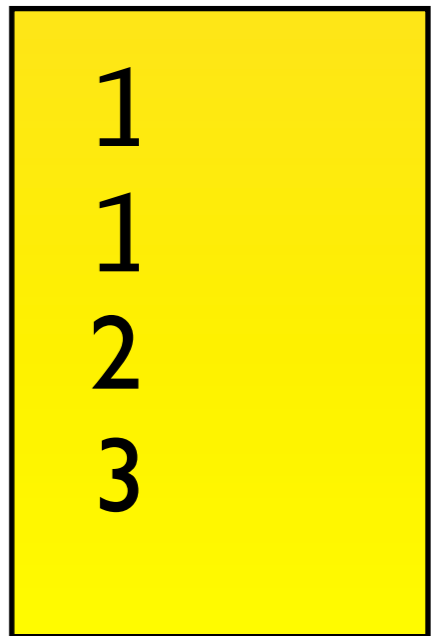


# Et lite oppdrag i bakgrunnen

- Under pultene på bakerste rad er det klistret post-it lapper med to tall skrevet på
- Regn ut summen av to nederste tall, skriv denne summen under de andre tallene, og send lappen til personen foran deg
- Alle som mottar en lapp (fra personen bak) gjør det samme som beskrevet over
- Jeg samler senere inn lappene fra fremste rad



# Hva er en “Computer”?

- "The idea behind digital computers may be explained by saying that these machines are intended to carry out any operations which could be done by a human computer" (Alan Turing, 1950)
- Tradisjonell definisjon av “computer”:  
En ansatt som gjør utregninger ved å blindt følge en liste instruksjoner med penn og papir, så lenge som det trengs, uten at det krever noen form for innsikt (fra 1600-tallet)
- Poeng for oss:  
Dette faget handler ikke om å få en printer til å virke (puh), men om å sette sammen presise oppskrifter!



# Dere har i mellomtiden regnet ut Fibonacci's tallrekke!

- 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89
- Merk at resultatet av denne prosedyren er mye mer avansert enn operasjonen i hvert steg (pluss to tall)

# Vi har programmert!

- Eller rettere sagt:
  - Dere har vært computere
  - Jeg har programmert dere
- Oppfordring:
  - Forsøk alltid å først løse et problem for hånd, ut fra presis sekvens av enkle operasjoner

# Hva er programmering?

- *"Software development happens in your head, not in an editor. Programming is all about problem solving. It requires creativity, ingenuity, and invention"*  
(Andy Hunt)
- ➡ **Programmering er utfordrende, men ikke minst kreativt og gøy!**
- *"Computer science education cannot make anybody an expert programmer any more than studying brushes and pigment can make somebody an expert painter"* (Eric Raymond)
- ➡ **Forelesningene kan hjelpe dere i gang, men det er gjennom mange timer med egen programmering dere virkelig lærer det**

# Et første Python-program

- Be datamaskinen om å:
  - Skrive "Hei INF1001" på skjermen

# Flere Python-program

- Be datamaskinen om å:
  - Skrive  $9+9$  på skjermen
  - Skrive hva  $9+9$  blir på skjermen
  - Skrive hva  $1+2+3+4$  blir på skjermen
  - Skrive på skjermen om det stemmer at 5 er større enn 4
  - Skrive på skjermen om det stemmer at 4 er større enn 5

# Noen formelle begrep

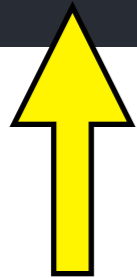
**Programsetning** (statement): *en linje i et Python-program*



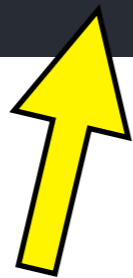
**Uttrykk** (expression): *noe som kan evalueres (beregnes) til en verdi*



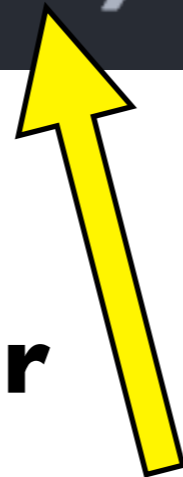
```
print(2+3)
```



**Biblioteks-  
funksjon**



**Operator**



**Literal**

(av typen tall)



# Datatyper

```
print("Hei")
```

Tekst (String) - str

```
print(5)
```

Heltall (Integer) - int

```
print(5.1)
```

Flyttall - float

```
print("5")
```

Tekst (pga hermetegn)

```
print('hei')
```

Enkle eller doble  
hermetegn

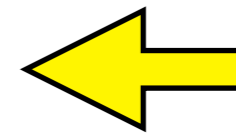
# Operasjoner avhenger av datatype

```
print(2+3)
```



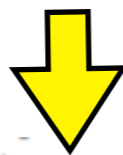
+ betyr addisjon

```
print("Hei" + "INF1001")
```



+ betyr  
konkatenerer  
tekst

```
print("Hei" + 1001)
```



```
guardian:kode_uke1 sandve$ python3 hei.py
```

```
Traceback (most recent call last):
```

```
File "hei.py", line 1, in <module>
```

```
    print("Hei" + 1001)
```

```
TypeError: Can't convert 'int' object to str implicitly
```

# Variabler

- En variabel er et bestemt navn som representerer en verdi
  - `tall = 13`
  - En (navngitt) parkeringsplass hvor man kan plassere en verdi
- Å sette verdien til en variabel (som i linjen over) kalles en tilordning (assignment)
  - En tilordning består av en variabel etterfulgt av et enkelt likhetstegn etterfulgt av et uttrykk (en verdi)

# Variabler

## Tilordning:

variabelnavn = uttrykk

```
tall = 5  
print(tall)
```

Tilordning av variabel:  
gir beskjed om at navnet tall  
skal representere verdien 5

Bruk av variabel i et uttrykk:  
henter frem verdien 5

# Variabler:

settes fra og brukes i uttrykk

## Tilordning:

variabelnavn = uttrykk

```
tall = 5+1  
print(tall+3)
```

Tilordning:  
navnet tall gis  
verdien av uttrykket (6)

Bruk av variabel i et uttrykk:

henter verdien 6, plusser med 3, og evaluerer dermed til 9

# Variabler: verdien kan endres

```
tall = 5  
tall = 8  
print(tall)
```

variabelen tall gis verdien 5

verdien til tall vil fra nå av  
(nedenfor) være 8

Bruk av variabel:  
henter verdien 8

# Variabler:

## venstresiden versus høyresiden

```
tall = 5  
tall = tall + 2
```

variabelen tall  
gis verdien 5

verdien til tall vil  
fra nå av være 7

Venstreside  
tilordner fremtidig  
fra og med neste  
**tall** ha verdien fra  
høyresiden (**7**)

Ikke "**er lik**" som i  
ligninger på ungdomsskolen,  
men "**gjør lik**" som i  
lignelser i bibelen

Høyreside  
uttrykk:  
" verdi -  
hadde før  
denne linjen (**5**)

# Variabler: regler for navn

- Noen krav og anbefalinger når det gjelder navn:
  - Kan inneholde både bokstaver og tall, men kan ikke begynne med et tall (tall5 er lovlig, men ikke 5tall)
  - Visse tegn, slik som mellomrom (space), er ikke lovlige å bruke i variabelnavn
  - Om et variabelnavn er bygd opp av flere ord, er det anbefalt å skille med understrek (underscore) - f.eks. `mitt_variabel_navn`
  - Merk at det som skiller en verdi av type tekst (f.eks. "Hei INF1001") fra et variabelnavn (f.eks. tall) er at tekstverdier har hermetegn rundt



# Variabler:

eksempel på å holde mellomverdier

- [Areal.py]

# Variabler:

eksempel på å bruke flere ganger

- [Epler.py]

# Kommentarer

```
#Disse to linjene  
#påvirker ingen ting  
print("Hei")
```

# Feilmeldinger

```
Print("hei")
```

```
guardian:kode sandve$ python3 livekode.py  
Traceback (most recent call last):  
  File "livekode.py", line 1, in <module>  
    Print("hei")  
NameError: name 'Print' is not defined
```

Hvor  
problemet er  
(linje 1)

Selve problemet

Hvilken type  
problem/feilmelding

# Sjekk antagelser

```
tall = 4+1 *2  
print(tall)
```

Kommer tallet 10  
på skjermen?

```
tall = 4+1 *2  
print(tall==10)
```

Kommer True  
på skjermen?

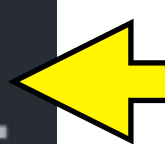
```
tall = 4+1 *2  
assert tall==10
```

Stopper programmet  
med en feilmelding?

# Sjekke antagelser

```
tall = 4+1 *2  
assert tall==10, tall
```

Dersom stopp:  
verdien av tall  
blir med i  
feilmeldingen..



```
guardian:kode sandve$ python3 livekode.py  
Traceback (most recent call last):  
  File "livekode.py", line 2, in <module>  
    assert tall==10, tall  
AssertionError: 6
```

**[Skriver og kjører et  
program]**

# Skrive et python-program

- Et python-program er bare en helt vanlig tekst-fil
- Man kan skrive koden med en rekke ulike program (editor)
- Det går an i word eller excel, men det er en veldig dårlig idé
- I INF1001 støtter vi primært en editor som heter Atom
- Atom håndterer innrykk bra (veldig viktig!), gir fargekoding, og er lett å installere
- Det finnes også nettsteder hvor man kan skrive (og kjøre) python-kode uten å måtte installere noe, men det er oftest mer praktisk med lokal editor og kjøring



# Kjøre et python-program

- Man kan sette opp kjøring gjennom å dobbelt-klikke i grafisk brukergrensesnitt
- I INF1001 legger vi opp til å i stedet kjøre via kommandolinjen - noe som i lengden er mer praktisk og fleksibelt
- Mange editorer har også mulighet for å kjøre python-filen man jobber på via menyen (og via tastatur-snarvei)

# Python-tolkeren (interpreter)

- I stedet for å skrive kode i en fil i en editor, og deretter kjøre filen, kan man direkte kjøre en og en linje i noe som kalles python-tolkeren.
  - Fra kommandolinjen skriver man "python3" for å starte tolkeren
  - Man kan da skrive og utføre en enkelt linje av gangen
  - Man kan også inspisere uttrykk ved å skrive uttrykket og få verdien det evaluerer til på linjen under
  - Miljøet (variabler) beholder sine verdier (om de ikke direkte endres)

# Python-tolkeren (interpreter) (forts)

- Tolkeren er veldig egnet for å detaljert forstå hva som skjer på en enkelt linje - f.eks. se verdier av variabler og uttrykk
- Når man skriver hele programmer (mange linjer) er det derimot mye mer praktisk å skrive og kjøre en fil (gir bedre oversikt og man kan endre en linje og kjøre hele programmet på nytt fra start)

# Eksamensoppgave fra fjorårets INF1000

## Oppgave 1

a) Hva er verdien til **tall** etter at følgende kode er utført?

```
int tall = 3+2+1;  
tall = tall*2;
```