

Samengestelde datatypes

In de vorige notebooks kwamen we reeds enkele eenvoudige datatypes tegen:

- bool
- int
- float
- string

en ook volgende containertypes:

- list
- set
- dict

De containertypes kunnen elk ander type bevatten, waaronder opnieuw een containertype. Op die manier kunnen we complexe datastructuren bouwen. We noemen dit *samengestelde datatypes*.

Stel dat we een matrix van getallen willen voorstellen, bijvoorbeeld:

$$\begin{bmatrix} 1 & 2 & 1 \\ 3 & 0 & 1 \\ 0 & 2 & 4 \end{bmatrix}$$

We zouden dit kunnen doen door de rijen van de matrix achter elkaar in een lijst te stoppen en een functie te schrijven die het element op rij i en kolom j teruggeeft:

```
In [1]: def elementOp(L,i,j):  
        return L[(i-1)*3+(j-1)]  
  
        L=[1,2,1,3,0,1,0,2,4]  
        print(elementOp(L,1,2))  
        print(elementOp(L,3,3))
```

```
2  
4
```

We kunnen er ook voor kiezen om de interne representatie (data structuur in het geheugen) dichter te laten aanleunen bij het object uit de "echte wereld" dat we willen voorstellen (de matrix). Bijvoorbeeld door de rijen van de matrix voor te stellen als lijsten, en de matrix zelf als een lijst van lijsten:

```
In [2]: rij1=[1,2,1]
rij2=[3,0,1]
rij3=[0,2,4]
M=[rij1,rij2,rij3]

# element op rij i en kolom j selecteren:
print(M[0][1]) # rij 1, kolom 2
print(M[2][2]) # rij 3, kolom 3
```

```
2
4
```

Merk op dat in de code `M[0][1]` gezien wordt als `(M[0])[1]`; dat is: operator `[1]` wordt toegepast op het resultaat van `M[0]`. We kunnen onze datastructuur ook directer schrijven als volgt:

```
In [3]: M=[[1,2,1],[3,0,1],[0,2,4]]
```

Deze manier om een matrix voor te stellen heeft bijkomend voordeel dat we de dimensies van de matrix kunnen afleiden uit de datastructuur; `len(M)` is het aantal rijen en bijvoorbeeld `len(M[0])` geeft het aantal kolommen. Volgende functie met parameter n geeft een eenheidmatrix van dimensie $n \times n$ terug:

```
In [4]: def unitMatrix(n):
    result=[]
    for i in range(n):
        rij=[0]*n
        rij[i]=1
        result.append(rij)
    return result
```

```
print(unitMatrix(5))
```

```
[[1, 0, 0, 0, 0], [0, 1, 0, 0, 0], [0, 0, 1, 0, 0], [0, 0, 0, 1, 0], [0, 0,
0, 0, 1]]
```

We kunnen de matrix ook mooi afdrukken:

```
In [5]: def printMatrix(M):
        for rij in M:
            for entry in rij:
                print(entry,end="\t") # Druk de entries af gescheiden door een tab-karakter om mooi uit te lijnen
            print()

        M=unitMatrix(10)
        printMatrix(M)
```

1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	1

Woordslang

Het tweede voorbeeld dat we behandelen is "woordslang". Bij dit typische auto-spel begint een van de spelers met de naam van een dier te noemen. De volgende speler moet dan een nieuw dier opnoemen waarvan de naam begint met de laatste letter van de naam van het vorige dier. Dit gaat dan door tot een van de spelers de naam van een dier een tweede maal zegt, een niet-bestaande dierennaam zegt, of niet tijdig met een dierennaam op de proppen kan komen.

We gaan dit spel nu in Python simuleren. We beginnen eerst met het inlezen van een lijst dierenamen uit een bestand. Dat kan in Python eenvoudig op volgende wijze:

```
In [6]: f=open("dier.txt","r") # open file "dier.txt" om te lezen (*r*ead)
        # f is het geopende bestand

n=5
for line in f: # itereer over de lijnen in f; Lees ze een voor een
    cleanLine=line.strip() # Haal overtollige spaties en \n weg aan begin en e
    inde van de ingelezen lijn
    print(cleanLine)
    n-=1      # We gaan niet het hele bestand afprinten, maar enkel de eerste
    n lijnen
    if n==0: # We verlagen n telkens en als n == 0 stoppen we
        break # break gaat de for-loop onmiddellijk stoppen

f.close() # sluit het bestand terug; anders gaan we volgende keer gewoon verde
r lezen waar we waren

aap
panda
rat
bosuil
kat
```

Bij woordslang moeten we onthouden welke woorden reeds gespeeld werden en of het gezegde woord een bestaand woord is. Dit kunnen we doen door de bestaande woorden en de gespeelde woorden bij te houden in een set.

```

In [11]: f=open("dier.txt","r")
dieren=set()
for line in f:
    cleanLine=line.strip()
    dieren.add(cleanLine) # voeg het dier toe aan de set dieren
f.close()

spelerAanBeurt=1
fout=False
gespeeld=set()
vorige=""
while not fout:
    s=input("Speler "+str(spelerAanBeurt)+" moet een dier noemen: ")
    if vorige!="" and vorige[-1]!=s[0]:
        print("Speler "+str(spelerAanBeurt)+" verliest want de beginletter is fout!")
        fout=True
    elif not s in dieren:
        print("Speler "+str(spelerAanBeurt)+" verliest want "+s+" bestaat niet.")
        fout=True
    elif s in gespeeld:
        fout=True
        print("Speler "+str(spelerAanBeurt)+" verliest want "+s+" werd reeds gespeeld.")
    else:
        gespeeld.add(s)
        if spelerAanBeurt==1:
            spelerAanBeurt=2
        else:
            spelerAanBeurt=1
        vorige=s

print("Speler "+str(spelerAanBeurt)+" wint het spel!")

```

Speler 1 verliest want aap werd reeds gespeeld.
 Speler 2 wint het spel!

Stel dat we nu een computerspeler willen maken. Dan is het belangrijk om te weten met welke letter een woord begint, of nog beter, voor elke beginletter een lijstje met dieren te hebben die met die letter beginnen. Zulk een lijstje kunnen we met een samengesteld datatype maken:

```
In [12]: dierenMet={} # nieuwe dictionary
for dier in dieren:
    startLetter=dier[0]
    if startLetter not in dierenMet:
        dierenMet[startLetter]=set()
    dierenMet[startLetter].add(dier)

print(dierenMet["a"])
```

```
{'alpaca', 'aziatische civetkat', 'alk', 'amerikaanse lepelsteur', 'alpensnee  
uwhoen', 'amerikaanse bever', 'amerikaanse das', 'asgrauwe keverslak', 'ameri  
kaanse rivierkreeft', 'asthenosoma varium', 'amerikaanse sneeuwhaas', 'angwan  
tibo', 'afrikaanse bos-olifant', 'aziatische goudkat', 'aasgier', 'axishert',  
'afrikaanse eieretende slang', 'antilopegrondeekhoorn', 'aziatische steenpatr  
ijs', 'afrikaanse civetkat', 'atlantische hagedisvis', 'antarctische zeebee  
r', 'australische zeebeer', 'amerikaanse bizon', 'afrikaanse jacana', 'arapai  
ma', 'aardwolf', 'aap', 'amazonedolfijn', 'atlantische vlekdoolfijn', 'axolot  
l', 'amerikaanse marter', 'apollovlinder', 'aardvarken', 'amerikaanse nerts',  
'anaconda', 'atlantische sousa', 'alpenmarmot', 'afrikaans boomschubdier', 'a  
frikaanse zeearend', 'alligatorschildpad', 'argalischaaap', 'afrikaanse longvi  
s', 'amerikaanse zeearend', 'afrikaanse olifant', 'arowana', 'alaskagrondeekh  
oorn', 'afrikaanse wilde kat', 'amerikaanse oehoe', 'agoeti', 'adder', 'afrik  
aanse buffel', 'aziatische wilde ezel', 'australische zeeleeuw', 'aziatische  
olifant', 'afrikaanse steppenvaraan', 'afrikaanse goudkat', 'amazone-lamantij  
n', 'aalscholver', 'alpenlandsalamander'}
```

De computer zal dus steeds een woord uit de juiste lijst halen en dat woord spelen. Verder wordt elk gespeeld woord uit de juiste lijst weggehaald:

```

In [14]: f=open("dier.txt","r")
dieren=set()
for line in f:
    cleanLine=line.strip()
    dieren.add(cleanLine) # voeg het dier toe aan de set dieren
f.close()

dierenMet={} # nieuwe dictionary
for dier in dieren:
    startLetter=dier[0]
    if startLetter not in dierenMet:
        dierenMet[startLetter]=set()
    dierenMet[startLetter].add(dier)

spelerAanBeurt=1
fout=False
gespeeld=set()
vorige=""
while not fout:
    if spelerAanBeurt==1:
        s=input("Speler "+str(spelerAanBeurt)+" moet een dier noemen: ")
    else:
        # computerspeler
        dierenMet[vorige[0]].remove(vorige)
        laatsteLetter=vorige[-1]
        if len(dierenMet[laatsteLetter])==0:
            # Computer kan niets meer spelen
            fout=True
            print("Computer weet geen dier meer dat begint met",laatsteletter)
        else:
            s=dierenMet[laatsteLetter].pop() # pop haalt een willekeurig element uit een lijst weg
            print("Computer speelt :",s)
        if vorige!="" and vorige[-1]!=s[0]:
            print("Speler "+str(spelerAanBeurt)+" verliest want de beginletter is fout!")
            fout=True
        elif not s in dieren:
            print("Speler "+str(spelerAanBeurt)+" verliest want "+s+" bestaat niet.")
            fout=True
        elif s in gespeeld:
            fout=True
            print("Speler "+str(spelerAanBeurt)+" verliest want "+s+" werd reeds gespeeld.")
        else:
            gespeeld.add(s)
        if spelerAanBeurt==1:
            spelerAanBeurt=2
        else:
            spelerAanBeurt=1
        vorige=s

print("Speler "+str(spelerAanBeurt)+" wint het spel!")

```

Computer speelt : pantserkrokodil

Speler 1 verliest want latijnse brilaap bestaat niet.
Speler 2 wint het spel!