

INFO132 oppgavesett om tupler

(Noen oppgaver kan være basert på AI-generert tekst, og kvalitetssikret av en lærer)

Enkle oppgaver

1. Skriv en funksjon `lag_fortegnelse` (*personer*) som tar inn en liste av personer, representert som par med navn og alder. Funksjonen skal returnere en tilsvarende fortegnelse der hver person er representert med sitt navn som nøkkel og alderen som verdi.

```
>>>lag_fortegnelse([('Alice', 25), ('Bob', 30), ('Charlie', 35)])
{'Alice': 25, 'Bob': 30, 'Charlie': 35}
```

2. Lag en funksjon som gitt tre like lange tupler, returnerer tuppelet som består av triplene med 1. elementene fra tuplene, 2. elementene fra tuplene og så videre. Eksempel:

```
tall=(1,2,3,4,5)
storetall=(99,88,77,66,55)
bokstaver=('a','b','c','d','e')

>>> tripler(tall,storetall,bokstaver)
((1, 99, 'a'), (2, 88, 'b'), (3, 77, 'c'), (4, 66, 'd'), (5, 55, 'e'))
>>>
>>> tripler(bokstaver,tall,storetall)
(('a', 1, 99), ('b', 2, 88), ('c', 3, 77), ('d', 4, 66), ('e', 5, 55))
>>>
>>> tripler(tall,tall,tall)
((1, 1, 1), (2, 2, 2), (3, 3, 3), (4, 4, 4), (5, 5, 5))
```

3. Tenk deg en konkurranse der deltakerne oppnår poengsummer i hver runde. Lag en funksjon `poengSnitt` (*deltakerListe*) som tar inn en liste med deltakere. Hvert deltaker-element er et par med deltakerens navn og en liste med poengsummer som hen har oppnådd i konkurransen. For eksempel:

```
deltakere=[('Liv', [5,9,5]), ('Ola', [4,3,8]),
            ('Gro', [2,6,9]), ('Jon', [5,6,4])]
```

Lag en funksjon som returnerer en tilsvarende liste der parene inneholder navn og gjennomsnittet av deltakerens poengsummer.

```
>>>poengSnitt(deltakere)
[('Liv', 6.33), ('Ola', 5.0), ('Gro', 5.67), ('Jon', 5.0)]
```

Middels vanskelige oppgaver

4. Lag en funksjon som gitt en liste returnerer listen av alle par som kan dannes fra elementene i listen.

```
>>> alleKombinasjoner([1,2,3,4])
[(1, 1), (1, 2), (1, 3), (1, 4), (2, 1), (2, 2), (2, 3), (2, 4),
 (3, 1), (3, 2), (3, 3), (3, 4), (4, 1), (4, 2), (4, 3), (4, 4)]
```

5. Gjensyn med kobling av data (jfr. oppgave 7 i lab-oppgavene om fortegnelser)

Du har to lister av par, kalt A og B. Disse kan *kobles* til en ny liste av par C.
Spesifikt: hvis (x,y) er i A og (y,z) er i B så skal (x,z) være i C.

Lag en funksjon for å koble to lister av par. Eksempler:

```
dager=[(1,'man'),(2,'tir'),(3,'ons'),(4,'tor'),(5,'fre'),
        (6,'lør'),(7,'søn')]

temperaturer=[('man',11),('tir',9),('ons',7),('tor',12),
               ('fre',11),('lør',9),('søn',8)]
```

```
>>> koble(dager, temperaturer)
[(1, 11), (2, 9), (3, 7), (4, 12), (5, 11), (6, 9), (7, 8)]
>>> koble(temperaturer, dager)
[('ons', 'søn')]
>>>
```

6. Ta utgangspunkt i filer med temperaturdata som denne

tempFil.txt

```
Bergen man 7.0
Bergen ons 10.0
Bergen tir 9.0
Bodø man 2.0
Bodø ons 5.0
Bodø tir 5.0
Kristiansand man 16.0
Kristiansand ons 13.0
Kristiansand tir 15.0
Oslo man 8.0
Oslo ons 8.0
Oslo tir 5.0
Stavanger man 9.0
Stavanger ons 7.0
Stavanger tir 11.0
Tromsø man 3.0
Tromsø ons 2.0
Tromsø tir 5.0
Trondheim man 8.0
Trondheim ons 8.0
Trondheim tir 9.0
```

a)

- Lag en funksjon som kan lese inn temperaturdata fra en fil og lagre dem som tripler (3-tupler) i en global liste tempData.
- Lag en funksjon som lar en bruker taste inn temperaturdata
- lag en funksjon som skriver ut snitt-temperaturen for hvert sted
- lag en funksjon som skriver ut dataene i tempData på fil

```
>>> lesFil('tempFil.txt')
>>> tempData
[('Bergen', 'man', 7.0), ('Bergen', 'ons', 10.0), ('Bergen', 'tir', 9.0), ('Bodø', 'man', 2.0), ('Bodø', 'ons', 5.0), ('Bodø', 'tir', 5.0), ('Kristiansand', 'man', 16.0), ('Kristiansand', 'ons', 13.0), ('Kristiansand', 'tir', 15.0), ('Oslo', 'man', 8.0), ('Oslo', 'ons', 8.0), ('Oslo', 'tir', 5.0), ('Stavanger', 'man', 9.0), ('Stavanger', 'ons', 7.0), ('Stavanger', 'tir', 11.0), ('Tromsø', 'man', 3.0), ('Tromsø', 'ons', 2.0), ('Tromsø', 'tir', 5.0), ('Trondheim', 'man', 8.0), ('Trondheim', 'ons', 8.0), ('Trondheim', 'tir', 9.0)]
>>>
>>> settTemperatur()
Sted: Drammen
Dag: man
Temperatur: 9
>>>
>>> snittSteder()
Bergen 8.67
Bodø 4.0
Drammen 9.0
Kristiansand 14.67
Oslo 7.0
Stavanger 9.0
Tromsø 3.33
Trondheim 8.33
```

b) Som a) men temperaturdataene skal ligge i en global fortegnelse.

```
>>> lesFil('tempFil.txt')
>>> tempData
{('Stavanger', 'ons'): 7.0, ('Tromsø', 'man'): 3.0, ('Oslo',
'ons'): 8.0, ('Trondheim', 'man'): 8.0, ('Kristiansand', 'tir'):
15.0, ('Stavanger', 'man'): 9.0, ('Bergen', 'tir'): 9.0, ('Bodø',
'ons'): 5.0, ('Kristiansand', 'man'): 16.0, ('Trondheim', 'tir'):
9.0, ('Oslo', 'man'): 8.0, ('Tromsø', 'tir'): 5.0, ('Stavanger',
'tir'): 11.0, ('Bergen', 'ons'): 10.0, ('Oslo', 'tir'): 5.0,
('Kristiansand', 'ons'): 13.0, ('Bergen', 'man'): 7.0, ('Bodø',
'man'): 2.0, ('Bodø', 'tir'): 5.0, ('Trondheim', 'ons'): 8.0,
('Tromsø', 'ons'): 2.0}
>>> #osv
```

7. En ekipasje består av en hest og en rytter. Tenk deg startlister for ridekonkurranse der de to første elementene utgjør den første ekipasjen, de to neste er den 2. ekipasjen og så videre, dvs lister på formen

$$[\text{rytter}_1, \text{hest}_1, \text{rytter}_2, \text{hest}_2, \dots, \text{rytter}_N, \text{hest}_N]$$

Lag funksjonen `deltakere` (*startliste*) som returnerer et par med listene som består av hhv hesten og rytterne i startlisten,.

```
>>> startliste
['Kari', 'Blakken', 'Liv', 'Dølaprinsen', 'Eli', 'Sofus', 'Gro',
'Blåne']
>>> (hester, ryttere) = deltakere(startliste)
>>> hester
['Blakken', 'Dølaprinsen', 'Sofus', 'Blåne']
>>> ryttere
['Kari', 'Liv', 'Eli', 'Gro']
```

8.

- a) Lag en fortegnelse over fagbøker, der nøkkelen er kombinasjonen av forfatter og tittel som knyttes til et fagområde og utgivelsesår. Bruk en global variabel `Bøker` og lag en funksjon `skrivBøker()` som skriver ut bøkene på skjermen.
- b) Lag en funksjon `leggTilBok()` som lar brukeren taste inn data om en ny bok som legges inn i fortegnelsen.
- c) Lag en funksjon `finnForfatter(forfatternavn)` som skriver ut tittel og utgivelsesår for alle bøker av forfatteren
- d) Lag en funksjon `finnFagfelt(fagfelt)` som skriver ut forfatter, tittel og årstall for alle bøker innen fagfeltet

Eksempler:

```
>>>#a)
>>> skrivBøker()
Blackburn ,  Modal logic ,  logikk ,  2002
Brook ,  Knowledge and Mind ,  filosofi ,  2000
Dowek ,  Computation, proof, machine ,  matematikk ,  2015
Dowek ,  Proofs and algorithms ,  logikk ,  2011
Hein ,  Discrete mathematics ,  matematikk ,  2003
Horstmann ,  Python for everyone ,  programmering ,  2016
Lowe ,  A survey of metaphysics ,  filosofi ,  2002
Severance ,  Java for somebody ,  programmering ,  1999
Severance ,  Python for everybody ,  programmering ,  2016
>>>
>>>#b)
>>> leggTilBok()
Forfatter: Pettersen
Tittel: Python for alle
Fagfelt: programmering
Utgivelsesår: 2010
>>>
>>> skrivBøker()
Blackburn ,  Modal logic ,  logikk ,  2002
Brook ,  Knowledge and Mind ,  filosofi ,  2000
Dowek ,  Computation, proof, machine ,  matematikk ,  2015
Dowek ,  Proofs and algorithms ,  logikk ,  2011
Hein ,  Discrete mathematics ,  matematikk ,  2003
Horstmann ,  Python for everyone ,  programmering ,  2016
Lowe ,  A survey of metaphysics ,  filosofi ,  2002
Pettersen ,  Python for alle ,  programmering ,  2010
Severance ,  Java for somebody ,  programmering ,  1999
Severance ,  Python for everybody ,  programmering ,  2016
>>>
>>> #c)
>>> finnForfatter('Dowek')
Computation, proof, machine ,  2015
Proofs and algorithms ,  2011
>>>
>>> #d)
>>> finnFagfelt('filosofi')
Brook ,  Knowledge and Mind ,  2000
Lowe ,  A survey of metaphysics ,  2002
>>>
```

Vanskelige oppgaver

9. Skriv en funksjon `sorter_produkter` (*produkter*) som tar inn en liste med produkter i form av par, der hvert par inneholder produktnavn og pris. Eksempel:

```
produkter=[('Banan', 10), ('Eple', 8), ('Appelsin', 15),  
           ('Druer', 8), ('Pære', 10)]
```

Funksjonen skal sortere produktene basert på pris, fra lavest til høyest, og returnere den sorterte listen. Hvis to varer har samme pris, skal de sorteres etter navn.

```
>>>sorter_produkter(produkter)  
[('Druer', 8), ('Eple', 8), ('Banan', 10), ('Pære', 10),  
 ('Appelsin', 15)]
```

Du finner noen tips i den vedlagte filen `sorteringstips.pdf`

Ekstra oppgaver

10. (Enkel)

Skriv en funksjon `snittkarakter` (*karakterliste*) som tar inn en liste med studentenes resultater på en eksamen. Hver student er representert som et par med navn og tall-karakter. Funksjonen skal beregne gjennomsnittskarakteren og returnere denne.

```
>>> snittkarakter([('Liv', 4), ('Jon', 5), ('Gro', 3), ('Ole', 4)])  
4.0
```

11. (Middels)

Lag en funksjon som tar inn to tupler og returnerer et nytt tuppel som fletter de to tuplene. Det vil si at resultat-tupplet skal ha verdier som er hentet vekselvis fra de to argument-tuplene. Dersom det ene tupplet er lengre enn det andre, legges de resterende elementene til slutten.

```
>>> flett((1, 2, 3), (4, 5, 6))  
(1, 4, 2, 5, 3, 6)  
>>>  
>>> flett((1, 2, 3), ('a', 'b', 'c', 'd', 'e'))  
(1, 'a', 2, 'b', 3, 'c', 'd', 'e')
```