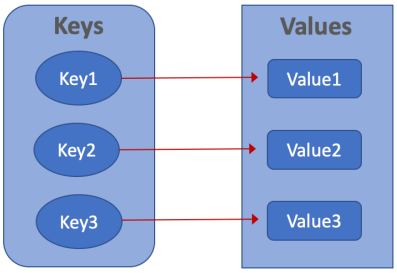
1. Dictionaries of woordenboeken

Een dictionary (data type: dict) is een verzameling van objecten die bewaard worden a.h.v. een sleutel (key). Iedere item van een dictionary bestaat uit een unieke key en een bijhorende waarde; waarbij deze waarde praktisch ieder Python object kan zijn.

Een belangrijk verschil met lijsten is dat voor dictionaries de volgorde van de items van geen belang is.

1.1. Syntax van een dictionary

Daar we elke key verbinden met een waarde creëren we een mapping-structuur bij het definiëren van een dictionary.

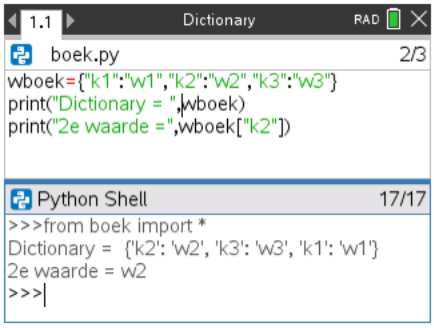


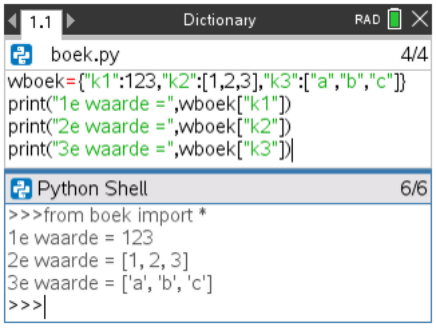
De syntax van een dictionary ziet er als volgt uit:

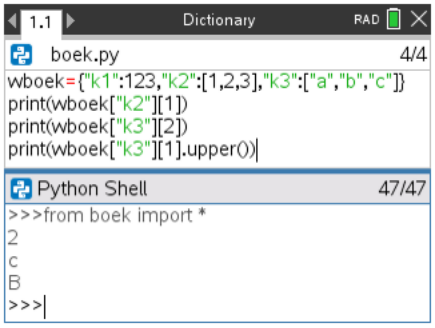
- De items van een dictionary staan tussen accolades { }
- Key en waarde worden gescheiden door een dubbele punt :
- De key staat altijd tussen aanhalingstekens “ “ (of ‘ ‘)

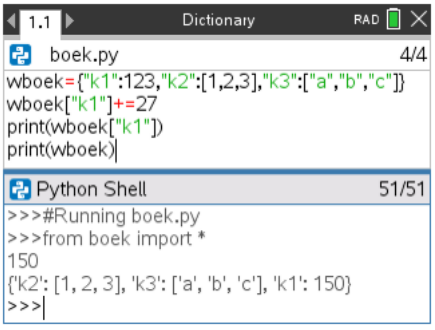
wboek={“key1”:”value1”, “key2”:”value2”, “key3”:”value3”}

We gebruiken de keys van een dictionary om een waarde aan te roepen, b.v. wboek[“key2”].



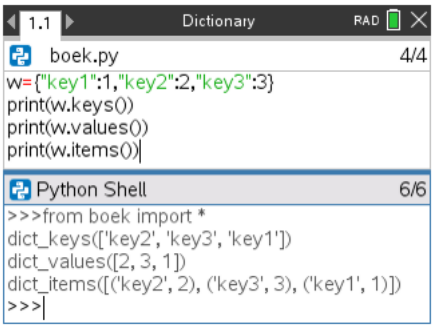


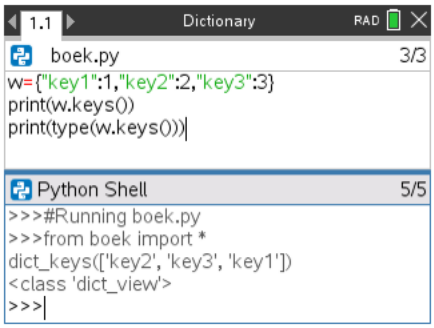




1.2. Enkele methodes voor dictionaries

Hieronder de methodes om de lijst met keys, de lijst met waarden en de lijst met items te bekijken.





1.3. Opdracht 1 – Schaar-Papier-Steen

Bij het spelletje *Schaar-papier-steen* is de uitslag onbeslist indien beide spelers dezelfde keuze maken. In het andere geval wint schaar van papier, papier van steen en steen van schaar.

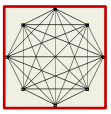
Hieronder een programma dat het spelen tegen de computer simuleert:

```
from random import *

keuzes=["schaar","papier","steen"]

k=int(input("1=schaar, 2=papier, 3=steen: "))
speler=k-1
comp=randint(0,2)

if speler==comp:
    print("Onbeslist, beide "+keuzes[speler])
else:
    if speler==0 and comp==1:
        print("jij wint met",keuzes[speler],"tegen",keuzes[comp])
    elif speler==1 and comp==2:
        print("jij wint met",keuzes[speler]+"tegen",keuzes[comp])
    elif speler==2 and comp==0:
        print("jij wint met",keuzes[speler],"tegen",keuzes[comp])
    else:
        print("jij verliest met",keuzes[speler],"tegen",keuzes[comp])
```

Pas bovenstaand programma aan om gebruikmakend van een dictionary met de spelregels voor winst, de voorwaardelijke statements wat eleganter te coderen, b.v.

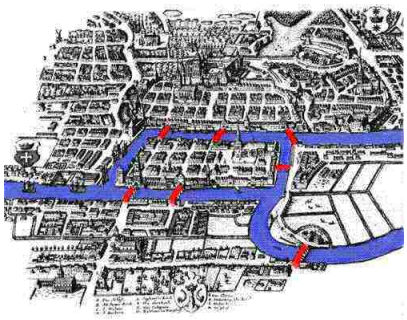
```
from random import *

keuzes=["schaar","papier","steen"]
regels={"schaar":"papier","papier":"steen",”steen":"schaar"}
.....
```

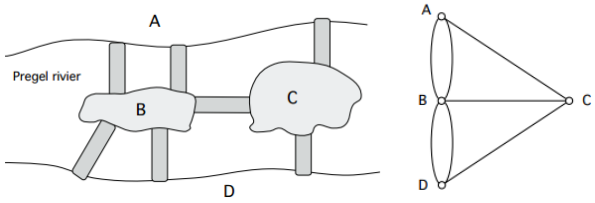
1.4. Opdracht 2 – Grafen



Een graaf is een verzameling van punten, knopen, waarvan sommige knopen verbonden zijn met lijnen, de zijden van de graaf. Grafen worden o.a. in de informatica gebruikt om het dataverkeer over netwerken weer te geven en te analyseren.

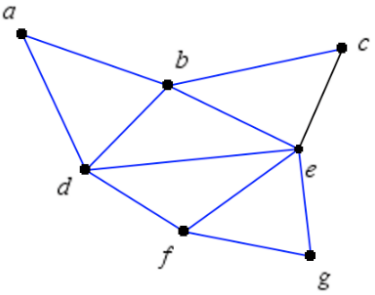
Eén van de eerste grafen-problemen is het probleem van de zeven bruggen van Koningsbergen: *Is het mogelijk een wandeling te organiseren zodat je precies één keer over iedere brug wandelt*. In 1736 loste Leonard Euler dit probleem op.





Euler bewees dat zo'n wandeling, een Euler pad, alleen mogelijk is indien de graaf geen of exact twee oneven knopen heeft. Een knoop is oneven als er een oneven aantal zijden samenkomen.

Schrijf een programma dat, gebruikmakend van een dictionary die als keys de knooppunten van de hiernaast afgebeelde graaf heeft, bij input van een knoop x als output een lijst van knopen geeft waarmee x verbonden is d.m.v. een zijde.



2. Sets of verzamelingen

Het data type set is een ongeordende collectie van unieke elementen; m.a.w. wiskundig gezien een verzameling. We laten het aan de lezer over te experimenteren met sets en de methodes beschikbaar voor sets.