

Week 5

Doelstellingen

Je bent in staat om te werken met de datastructuur tuple, dictionary, set.
Je bent in staat om aan deze functies 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 5 dienen op Github geplaatst te worden. Volg hiervoor de procedure uitgelegd op Leho. Via Github krijg je ook alle bronmateriaal voor deze opgave.

Om je repository in onze Github-classroom aan te maken, klik je op volgende link:
<https://classroom.github.com/a/-SCmu6gi>

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 theorieles na. Je vindt ze op Leho terug.
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 1MCT, 2MCT, 3MCT
- tuple2 met de elementen 1MIT, 2MIT
- 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 MCT:
1MCT zit op positie 0
2MCT zit op positie 1
3MCT zit op positie 2
Verzameling MIT:
1MIT zit op positie 0
2MIT zit op positie 1
```

Gebruik van dictionaries

Oef 03

Maak volgende 3 dictionaries aan.

Ze stellen beide de verschillende studentenaantallen per opleidingsjaar voor.

- 'mit' met de elementen {"1MIT": 58, "2MIT": 36}
- 'mct' met de elementen {"1MCT": 131, "2MCT": 88, "3MCT": 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 MCT:
key: 1MCT -> value: 101
key: 2MCT -> value: 88
key: 3MCT -> value: 77
```

```
Dictionary MIT:
key: 1MIT -> value: 58
key: 2MIT -> value: 36
```

```
Dictionary Devine:
key: 1Devine -> value: 123
key: 2Devine -> value: 98
key: 3Devine -> value: 55
```

Oef 05

Maak een nieuwe lege dictionary 'Howest' aan.

Voeg bovenstaande dictionaries eraan toe.

Oef 06

Maak een methode 'voeg_nieuw_element_toe' 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).

```
Toevoeging nieuw element geslaagd.
```

```
Toevoegen mislukt. Key 1IPO reeds aanwezig!
```

```
Dictionary Howest:
key: 1MCT -> value: 101
key: 2MCT -> value: 88
key: 3MCT -> value: 77
key: 1MIT -> value: 58
key: 2MIT -> value: 36
key: 1Devine -> value: 123
key: 2Devine -> value: 98
key: 3Devine -> value: 55
key: 1IPO -> value: 51
```

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.

```
Dit is Howest, is het niet zo? Uiteraard, welkom!
```

```
Dictionary dic_woorden:
key: dit -> value: 1
key: is -> value: 2
key: howest -> value: 1
key: het -> value: 1
key: niet -> value: 1
key: uiteraard -> value: 1
key: welkom -> value: 1
```

Oef 08

Maak een pythonapplicatie 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 verschillende evaluatiecijfers door (sluit af met lege waarde)

```
Uitmuntend: A, Zeer goed: B, Goed: C, Voldoende: D, Onvoldoende: E, Zwak: F
Geef de nieuwe feedbackscore op: B
Geef de nieuwe feedbackscore op: C
Geef de nieuwe feedbackscore op: A
Geef de nieuwe feedbackscore op: A
Geef de nieuwe feedbackscore op: D
Geef de nieuwe feedbackscore op: A
Geef de nieuwe feedbackscore op: B
Geef de nieuwe feedbackscore op: B
Geef de nieuwe feedbackscore op: C
```

```

Geef de nieuwe feedbackscore op: D
Geef de nieuwe feedbackscore op: E
Geef de nieuwe feedbackscore op: D
Geef de nieuwe feedbackscore op: D
Geef de nieuwe feedbackscore op: B
Geef de nieuwe feedbackscore op: B
Geef de nieuwe feedbackscore op: C
Geef de nieuwe feedbackscore op: A
Geef de nieuwe feedbackscore op: A
Geef de nieuwe feedbackscore op: B
Geef de nieuwe feedbackscore op: D
Geef de nieuwe feedbackscore op:
De gegevens zijn verwerkt. Dit is het resultaat:
score: A -> aantal: 5
score: B -> aantal: 6
score: C -> aantal: 3
score: D -> aantal: 5
score: E -> aantal: 1
score: F -> aantal: 0

```



Oef 9

Maak een pythonapplicatie die voor een groep personen (studenten/docenten) een snelheidscontrole in de straat 'Sint Martens Latemlaan' simuleert.

Stap 0: we starten met een een **list** met personeelsleden.

```
list_personeel = ["Stijn", "Gilles", "Dieter", "Christophe", "Marie"]
```

Stap 1: als simulatie wordt voor elke persoon een willekeurige snelheid tussen 10 en 70km/u gegenereerd. Hiervoor gebruik je de functie '**registreer_snelheidsmeting**'. Het resultaat van deze functie is een **dictionary**. De **key** is het **personeelslid**, de **value** de willekeurige **snelheid**

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:

Geregistreeerde snelheden in de straat Sint Martens Latemlaan:

Stijn	62 km/u
Gilles	13 km/u
Dieter	13 km/u
Christophe	24 km/u
Marie	25 km/u

Geef de maximum toegelaten snelheid op:> 30

De te innnen boetes zijn:

Stijn boete: 320€



Oef 10

Gegeven een dictionary waarbij de keys verschillende personen voorstellen en de values de bijhorende landen die bezocht werden.

Bijvoorbeeld:

```
mijn_team = {
    "Stijn": ["Frankrijk", "Zwitserland", "Oostenrijk", "Nederland"],
    "Christophe": ["USA", "Frankrijk", "Duitsland"],
    "Dieter": ["Nederland", "Duitsland", "Zwitserland", "Oostenrijk"],
    "Gilles": ["UK", "Spanje", "Portugal", "Frankrijk", "Nederland"],
    "Marie": ["USA", "Spanje", "Lichtenstein", "Vaticaanstad"]
}
```

Schrijf een functie **'filter_op_land'** met als parameters bovenstaande dictionary en een gezocht land. De functie geeft in een list alle personen terug die dit land bezocht hebben.

Geef een land op:> Nederland

Deze personen hebben reeds dit land bezocht: ['Stijn', 'Dieter', 'Gilles']



Oef 11

Gegeven een dictionary waarbij de keys verschillende personen voorstellen en de values de bijhorenden landen die bezocht werden. Bijvoorbeeld:

```
mijn_team = {
    "Stijn": ["Frankrijk", "Zwitserland", "Oostenrijk", "Nederland"],
    "Christophe": ["USA", "Frankrijk", "Duitsland"],
    "Dieter": ["Nederland", "Duitsland", "Zwitserland", "Oostenrijk"],
    "Gilles": ["UK", "Spanje", "Portugal", "Frankrijk", "Nederland"],
    "Marie": ["USA", "Spanje", "Lichtenstein", "Vaticaanstad"]
}
```

Schrijf een functie **'geef_populariteit_landen'** met als parameters bovenstaande dictionary. De functie geeft een nieuwe dictionary terug waarbij de landen de keys zijn, en de values telkens het aantal bezochte personen. Print nadien alles netjes af.

Populariteit van de verschillende landen:

```
Land: Frankrijk -> aantal: 3
Land: Zwitserland -> aantal: 2
Land: Oostenrijk -> aantal: 2
Land: Nederland -> aantal: 3
Land: USA -> aantal: 2
Land: Duitsland -> aantal: 2
Land: UK -> aantal: 1
Land: Spanje -> aantal: 2
Land: Portugal -> aantal: 1
Land: Lichtenstein -> aantal: 1
Land: Vaticaanstad -> aantal: 1
```



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?