

Samengestelde variabelen - Sets

Een set is een mutable dataverzameling van unieke elementen.

Set variabelen aanmaken en afdrukken

```
In [86]: spaceCrafts = set()
         counter = 0
         spaceCrafts = {"Pioneer", "Voyager"}
         print(spaceCrafts)
```

```
{'Pioneer', 'Voyager'}
```

Elementen kunnen aan een bestaande set worden toegevoegd..

```
In [87]: spaceCrafts.add("Voyager")
```

maar elk element komt slechts 1 keer voor

```
In [88]: print(spaceCrafts)

{'Pioneer', 'Voyager'}
```

Exercise 1

Gegeven

```
In [89]: fibonacciList = [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377]
         print(fibonacciList)
```

```
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377]
```

Gevraagd

Maak gebruik van casting om variabele *fibonacciList* te ontdebellen

Oplossing

```
In [90]: fibonacciList = set([0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377])
         print(fibonacciList)
```

```
{0, 1, 2, 3, 34, 5, 8, 233, 377, 13, 144, 21, 55, 89}
```

De lijst is ontdebeld maar het type van de variabele *fibonacciList* is veranderd

Operaties

```
In [91]: nordics = {"Denmark", "Finland", "Iceland", "Norway"}
baltics = {"Estonia", "Latvia", "Lithuania"}
eu = {"Austria", "Belgium", "Bulgaria", "Croatia", "Cyprus", "Czechia", "Der
```

Exercise 2

Gegeven

```
In [92]: benelux = {"Belgium", "The Netherlands", "Luxembourg"}
```

Gevraagd

Maak een String met als value "BeNeLux" die is opgebouwd uit de letters van corresponderende items uit de set *benelux* zonder de set variabele zelf aan te passen

Hints:

1. Maak gebruik (tijdelijke) variable van het type list
2. Pas daarin het item "The Netherlands zodat het consistent is met de andere items

Oplossing

```
In [119]: benelux2 = {"Belgium", "The Netherlands", "Luxembourg"}

s = str(benelux2)
s2 = s.replace('The ', '')

print (s2)

print (s3[29:31],s2[2:4], s2[17:20])

{'Netherlands', 'Luxembourg', 'Belgium'}
Be Ne Lux
```

De methode is correct. Het resultaat bevat helaas spaties

Visualisatie

Open een (git)bash terminal en run

```
pip install matplotlib-venn
```

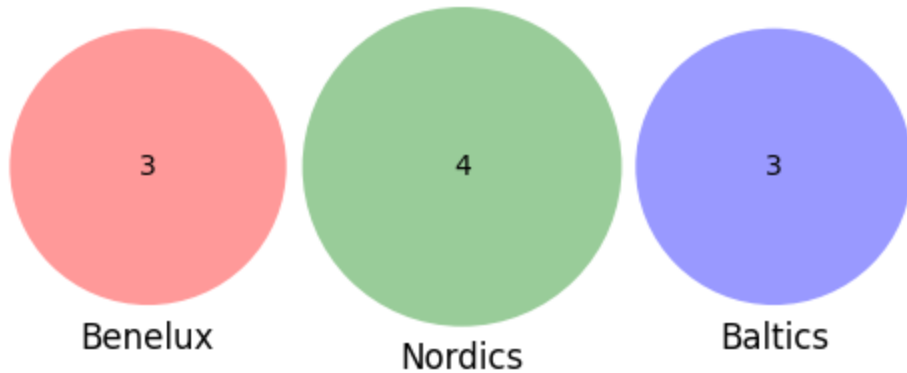
Vervolgens importeren we de modules *venn2*, *venn3* en *pyplot*

```
In [94]: from matplotlib_venn import venn2, venn3
from matplotlib import pyplot as plt
```

Gebruik een Venn diagram om Sets en hun relaties te visualiseren

De Sets benelux, nordics en baltics zijn **disjunct**..

```
In [95]: venn3([benelux, nordics, baltics], ('Benelux', 'Nordics', 'Baltics'))  
plt.show()
```



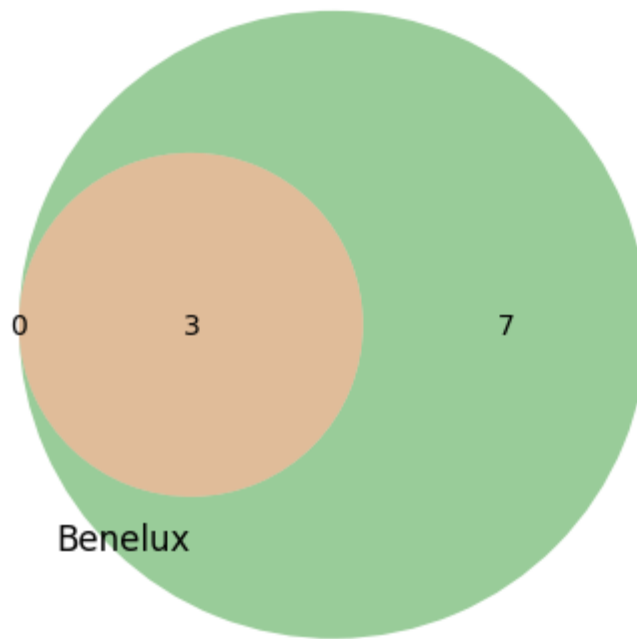
dat wil zeggen dat ze geen enkel item met elkaar gemeen hebben

```
In [96]: benelux.symmetric_difference(nordics).symmetric_difference(baltics)
```

```
Out[96]: {'Belgium',  
          'Denmark',  
          'Estonia',  
          'Finland',  
          'Iceland',  
          'Latvia',  
          'Lithuania',  
          'Luxembourg',  
          'Norway',  
          'The Netherlands'}
```

Gebruik *union* om verzamelingen verenigen

```
In [97]: subUnion = benelux.union(nordics).union(baltics)  
venn2([benelux, subUnion], ('Benelux', 'Benelux, Nordics and Baltics'))  
plt.show()
```



Benelux, Nordics and Baltics

De Sets benelux, nordics en baltics zijn allen een eigen **subset** van eu

Niet alle landen uit *subUnion* zijn lid van de Europese Unie

```
In [98]: baltics.issubset(eu) and benelux.issubset(eu) and nordics.issubset(eu)
```

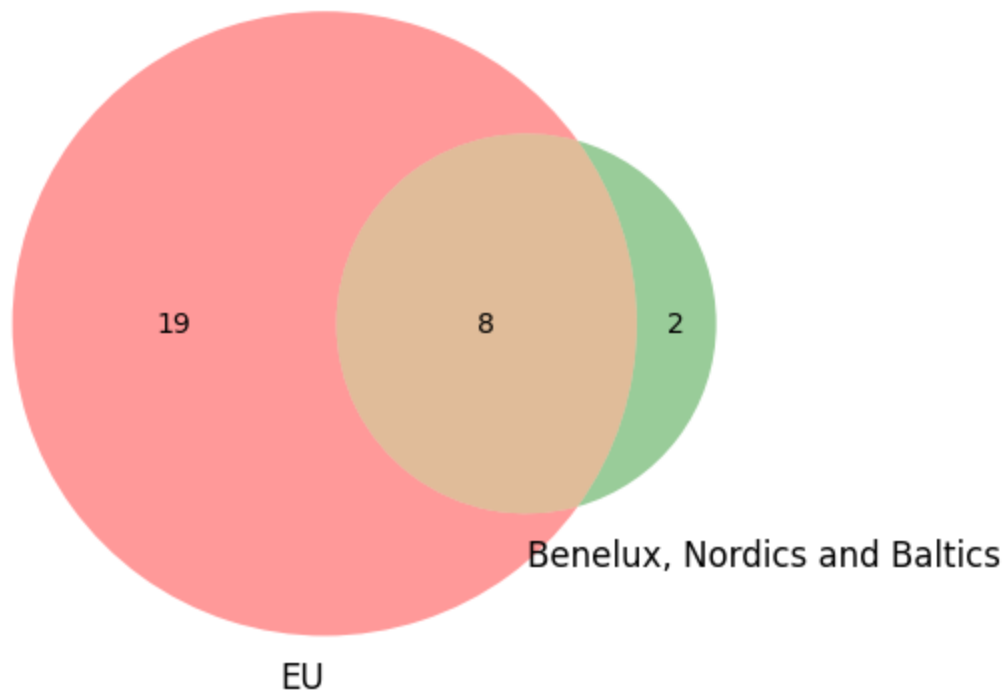
```
Out[98]: False
```

Gebruik de methode *intersection* voor de doorsnede van twee Sets

```
In [99]: eu.intersection(nordics)
```

```
Out[99]: {'Denmark', 'Finland'}
```

```
In [100... venn2([eu, subUnion], ('EU', 'Benelux, Nordics and Baltics'))  
plt.show()
```



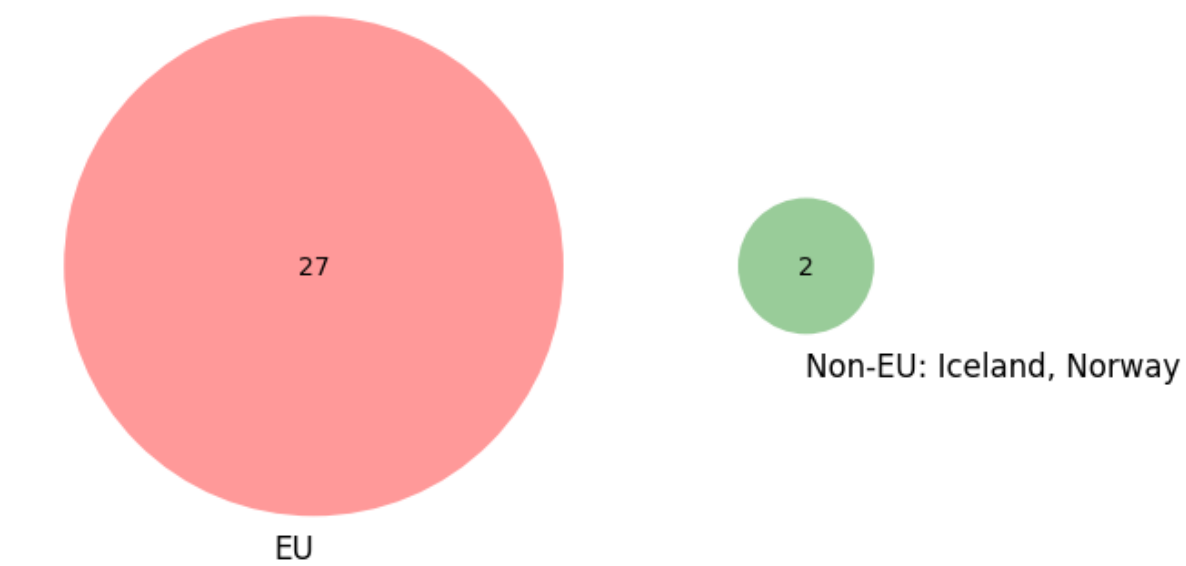
Exercise 3

Gebruik de methode `difference` om de landen weer te geven die (nog) geen lid zijn van de EU

```
In [101]: nordics.difference(eu)
```

```
Out[101]: {'Iceland', 'Norway'}
```

```
In [102]: difference = nordics.difference(eu)
          venn2([eu, difference], ('EU', 'Non-EU: Iceland, Norway'))
          plt.show()
```



Correct