



Seminartime uke 4 – Gr. 10

IN1000 – HØST 2021

Hvor langt har vi kommet?

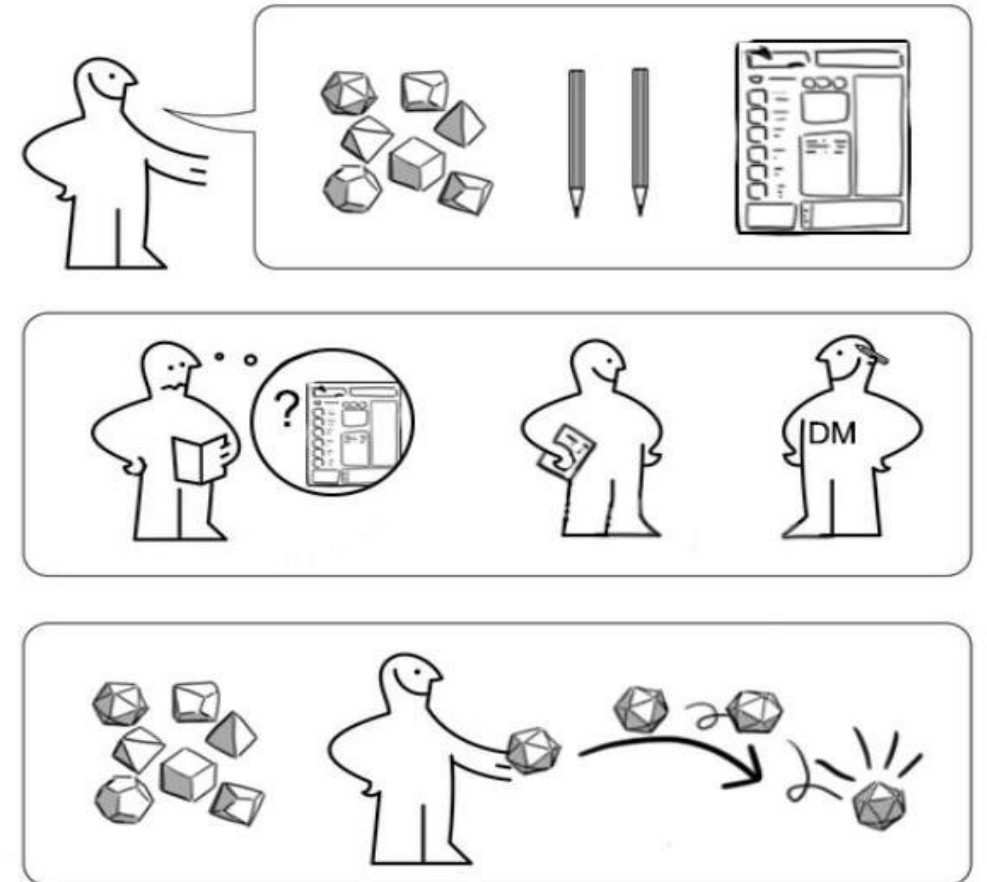
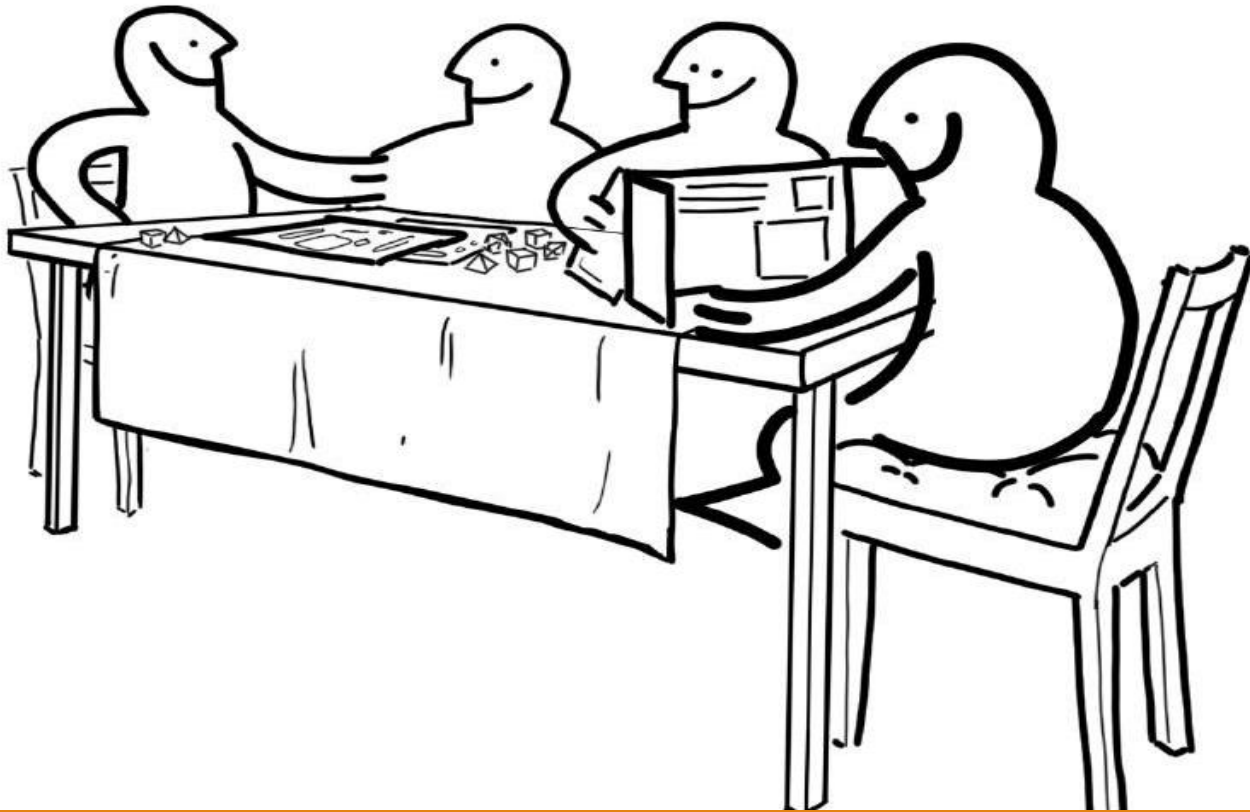
NB: Endringer kan komme	Legges ut ca.	Frist
1.innlevering (3 poeng)	25. august kl 13.00	31. august kl 23.59
2. innlevering (5 poeng):	1. september kl 13.00	7. september kl 23.59
3.innlevering (5 poeng):	08. september kl 13.00	14. september kl 23.59
4.innlevering (5 poeng):	15. september kl 13.00	21. september kl 23.59
5.innlevering (5 poeng):		
6. innlevering (6 poeng):		

Uke	Tema
4	Løkker <ul style="list-style-type: none">- For-løkker (for loops)- While-løkker (while loops)- Kombinere løkker med samlinger- Funksjoner & prosedyrer m. parametre+returverdi
5	Filer <ul style="list-style-type: none">- Lese fra fil- Skrive til fil- Parameter-overføring og variabel-skop- Kombinere data for å løse problemer
6	<ul style="list-style-type: none">- Repetisjon- Repetisjon- Repetisjon!

Plan for i dag

- Prosedyrer/funksjoner
- For-løkker
- While-løkker
- Scope
- Lag et program med lister!

GÅME NIGHT



Prosedyrer/funksjoner

Prosedyrer/funksjoner

Prosedyrer og funksjoner:

- Lagre kodeblokker for gjenbruk
- KAN ta imot argumenter/parametre for å bruke inne i funksjonen
- Fleksibel bruk av kode

Funksjoner:

- Retur-verdi

Prosedyrer:

- Ingen retur-verdi

```
27 def start(arg):
28     oppgaveliste.append(arg)
29     print("LABEL:", oppgaveliste)
30     print("START:", datetime.datetime.now().strftime("%c"))
31
32 def ferdig_oppgave():
33     fjern=oppgaveliste.pop(0)
34     print("AVSLUTTET:", datetime.datetime.now().strftime("%c"))
35     print("LABEL:",fjern)
36     if sjekk_lengde() == 0:
37         print("Du har ingen flere oppgaver ingke")
38
39 def sjekk_lengde():
40     return len(oppgaveliste)
```

Parametre/argumenter

argument1



argument2



Funksjon



Retur-verdi

- Velger selv hvor mange parametre man skal ha
- «Velger» selv datatypen man vil returnere (viktig for videre bruk!)
- Argumenter man sender inn MÅ svare til riktig rekkefølge på definerte parametre
- Kan ha mange datatyper som parameter, men kun 1 type returverdi
- Man kan gjøre hva som helst i en prosedyre/funksjon, til og med kalle på andre funksjoner

Syntaks:

```
def funkNavn(arg1, arg2, arg3,.....):  
  <kodeblokk>  
  return <verdi>
```

Hva printes ut?

```
def multipliser():  
    return 2*2  
  
print(multipliser())
```

→ 4

PS! Husk kodeflyt!

```
def multipliser2(arg):  
    return arg*arg  
  
print(multipliser2(multipliser2(2)))
```

→ 16



Løkker

While-loops, for-each-loop, for-in-range-loop

Hva er løkker?

- Løkker er noe som går rundt og rundt eller skjer igjen og igjen
- Kanskje den vanskeligste av grunnkonseptene innenfor programmering
- Krever god forståelse av kodeflyt og skop
- Det finnes tre typer løkker: while, for-in-range, og for-each

While-løkker	For-in-range-løkker	For-each-løkker
<pre>while <uttrykk>: <kode som kjøres så lenge uttrykk er true></pre>	<pre>for <variabel> in <range>: <kode som kjøres> <variabel er en teller></pre>	<pre>for <element> in <samling>: <kodesnutt> <element kan brukes></pre>
<pre>teller = 0 while teller<4: print(`Teller = `, teller) teller += 1</pre>	<pre>for i in range(3): print(`teller: `, i)</pre>	<pre>popcorn = [1, 2, 3, 4] for pop in popcorn: print(pop)</pre>

While-løkker

- Les: «så lenge *uttrykket* er sant»
- Alt som defineres i koden vil kjøre helt frem til uttrykket ikke lenger er sant
- Ønsker å kontrollere loopen med et uttrykk (og ikke True) for å hindre evig løkke!
 - Og du slipper å bruke break

Flere eksempler:

```
while (i < 100):  
    i += 10
```

```
while (bruker_input.isdigit() == False):  
    brukerinput = input("Skriv inn et gyldig tall: ")
```

```
while (int(bruker_input) < 0):  
    brukerinput = input("Skriv inn et positivt tall")
```

For-løkker

- Vi selv definerer hvor mange ganger løkken skal kjøre
- Mer kontrollert enn while-løkker
- Nyttig hvis man skal gjøre én ting flere ganger
- Antall ganger kan være lik lengden på en liste, innenfor en range, eller for alle elementer

Hva er forskjellen:

```
sjokolade = [``nidar``, ``freia``, ``mondelez``, ``hval``]  
for sjoko in sjokolade:  
    print(sjoko)
```

```
→ nidar  
   freia  
   mondelez  
   hval
```

```
sjokolade = [``nidar``, ``freia``, ``mondelez``, ``hval``]  
for i in range(len(sjokolade)):  
    print(i)
```

```
→ 0  
   1  
   2  
   3
```



Scope

Scope

- Veldig viktig konsept i programmering!
- Handler om hva som er tilgjengelig for alle vs for noen
- Hver gang du indenterer koden din (if/elif/else/def/for/while) lager du en boks
- Alle variablene som opprettes innenfor tilhører den boksen
- Alle variablene som opprettes globalt er tilgjengelig for ALT (bør unngå dette! Ha hovedprogram!)

Hva printes i terminalen:

```
global_liste = [1, 2, 3, 4, 5]
def inner():
    lokal_liste = [11, 22, 33, 44, 55]
    print("Global liste innenfra: ", global_liste)

print("Global liste utenfra: ", global_liste)
print("Lokal liste utenfra:", lokal_liste)
```

→ Global liste utenfra: [1, 2, 3, 4, 5]
<Feilmelding>
NameError: name 'lokal_liste' is not defined



KODING – LAG ET PROGRAM

Kodeoppgave!

- Lage et program som håndterer lister med tall (se demo)
- Hver gruppe får tildelt sin funksjon
- Send ferdigskrevet funksjon til annvp@uio.no
- Får 30 min til å skrive
- 30 min til felles gjennomgang