

Øvingsforelesning i Python (TDT4110)

Tema: Øving 10, rekursjon, sudoku,

Kristoffer Hagen

Oversikt

- Praktisk informasjon
- Gjennomgang av Øving 9
- Oppgaver for Øving 10

Oversikt

- Praktisk informasjon
- Gjennomgang av Øving 9
- Oppgaver for Øving 10

Generelt om dictionary

<https://www.ntnu.no/wiki/display/tdt4110/Generelt+om+dictionary>

Bursdagsdatabasen

<https://www.ntnu.no/wiki/display/tdt4110/Bursdagsdatabasen>

Ostevirus

<https://www.ntnu.no/wiki/display/tdt4110/Osteviruset>

Oversikt

- Praktisk informasjon
- Gjennomgang av Øving 9
- Oppgaver for Øving 10

Om oppgave vi gjør

- Forventer ikke at folk skal klare oppgavene i første omgang, men vil at dere skal prøve
- Blir lettere og mer konstruktivt å følge med på min løsning etter at dere har prøvt litt
- Kommer til å gå gjennom litt nye ting dersom det trengs i neste øving

Oppgave 1.1

Skriv en funksjon/metode som regner ut $n!$ Ved hjelp av rekursjon

```
def recursiveFac(n):
```

Hint: `return n * recursiveFac(n-1)`

fac(4):

return 4 * fac(3)

return 4 * 3 * fac(2)

return 4 * 3 * 2 * fac(1)

return 4 * 3 * 2 * 1

return 24

Oppgave 1.2

Lag en funksjon som regner ut summen av alle elementene i en liste som selv kan inneholde nye lister (osv.)

```
liste = [1,2,3,[4,3],5,6,[7,4,[2,3]],8,9,10]
```

Hint: `type(element) == type([])`

Oppgave 2: 4x4 Sudoku

3	4	1	
	2		
		2	
	1	4	3

Oppgave 2.1

- Lag en en fil sudoku.txt som inneholder:

4

3410

0200

0020

0143

Oppgave 2.2

Lag en funksjon `readFromFile(filename)`: som leser fra filen vi nettopp laget, og lagrer informasjonen i en 2d liste (matrise)

Resultatet skal bli:

```
boardInfo = [[3, 4, 1, 0], [0, 2, 0, 0], [0, 0, 2, 0], [0, 1, 4, 3]]
```

Oppgave 2.3

Lag en funksjon som printer brettet/sudokuen på en ryddig måte:

Eksempel:

```
      0  1  2  3
-----|
0 | 3 | 4 # 1 | 0 |
-----|
1 | 0 | 2 # 0 | 0 |
=====|
2 | 0 | 0 # 2 | 0 |
-----|
3 | 0 | 1 # 4 | 3 |
-----|
```

Oppgave 2.4

Lag en funksjon `getHorizontalList(x,y)` som returnerer en liste med alle tallene som finnes på samme linje som koordinaten i argumentet (parameteren)

Oppgave 2.5

Lag en funksjon `getVerticalList(x,y)` som returnerer en liste med alle tallene som finnes på samme rad som koordinaten i argumentet (parameteren)

Oppgave 2.6

Lag en funksjon `getSquareList(x,y)` som returnerer en liste med alle tallene som finnes i samme «rute» som koordinaten i argumentet (parameteren)

Oppgave 2.7

Lag en funksjon `getConflictNumbers(x,y)` som kombinerer de tre forrige funksjonene

`getHorizontalList(x,y)`

`getVerticalList(x,y)`

`getSquareList(x,y)`

for å finne alle tall som IKKE kan brukes i den gitte koordinaten

Oppgave 2.8

Tror aldri at me komme så langt som dette :D

Lag en funksjon `checkMove(number,x,y)`: som sjekker om tallet «number» kan plasseres i koordinat `x,y` uten å lage noen nye konflikter med andre tall.

Spørsmål?

- Send meg evt. spørsmål og tilbakemelding (relatert til øvingsforelesningene) på kristoffer.hagen@idi.ntnu.no