovenstaande tuples:

- Wat is het effect als we in een tuple een element op een bepaalde positie willen wijzigen of verwijderen?
- Kan je op een tuple de methodes insert/append (zoals bij List) gebruiken?
- Kan je via de plus-operator een tuple bij een andere tuple voegen?
- Kan je de vermenigvuldigingsoperator op een tuple gebruiken?

Oef 02

Maak één functie 'print_tuple' die de verschillende elementen overloopt waarbij elk element samen met zijn index wordt afgeprint. De functie heeft als parameters een naam (voor de tuple) en de tuple zelf.

Voorbeeld:

```
Verzameling NMCT:
1NMCT zit op positie 0
2NMCT zit op positie 1
3NMCT zit op positie 2
Verzameling Devine:
1Devine zit op positie 0
2Devine zit op positie 1
3Devine zit op positie 2
```

Gebruik van Dictionarys

Oef 03

Maak volgende 2 dictionaries aan. Ze stellen beide de verschillende studentenaantallen per opleidingsjaar voor.

- 'nmct' met de elementen {"1NMCT": 131, "2NMCT": 88, "3NMCT": 77}
- 'devine' met de {"1Devine": 123, "2Devine": 98, "3Devine": 55}



Geef een antwoord op onderstaande vragen d.m.v. een kort codevoorbeeld op bovenstaande dictionaries:

- Wat is het effect van het print-commando op een dictionary?
- Hoe kan je een element uit de dictionary opvragen?
- Hoe voeg je een nieuw element aan een dictionary toe?
- Wat gebeurt er als een nieuw element met dezelfde key aan een dictionary toegevoegd wordt?
- Hoe controleer je of een key in een dictionary reeds in gebruik is?
- Hoe kan je een key (met value) uit een dictionary verwijderen? Wat als key niet aanwezig is?

Oef 04

Maak nu één functie 'print_dictionary' die de verschillende elementen overloopt waarbij telkens key & value samen op één lijn worden afgeprint. De functie heeft als parameters een naam (voor de dictionary) en de dictionary zelf. Voorbeeld:

```
Dictionary NMCT:
key: 3NMCT -> value: 77
key: 1NMCT -> value: 101
key: 2NMCT -> value: 88
Dictionary Devine:
key: 3Devine -> value: 55
key: 1Devine -> value: 123
key: 2Devine -> value: 98
```

Oef 05

Maak een nieuwe lege dictionary 'Howest' aan. Voeg bovenstaande dictionaries eraan toe.

Oef 06

Maak een methode 'voeg_nieuw_element_toe' aan die een nieuw element aan een dictionary toevoegt nadat er eerst gecontroleerd wordt of de key nog niet in gebruik is. Indien de key wel reeds in gebruik is, wordt er een foutboodschap afgeprint.

De methode heeft als parameters de key, de value en de dictionary zelf.

Test dit uit door aan dictionary Howest achtereenvolgens toe te voegen

- Element met key '1IPO' met waarde '51'
- Element met key '1IPO' met waarde '10'

Print nadien de dictionary Howest af (via methode uit eerdere oefening).

```
Dictionary Howest:
key: 3Devine -> value: 55
key: 2NMCT -> value: 88
key: 1IPO -> value: 51
key: 3NMCT -> value: 77
key: 2Devine -> value: 98
key: 1Devine -> value: 123
key: 1NMCT -> value: 101
Toevoegen mislukt. Key '1IPO' reeds aanwezig
```

Oef 07

Maak een functie 'tel_voorkomen_woorden' met als parameter een string. De functie geeft een dictionary terug waarbij de keys de verschillende woorden uit de zin zijn en de bijhorende values het aantal keer is dat het woord in de zin voorkomt. Print nadien de dictionary af.



Onderzochte zin:
dit is nmct is het niet waar ? Uiteraard NMCT

```
Dictionary Woorden_dic:
key: waar -> value: 1
key: niet -> value: 1
key: is -> value: 2
key: uiteraard -> value: 1
key: dit -> value: 1
key: het -> value: 1
key: ? -> value: 1
key: nmct -> value: 2
```

Of 08

Maak een python applicatie die in staat is om de binnengekomen feedbackscores (van een evenement) te verwerken. Hierbij wordt aan een medewerker gevraagd alle feedbackscores één voor één in te geven. De medewerker kan afsluiten met een lege score. Op het einde worden de scores geteld en afgeprint.

```
Geef de verschillenden evaluatiecijfers door (sluit af met lege waarde)
Uitmuntend: 5, Zeer goed: 4, Goed: 3, Voldoende: 2, Onvoldoende: 1, Zwak: 0
Geef de nieuwe feedbackscore op: 3
Geef de nieuwe feedbackscore op: 5
Geef de nieuwe feedbackscore op: 3
Geef de nieuwe feedbackscore op: 4
Geef de nieuwe feedbackscore op: 5
Geef de nieuwe feedbackscore op: 1
Geef de nieuwe feedbackscore op: 1
Geef de nieuwe feedbackscore op: 2
Geef de nieuwe feedbackscore op: 3
Geef de nieuwe feedbackscore op: 3
Geef de nieuwe feedbackscore op: 2
Geef de nieuwe feedbackscore op: 2
Geef de nieuwe feedbackscore op: 5
Geef de nieuwe feedbackscore op: 5
Geef de nieuwe feedbackscore op:
De gegeven zijn verwerkt. Dit is het resultaat:
score: 3 -> aantal: 4
score: 1 -> aantal: 2
score: 2 -> aantal: 3
score: 4 -> aantal: 1
score: 5 -> aantal: 4
score: 0 -> aantal: 0
```

Of 9

Maak een python applicatie dat voor een groep personen (studenten/docenten) de snelheid in de straat 'Graaf Karel de goedelaan' registreert.

Stap 0: we starten met een **dictionary** waarin voor elke persoon de snelheid nog op nul staat.

Stap 1: als simulatie wordt voor elke persoon een willekeurige snelheid tussen 10 en 70km/u gekozen. Het opvullen van de dictionary gebeurt in een afzonderlijke functie '**registreer_snelheidsmeting**'

Stap 2: print alle snelheden per persoon af. Vermeld ook de straatnaam. Gebruik hiervoor een functie '**print_snelheden_personen**'

Stap 3: vraag aan de gebruiker de maximum toegelaten snelheid in de betrokken straat.

Stap 4: filter de overtredingen uit de dictionary. Gebruik hiervoor een afzonderlijke functie '**filter_overtredingen**'. Geef als parameter de startdictionary en de maximum toegelaten snelheid mee. Wat geef deze functie terug?

Stap 5: print de boetes uit voor de overtredingen uit. De boete wordt als volgt berekend: 10€ per km boven de toegelaten snelheid. Gebruik hiervoor opnieuw een afzonderlijke functie '**print_boetes**'



Een voorbeeld:

Geregistreerde snelheden in de straat Graaf Karel de goedelaan:

Stijn	59 km/u
Johan	29 km/u
Dieter	49 km/u
Christophe	39 km/u

Geef de maximum toegelaten snelheid op:> 30

De te innnen boetes zijn:

Stijn	boete: 290€
Dieter	boete: 190€
Christophe	boete: 90€



Uitbreiding – Eigen onderzoek

Onderzoek zelf de datastructuur 'Set':

Maak volgende set aan. De set stelt enkele postcodes uit West-Vlaanderen voor.

'postcodes' met de elementen (8500,8531,8010,8850,8370)

Geef een antwoord op onderstaande vragen dmv een kort codevoorbeeld op bovenstaande set:

- Wat is het effect van het print-commando op een set?
- Hoe voeg je een nieuw element aan een set toe?
- Kan je een element toevoegen dat reeds in de set aanwezig is?
- Kan je een element van een set via een positie benaderen (zoals een list)?
- Hoe kan je element per element van een set overlopen?
- Hoe kan je een element uit een set verwijderen?



Oefeningen voor thuis



Studeer voor je Q!



Studeer voor je Q!