Innebygde funksjoner i dictionaries

Læringsmål:

- Dictionaries
- Innebygde funksjoner

Starting Out with Python:

- Kap. 9.1

Info om innebygde innebygde funksjoner

Det kan være lurt å lese gjennom dette før du går videre

Akkurat som at det eksisterer mange innebygde funksjoner for lister, eksisterer det også mange innebygde funksjoner for dictionaries. Fra øving 6 husker du kanskje at du ble introdusert for len() som returnerer lengden til en liste, dvs. antall elementer i listen. Denne funksjonen kan også benyttes på dictionaries. Kjør kodeblokken under og se hva som skjer.

In [ ]:

```
dictionary = {}
dictionary['red'] = 'primærfarge'
num_items = len(dictionary)
print(num_items)
dictionary['blue'] = 'primærfarge'
dictionary['green'] = 'sekunærfarge'
num_items2 = len(dictionary)
print(num_items2)
```

Dersom du prøver å få tak i en verdi fra en nøkkel som ikke er lagt inn, vil du få en feilmelding. For å hindre dette kan det være lurt å sjekke om nøkkelen er i dictionaryen.

In [ ]:

```
if 'yellow' in dictionary:
    print(dictionary['yellow'])
```

Som du ser skjer det ingenting når man kjører koden over. Dette er fordi 'yellow' ikke finnes i dictionaryen. Hvis du vil kan du prøve å fjerne if-setningen og se hva som da skjer.

Man kan også fjerne elementer ganske greit i en dictionary, ved bruk av del og pop() som under:

In [ ]:

```
if 'green' in dictionary:
    del dictionary['green']  # green: sekundærfarge blir nå slettet fra dictionaryen
print(dictionary.pop('blue', 'Entry not found'))  # blue: primærfarge blir nå slettet f
ra dictionaryen og primærfarge returneres.
```

'Entry not found' er her en default-verdi som blir returnert dersom nøkkelen 'blue' ikke finnes i dictionaryen. Hvis du kjører kodeblokken over to ganger vil du se at 'Entry not found' blir printet den andre gangen.

Dersom det er ønskelig å tømme hele dictionaryen kan en bruke `dictionary.clear()` .

Ettersom en dictionary består av elementer som inneholder både en nøkkel og en verdi, er det litt mer avansert å iterere gjennom en dictionary enn en liste.

La oss tenke oss at vi har følgende dictionary:

In [4]:

```
my_family={'bror': 'Martin', 'søster': 'Helene', 'mor': 'Anne', 'far': 'Bob Bernt', 'hu
nd': 'Lovise'}
```

Dersom du ønsker å iterere gjennom alle nøklene i dictionaryen, dvs. 'bror', 'søster', 'mor', 'far' og 'hund' kan du skrive for-løkken som under. **Kjør bare koden under dersom du har kjørt koden over først.**

In [5]:

```
for key in my_family:
    print(key)
```

```
bror
søster
mor
far
hund
```

Dersom du ønsker å skrive ut alle nøklene samt navnene til familiemedlemmene kan du skrive:

In [6]:

```
for key in my_family:
    print(key, my_family[key])
```

```
bror Martin
søster Helene
mor Anne
far Bob Bernt
hund Lovise
```

En annen måte å få denne utskriften på, altså en iterering som itererer gjennom både verdier og nøkler samtidig, er ved å bruke følgende format:

In [7]:

```
for key, value in my_family.items():
    print(key, value)
```

```
bror Martin
søster Helene
mor Anne
far Bob Bernt
hund Lovise
```

`items()` returnerer alle nøklene i en dictionary samt deres tilhørende verdier.

Dersom det hadde vært ønskelig å kun printe ut verdiene, dvs. navnene, kunne man benyttet seg av `values()` :

In [8]:

```
for val in my_family.values():
    print(val)
```

```
Martin
Helene
Anne
Bob Bernt
Lovise
```

a)

Gitt følgende dictionary:

```
scores = {'Amanda': [88, 92, 100], 'Kennet': [30, 45, 50], 'Einstein': [100,100,
100]}
```

1. hva skrives ut dersom vi skriver `print(scores['Amanda'])`
2. hva skrives ut dersom vi skriver `print(scores['Amanda'][2])`

1. [88, 92, 100]
2. 100

b)

Opprett en tom dictionary som kalles fruit, for så å legge til tre frukter du liker som nøkler, og antall frukt av denne typen du pleier å spise hver dag som verdi.

Eksempel

```
fruit = {'epler': 2, 'pærer': 3, 'appelsiner': 1}
```

Skriv koden i kodeblokken under

In [10]:

```
fruit = {}
fruit['Mango'] = 9
fruit['Ananas'] = 2
fruit['Banan'] = 1
```

c)

Legg til to frukter som du misliker og fjern de tre fruktene du liker. Benytt deg av de innebygde funksjonene `del()` og `dict[key]=value` .

Skriv koden i kodeblokken under

In [11]:

```
fruit['Blodappelsin'] = 1
fruit['Epler'] = 100
del fruit['Mango']
del fruit['Ananas']
del fruit['Banan']
```

d)

Skriv ut begge verdiene (antall frukt du spiser hver dag som du misliker) i dictionaryen.

Skriv koden i kodeblokken under

In [12]:

```
for key in fruit:
    print(fruit[key])
```

```
1
100
```

Eksempel på kjøring dersom `fruit = {'bananer': 0, 'druer': 1}` :

```
0
1
```

e)

Sjekk om 'bananer' er i dictionaryen, og fjern 'bananer' fra dictionaryen dersom den er oppført.

Skriv koden din på angitt plass i kodeblokken under

In [13]:

```
fruit = {'bananer': 0, 'druer': 1}
print(fruit)
if 'bananer' in fruit:
    del fruit['bananer']
print(fruit)
```

```
{'bananer': 0, 'druer': 1}
{'druer': 1}
```

Hvis du har gjort alt riktig skal output fra koden over være:

```
{'bananer': 0, 'druer': 1}
{'druer': 1}
```

f)

Legg til et par bær som du liker i dictionaryen og skriv ut både verdiene og nøklene i dictionaryen på et greit format.

Skriv koden din i kodeblokken under

In [16]:

```
fruit['Jordbær'] = 10
fruit['Blåbær'] = 5000
for key in fruit:
    print(key+':',fruit[key])
```

```
druer: 1
Jordbær: 10
Blåbær: 5000
```

Eksempel på utskrift:

```
epler 2
pærer 0
jordbær 10
blåbær 50
```

Hint

Benytt deg av for key, value in fruit.items().