



# Python

---

Informatica

Renske



# Planning komende tijd

|                                                                      |                                                                                                                                         |                                                                                                                                                          |
|----------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| Week 3:<br>12-09                                                     | Lists: Theorie en opgaven<br>Lists: Afsluitende opgaven<br>Lists (met For): Theorie en opgaven<br>Lists (met For): Afsluitende opgaven  | Lists: sheets<br>Lists: Leerdoelen<br>Lists (met For): sheets<br>Lists (met For): Leerdoelen<br>Toetsvoorbereiding (alle stof behalve geneste for-loops) |
| Week 4:<br>19-09                                                     | Geneste Lijsten: Theorie en opgaven<br>Geneste Lijsten: Afsluitende opgaven<br>Toetsvoorbereiding (alle stof behalve geneste for-loops) | Geneste Lijsten: sheets<br>Geneste Lijsten: Leerdoelen                                                                                                   |
| Week 5:<br>26-09                                                     | <b>In de les: TOETS over Python (alle stof tot nu toe)</b><br>Daarna beginnen met PO Galgje                                             |                                                                                                                                                          |
| Week 6:<br>3-10                                                      | PO Galgje                                                                                                                               | Uitleg Pycharm<br>Sheets met voorbeeld ontwerp (stroomdiagram en data) en uitleg PO<br>Zeeslag                                                           |
| Activiteitenweek (geen les) en<br>Herfstvakantie<br>(7-10 t/m 18-10) |                                                                                                                                         |                                                                                                                                                          |
| Week 8:<br>24-10                                                     | PO Galgje                                                                                                                               |                                                                                                                                                          |
| Afrondingsweek<br>(28-10 t/m 01-11)                                  | <b>Deadline PO Galgje (code + verslagje): 2-nov 23:59</b><br>Eindcijfer: Tussentijsds toets en PO (tellen ieder voor de helft<br>mee)   | -                                                                                                                                                        |



# FOR loop & Lijsten

---

- Lijsten
- For-loop

Na deze les kun je:

- For-loop te gebruiken om lijst te doorlopen



# Oefenen met for en while

---

Opdracht: Zet deze while om in een for-loop

```
getal = 3
while getal < 6:
    print(getal)
    getal += 1
```

Antwoord:

```
for getal in range(3,6):
    print(getal)
```

# Lijsten

- Een "list" (Engelse woord voor "lijst") is een geordende verzameling van data elementen

## Voorbeelden:

namenLijst = ["Ben", "Piet", "Charlie", "Dan", "Edward"]

leeftijdenLijst = [5, 14, 8, 12, 9]

cijferLijst = [8.0, 5.6, 6.0, 8.1 ]

- In Python mag je van alles in een lijst stoppen,
  - ook gemengd (bv. getallen en strings)
  - Bv. rommeltje = [3, "banana", '\$', "Hello World!"]





# Index van een lijst

- Elk element in een lijst heeft een index (positie).
- Tellen begint altijd bij 0
- Lijst eindigt bij index van lengte -1      hier: `len(fruits)-1`

De lijst:

```
fruitLijst = ["appels", "peren", "mandarijnen", "bananen"]
```

index:      0                      1                      2                      3

## Waarde opvragen:

- `fruitLijst[0]`                      antwoord: "appels"
- `fruitLijst[3]`                      antwoord: "bananen"

## Waarde toekennen:

- `fruitLijst[0] = "aardbeien"`

gevolg: `["aardbeien", "peren", "mandarijnen", "bananen"]`



# Lijsten kun je aanpassen

- Waarden toekennen:

```
fruitLijst = ["appel", "banaan"]
```

```
[ "appel", "banaan" ]
```

- Iets toevoegen aan lijst → lijst wordt langer

```
fruitLijst += ["kers"]    #voegt "kers" toe achteraan lijst
```

- Iets in een lijst aanpassen

```
#vervangt element op positie 0 (dus appel) met aardbei
```

```
fruitLijst [0] = "aardbei"
```

```
[ "appel", "banaan", "kers" ]
```

```
[ "aardbei", "banaan", "kers" ]
```

- Iets uit lijst verwijderen → lijst wordt korter

```
fruitLijst.remove( "banaan" ) #verwijdert eerste voorkomen
```

```
[ "aardbei", "kers" ]
```



# Standaard List functies

numlist = [12, 1, 243, 1]

- Bepaal de grootste getal uit de lijst:

`max( numlist )` # levert 243 op

- Bepaal de som van getallen in de lijst:

`sum( numlist )` # levert 257 op

- Bepaal de lengte van de lijst:

`len ( numlist )` # levert 4 op

- Controleer of getal 999 in de lijst zit

`999 in numlist` # levert False op

- Tel hoe vaak getal 1 voorkomt

`numlist.count( 1 )` # levert 2 op

- Lijst sorteren

`numlist.sort()` # sorteert lijst: [1,1,12,243]





# Onderzoekend leren

---

- Werkblad Lijsten



# Lijst doorlopen met een for loop

---

## Opdracht:

Druk alle getallen in mijn lijst af: **favNumbers = [ -4, 0, 11, 42 ]**

## Code:

```
favNumbers = [ -4, 0, 11, 42]
```

```
for getal in favNumbers:  
    print( getal )
```



# Lijst doorlopen met for en while loop

---

```
favNumbers = [ -4, 0, 11, 42]
```

```
for getal in favNumbers:  
    print( getal )
```

```
favNumbers = [ -4, 0, 11, 42]
```

```
index = 0  
while index < len (favNumbers):  
    print( favNumbers[index] )  
    index += 1
```



# Lijst aanpassen met een loop

---

- Een loop gebruiken om dingen in een lijst aan te passen?
- **Gouden regel: gebruik een while loop**

Dus een **for**-loop is lekker makkelijk te schrijven en gebruiken, maar waarom een **while**?

- Praktisch: je hebt niet de index (positie)
- Veilig:
  - De for-loop berekent eerst hoe vaak hij iets moet gaan uitvoeren: lengte van de lijst -1
  - als je in een for-loop dan iets toevoegt/verwijdert aan de lijst, dan verandert de lengte.. Probleem!



# Lijst doorlopen en aanpassen

---

Stel je wilt je cijferLijst opleuken:

```
cijferLijst = [8.0, 5.5, 2.3, 4.6, 6.1, 5.6]
```

... elk cijfer een puntje hoger.

*Tip. Afdrukken ging zo:*

```
favNumbers = [ -4, 0, 11, 42]  
index = 0  
while index < len(favNumbers):  
    print( favNumbers[index] )  
    index += 1
```

## Strategie

Lijst aanpassen? Gebruik een while

*Doorloop de cijferLijst met een index (van 0 tot lengte)*

*Pak het cijfer in cijferLijst en tel daar 1 bij op*

*Print de lijst*



# Lijst doorlopen en aanpassen

---

## Strategie

*Lijst aanpassen? Gebruik een **while**!*

*Doorloop de cijferLijst met een index (van 0 tot lengte)*

*Pak het cijfer in cijferLijst en tel daar 1 bij op*

*Print de lijst*

Een oplossing:

```
cijferLijst = [8.0, 5.5, 2.3, 4.6, 6.1, 5.6]
```

```
while                                :
```