**Start les** (5 min)

* Doel van de les: Werken met functies
* Functies zijn langsgekomen in het hoorcollege.

**Opfris week 2** (15 min)

* If then else
* variabelen
* Bespreken opdracht 8van week 2 eerste werkcollege

**Introductie functies** (10 min)

* Belangrijke functies
* Imports

**Main functie** (10 min)

* Opdracht 1
* main functie
* assertions

**Signature en docstring** (5 min)

**Opdracht 2** (15 min)

* Studenten moeten eerst bepalen wat de functie doet.
* Daarna kunnen ze de code uitvoeren en met behulp van assertions erachter komen of ze gelijk hebben of niet.
* Zodra ze overtuigd zijn maken ze een docstring

**Opdracht 3** (15 min)

* Voor het a gedeelte, leg nadruk op Probeer, Plan, Programeer.
  + Eerst het probleem op papier oplossen. Hoe reken je zelf dagen/uren/minuten uit met een rekenmachine
  + Zet dat om in instructies. Hoe leg je iemand anders uit hoe het berekend moet worden.
  + Ga dan programmeren
* b. Dit is een voorbeeld waarbij de main functie wordt gebruikt. Een functie returned het antwoord, de main functie print het antwoord.

**Opdracht 4** (15 min)

* Gebruik van booleans als variabelen
* Teruggeven van booleans.
* Vergelijk verschillende oplossingen van studenten.
  + Welke is efficiënter?
  + Welke is leesbaarder?

**Antwoorden**

**Opdracht 1**

1. 42  
   “wauw!wauw!”
2. –
3. Meerdere antwoorden mogelijk

**Opdracht 2**

1. fraction staat voor een fractie, een deel van een geheel dat van een laagste waarde low naar een hoogste waarde hi loopt. Bijvoorbeeld, in het geval van een taart zou dit van waarde 0 (geen taart) naar 2 (twee, hmmm!) kunnen lopen, waar fractie 0.75 gelijk staat aan 1.5 (anderhalve taart, ook fijn).  
     
   Wat hier uit volgt is dat als fraction 0 is low moet worden teruggegeven. Als fraction 1 is zal hi moeten worden teruggegeven. Waarden voor fraction tussen 0 en 1 geven resultaten tussen low en hi.  
   Sterker nog, fraction kan een waarde onder 0 krijgen en het resultaat zal dan kleiner zijn dan low. Ook kan fraction een waarde boven 1 krijgen, en het resultaat zal dan groter zijn dan hi (puristen zouden dit waarschijnlijk extrapolatie in plaats van interpolatie, noemen).
2. –
3. Meerdere antwoorden mogelijk
4. Meerdere antwoorden mogelijk

**Opdracht 3**

1. Voorbeeld uitwerking:

def convert\_from\_seconds(s):

"""Een getal naar dagen, uren, minuten en seconden

Zet een getal om naar naar een lijst van

[days, hours, minutes, seconds]

input s: een int

"""

days = s // (24 \* 60 \* 60) # aantal dagen

s = s % (24 \* 60 \* 60) # restant s

hours = s // (60 \* 60) # aantal uren

s = s % (60 \* 60) # restant s

minutes = s // 60 # aantal minuten

s = s % 60 # restant s

return [days, hours, minutes, s]

tijd = convert\_from\_seconds(100000)

print(tijd[0], "dagen")

print(tijd[1], "uren")

print(tijd[2], "minuten")

print(tijd[3], "seconden")

**Opdracht 4**

1. Voorbeeld uitwerking

def checkends(s):

    return s[0] == s[-1]

Dit is de kortste uitwerking. Andere uitwerkingen met if statements kunnen ook.