

Plan for dagen

- Hvordan én enkelt linje utføres:
 - Datatyper
 - **Evaluering av uttrykk og funksjoner**
- Hvordan et helt program utføres:
 - Kodeflyt fra linje til linje
 - Prosedyrer
- Sjekke antagelser og tolke feilmeldinger

Hva skjer innad på en linje?

- $\text{alder} = 6$
 - veldig rett frem..
- $\text{alder} = \text{alder} + 3$

Hva skjer innad på en linje?

- $\text{alder} = 6$
 - veldig rett frem..
- $\text{alder} = \text{alder} + 3$
 - Gjør ferdig høyresida for likhetstegnet først

Hva skjer innad på en linje?

- alder = 6
 - veldig rett frem..
- alder = $\overset{6}{\cancel{\text{alder}}} + 3$
 - Gjør ferdig høyresida for likhetstegnet først
 - Alle verdier er på høyresida slik de var før denne linja (alder er altså 6)

Hva skjer innad på en linje?

- alder = 6

- veldig rett frem..

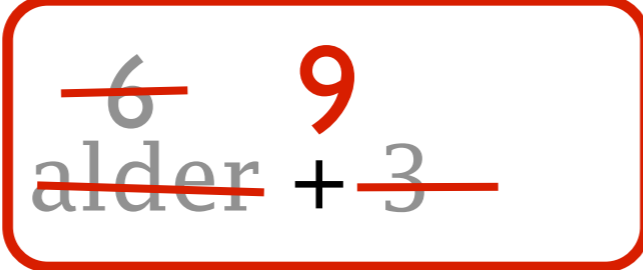
- alder = $\overset{6}{\text{alder}} + \overset{3}{\text{alder}}$

- Gjør ferdig høyresida for likhetstegnet først
 - Alle verdier er på høyresida slik de var før denne linja (alder er altså 6)
 - Regner ut $6+3$ og får 9

Hva skjer innad på en linje?

- alder = 6

- veldig rett frem..

- alder = 

- Gjør ferdig høyresida for likhetstegnet først

- Alle verdier er på høyresida slik de var før denne linja (alder er altså 6)

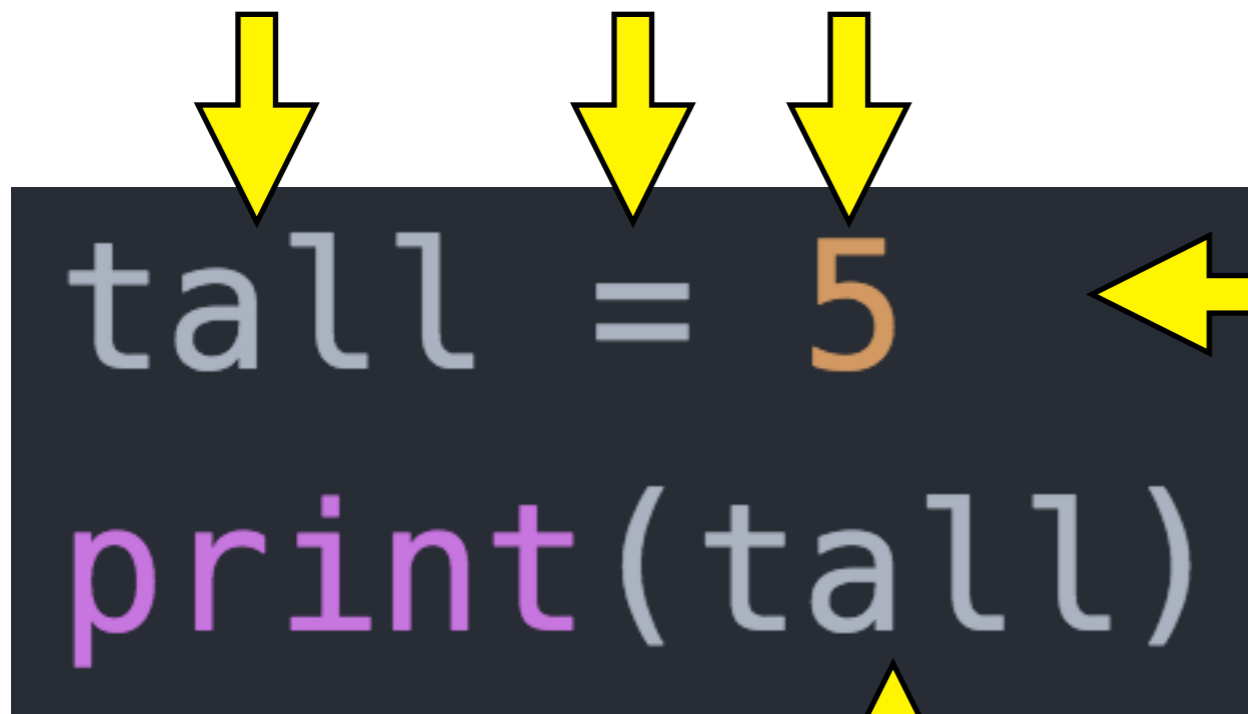
- Regner ut $6+3$ og får 9

- Setter til slutt verdien 9 inn i alder

Formelt om tilordning av variabler

Tilordning:

variabelnavn = uttrykk



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

The diagram shows a dark grey rectangular box containing two lines of code. The first line is 'tall = 5' and the second line is 'print(tall)'. Three yellow arrows point downwards from the text 'variabelnavn = uttrykk' above to the 'tall', '=', and '5' respectively. A yellow arrow points from the right towards the '=' sign. A yellow arrow points upwards from the text 'Bruk av variabel i et uttrykk: henter frem verdien 5' below to the 'tall' in the second line of code.

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
(vi sier "settes lik")

Høyreside
uttrykk:
" verdi -
adde før
njen (**5**)

Evaluering av uttrykk

- Visse operasjoner gjøres alltid før andre
 - Python gjør f.eks. uansett gangning før plussing
- Følgende gir altså samme resultat 33:
 - ~~alder = 5 * 3 + 18~~
15 33

Evaluering av uttrykk

- Visse operasjoner gjøres alltid før andre
 - Python gjør f.eks. uansett gangning før plussing
- Følgende gir altså samme resultat 33:
 - ~~$alder = 5 * 3 + 18$~~
15 33
 - $alder = 18 + 5 * 3$

Evaluering av uttrykk

- Visse operasjoner gjøres alltid før andre
 - Python gjør f.eks. uansett ganging før plussing
- Følgende gir altså samme resultat 33:
 - ~~$\text{alder} = 5 * 3 + 18$~~
15 33
 - $\text{alder} = 18 + 5 * 3$
- Blir tydeligere med paranteser, bruk det!
 - $\text{alder} = (5 * 3) + 18$

Evaluering av uttrykk (forts.)

- For noen formål må man uansett ha paranteser

- $\text{alder} = \frac{5}{(3+2)} * 3 + 18$

Evaluering av uttrykk (forts.)

- For noen formål må man uansett ha paranteser
 - $\text{alder} = \overbrace{(3+2)}^{-5} * \overbrace{3}^{15} + \overbrace{18}^{33}$
- Og for å gjøre det samme tydeligere - nøsting av paranteser er definitivt lovlig:
 - $\text{alder} = ((3+2) * 3) + 18$

Evaluering av uttrykk (forts.)

- For noen formål må man uansett ha paranteser

- $\text{alder} = \overbrace{(3+2)}^{-5} * \overbrace{3}^{15} + \overbrace{18}^{33}$

- Og for å gjøre det samme tydeligere - nøsting av paranteser er definitivt lovlig:

- $\text{alder} = ((3+2) * 3) + 18$

- I praksis er det ikke tall man putter inn på slik måte, men variabler:

- $\text{alder} = ((\text{bachelor} + \text{master}) * \text{antallFagfelt}) + \text{barndom}$

En liten oppgave om variable og uttrykk

(fra eksamen 2016)

- **Oppgave 1a:**

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

```
tall = 4 + (3 * 2)
```

```
tall = tall - 1
```

Hva er en condition?

if condition:
Statements

- **condition** er et boolsk uttrykk (*boolean expression*)
 - Noe som er sant eller ikke sant (**True** eller **False**)
 - Mer presist:
Et *uttrykk* som *evaluerer* til enten *verdien* **True** eller *verdien* **False**
 - Uttrykket kan være sammensatt

*"Just one condition you go to sleep right now:
That you don't touch my daughter
and in the morning, milk the cow"*
(Bob Dylan- Motorpsycho Nightmare)

Boolske uttrykk og verdier (Sannhetsverdier)

- Grunnleggende operasjoner: `<` `>` `==` `!=` `>=` `<=`
 - Kan brukes på tall (`5>3`) og på tekst ("`hei`" `==` "`hei`")
 - Slike uttrykk evaluerer til en verdi av typen **bool**
 - Typen bool har kun to mulige verdier: **True**, **False**
- Kan kombinere sannhetsverdier: `and`, `or`, `not`
 - `5>3 and 8<4`
 - `1>2 or 99>11`
 - `not 3>5`

Boolske uttrykk (forts)

- Utrykk kan naturligvis inneholde ulike variabler
 - `if alder>80 or dager_til_termin<10:`
`print("Ta mitt sete!")`
- Og så kan uttrykkene være nøstede
 - `if (dagerTilTermin>180 and tidspunkt=="morgen")`
`or (alder<6 and antallKilometerBiltur>30):`
`print("du er sikkert kvalm!")`

Boolske variable

- Boolske verdier kan holdes på av variabler (på samme måte som for tall og tekst)
- En boolsk variabel kan brukes alle steder hvor man kan bruke en boolsk verdi

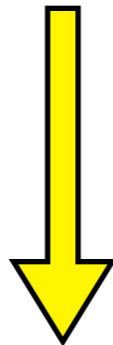
```
betalerHalvPris = alder<18 or alder>66  
if betalerHalvPris:  
    ...
```

Evaluering av funksjoner

- En viktig del av programmering (*som vi kommer mer tilbake til senere*) er ***funksjoner***
 - Kan være biblioteksfunksjoner eller egne
 - Vi har f.eks. brukt *input* som henter en verdi fra brukeren (tastaturet)

Hvordan **input** egentlig virker

- 1: **input** fryser programmet inntil brukeren skriver et svar
- 2: deretter *evaluerer* **input** til en *verdi* av typen tekst (str)
(det brukeren skrev inn)



```
navn = input()
```



Verdien (som brukeren skrev inn)
tilordnes til variabelen navn

Bruk av en funksjon evaluerer til en verdi

```
tall = int("1" + "3")
```



evaluerer til "13"
(og sendes til int)



evaluerer til 13
(av type tall)



verdien 13 tilordnes variabelen tall

Bruk av funksjoner kan også nøstes

```
tall = int( input() )
```



input evaluerer til f.eks. "13"
(som sendes videre til int)



int evaluerer til 13



verdien 13 tilordnes variabelen tall

Hva er galt her?

Oppgave:

Les inn en fart. Dersom fart er 60 eller mindre, skal man lage en streng som består av "fart:" og selve farten (f.eks. "fart:56"). Dersom fart er høyere enn 60, skal man lage strengen "fart:over 60".

```
fart = int(input())
if fart <= 60:
    svar1 = print("Fart: " + fart)
else:
    svar2 = print("Fart: over 60.")
```

Hva er galt her?

Oppgave:

Les inn en fart. Dersom fart er 60 eller mindre, skal man lage en streng som består av "fart:" og selve farten (f.eks. "fart:56"). Dersom fart er høyere enn 60, skal man lage strengen "fart:over 60".

```
fart = int(input(""))  
if fart <= 60:  
    svar1 = print("Fart: " + fart)  
else:  
    svar2 = print("Fart: over 60.")
```

svar1 blir tilordnet
det som print
evaluerer til
- ikke noe (None)

Hva er galt her?

Oppgave:

Les inn en fart. Dersom fart er 60 eller mindre, skal man lage en streng som består av "fart:" og selve farten (f.eks. "fart:56"). Dersom fart er høyere enn 60, skal man lage strengen "fart:over 60".

```
fart = int(input(""))
if fart <= 60:
    svar1 = print("Fart: " + fart)
else:
    svar2 = print("Fart: over 60.")
```

Nå har man enten en variabel svar1 eller svar2 - men har ikke kontroll på hvilken..

Endre det til å bli korrekt

Oppgave:

Les inn en fart. Dersom fart er 60 eller mindre, skal man lage en streng som består av "fart:" og selve farten (f.eks. "fart:56"). Dersom fart er høyere enn 60, skal man lage strengen "fart:over 60".

```
fart = int(input(""))
if fart <= 60:
    svar = "Fart: " + fart
else:
    svar = "Fart: over 60."
```

Endre det til å bli korrekt

Oppgave:

Les inn en fart. Dersom fart er 60 eller mindre, skal man lage en streng som består av "fart:" og selve farten (f.eks. "fart:56"). Dersom fart er høyere enn 60, skal man lage strengen "fart:over 60".

```
fart = int(input(""))
if fart <= 60:
    svar = "Fart: " + str(fart)
else:
    svar = "Fart: over 60."
```

Nøtt: hva skjer her?

Kode:

```
svar = input( input("Sporsmaal: ") )  
print("Du svarte: " + svar)
```


Kjøring:

```
Sporsmaal: hvilket studium?  
hvilket studium?informatikk  
Du svarte: informatikk
```


Nøtt: hva skjer her?

```
Sporsmaal: hvilket studium?  
hvilket studium?informatikk  
Du svarte: informatikk
```

```
svar = input( input("Sporsmaal: ") )
```



skriver ut "Sporsmaal: "
evaluerer til "hvilket studium?"
som sendes til den ytre *input*



skriver ut "hvilket studium?"
evaluerer til "informatikk"
som tilordnes variabelen *svar*

```
print("Du svarte: " + svar)
```

skriver ut "Du svarte: informatikk"

Nøtt: hva skjer her?

```
Sporsmaal: hvilket studium?  
hvilket studium?informatikk  
Du svarte: informatikk
```

```
svar = input( input("Sporsmaal: ") )
```

←→ skriver ut "Sporsmaal: "
evaluerer til "hvilket studium?"
som sendes til den ytre *input*

←→ skriver ut "hvilket studium?"
evaluerer til "informatikk"
som tilordnes variabelen *svar*

```
print("Du svarte: " + svar)
```

skriver ut "Du svarte: informatikk"

Prinsippene for evaluering av en programsetning

- Man begynner alltid innerst og beveger seg utover
 - $18 + ((3+2) * 3)$
 - `int(input())`
- Visse operasjoner utføres før andre
 - $3 + 2 * 3$
- I en tilordning bestemmer man høyresiden først
 - `alder = alder + 3`