

Logiske operatører og logiske uttrykk

Læringsmål:

- Logiske uttrykk
- Betingelser
- Kodeforståelse

Starting Out with Python:

- Kap. 3.2
- Kap 3.5-3.6

I denne oppgaven skal vi lære om logiske uttrykk og hvordan de kan settes sammen med `and` , `or` og `not` .

Generelt om logiske operatører og logiske uttrykk

Dette er ikke en del av oppgaven, men kan være lurt å lese før du begynner.

I de tidligere oppgavene i denne øvingen var alle betingelsene enkle betingelser hvor vi typisk sammenlignet to verdier. De vanlige operatorene vi har for å sammenligne verdier er:

- `==` (som betyr "er lik", merk at her er to likhetstegn nødvendig for å skille fra tilordningsoperatoren)
- `!=` (som betyr "ulik", altså det motsatte av `==`)
- `>` , `<` , `>=` , `<=` (som betyr henholdsvis større, mindre, større eller lik, og mindre eller lik)

Ofte kan beslutninger være avhengig av verdien til **flere** variable, eller for den del flere betingelser for samme variabel.

Ved hjelp av logiske operatører kan vi teste for flere betingelser i samme if-setning. Eksemplet nedenfor viser et lite program som leser inn temperatur og vind og så skal skrive et varsel om ekstremvær hvis det er kaldere enn -30, varmere enn 40, eller mer vind enn 20 m/s.

Her viser vi dette løst på to alternative måter, først med enkle betingelser og en if-elif-setning (linje 5-10), deretter med en enkelt if-setning med en sammensatt betingelse med `or` mellom (linje 13-14). **OBS:** Trykk på View -> Toggle Line Numbers i menyen på toppen for å se linjenummer.

Når det står `or` mellom betingelser er det nok at en av dem er sann for at hele uttrykket skal bli sant (men også sant om begge er sanne); den sammensatte varianten vil dermed gi samme resultat som if-elif-setningen bare at vi med vilje har droppet utropstegn i siste print så du lett skal se hva som er hva.

In []:

```
temp = float(input("Gi inn temperatur i Celsius: "))
vind = float(input("Gi inn vind i m/s: "))

# VARIANT MED if-elif
if temp < -30:
    print("VARSEL: Ekstremvær!")
elif temp > 40:
    print("VARSEL: Ekstremvær!")
elif vind > 20:
    print("VARSEL: Ekstremvær!")

# VARIANT MED SAMMENSATT BETINGELSE
if temp < -30 or temp > 40 or vind > 20:
    print("VARSEL: Ekstremvær")
```

Merk at varianten med sammensatt betingelse kun funker her fordi det er samme tekst som skal skrives i alle de tre tilfellene. Hvis utskriften skulle ha vært mer spesifikk (f.eks. ekstremt kaldt / ekstremt varmt / ...), måtte vi ha brukt if-elif...

Tilsvarende kan vi i noen tilfeller unngå nøstede if-setninger (linje 5-7 i eksemplet under) ved å sette sammen betingelser med `and` (linje 10-11 under). Når det er `and` mellom to betingelser, må **begge** være sanne for at hele uttrykket skal bli sant.

In []:

```
regn = float(input("Hvor mange mm regn er det meldt? "))
vind = float(input("Gi inn vind i m/s: "))

# VARIANT MED NØSTEDE if-setninger
if regn > 0.2:
    if vind < 7.0:
        print("Anbefaler paraply.")

# VARIANT MED SAMMENSATT BETINGELSE
if regn > 0.2 and vind < 7.0:
    print("Anbefaler paraply.")
```

I begge de foregående eksemplene vil nok de fleste si at variantene med sammensatte betingelser er klart å foretrekke framfor if-elif eller nøstet if. Bruk av sammensatte betingelser gjør koden kortere og enklere å forstå.

Det er typisk tre operatører vi bruker for å sette sammen betingelser: `and` , `or` , `not`

Disse virker på følgende måte:

- `betingelse1 and betingelse2` blir True (sant) bare hvis **både** `betingelse1` og `betingelse2` er True, ellers blir uttrykket False (usant)
- `betingelse1 or betingelse2` blir True (sann) hvis **minst en** av betingelsene er True, ellers False
- `not betingelse1` blir True hvis `betingelse1` er False, og False hvis `betingelse1` er True
- Man kan sette sammen mer komplekse betingelser ved å bruke flere av dem.

Presedensregler: `not` har større presedens enn `and` , som har større enn `or` .

F.eks. anta at

`if regn > 0.2 and vind < 7.0 or solbrentfare > 0.9 and not solkrembeholdning > 0:` er gitt som betingelse for å ta med paraply

Ifølge presedensreglene vil `not` evalueres først, deretter `and` , og til slutt `or` . Uttrykket kan dermed ses som to alternative måter for å anbefale paraply (delt av `or` i midten), nemlig:

ENTEN at både `regn > 0.2` og `vind < 0.7` er sanne. Det spiller da ingen rolle hvilken verdi vi har for `solbrentfare` osv. (siden den har `or` foran seg) ELLER at `solbrentfare > 0.9` er sann, samtidig som `solkrembeholdning > 0` er usann (slik at `not solkrembeholdning > 0` vil være sann) I det siste tilfellet er det rimelig å anta at paraply er tenkt å beskytte mot sol, ikke mot regn.

Merk at selv om sammensatte betingelser gjorde koden kortere og enklere å forstå i eksemplene over, er ikke dette alltid tilfelle. Hvis man ender med veldig store sammensatte betingelser, vil disse i seg selv være vanskelig å forstå, slik at kanskje noe oppsplitting med if-elif... eller nøsting av if-setninger kunne ha vært bedre.

Men se først om det er mulig å skrive betingelsen enklere. Dette gjelder særlig hvis det er mye bruk av `not` . Akkurat som mye bruk av negasjoner i vanlig norsk vil gjøre en tekst vanskelig å forstå, vil mye bruk av `not` gjøre Python-betingelser vanskelige å forstå - og kunne kanskje vært unngått. F.eks.

- `not a < 10` kan like gjerne skrives `a >= 10`
- `not a == 10` kan like gjerne skrives `a != 10`
- `not (c < 0 or c > 255)` kan like gjerne skrives `c >= 0 and c <= 255` eller enda enklere `0 <= c <= 255` . Det fins tilfeller hvor `not` er lurt å bruke, men bruk den med måte, bare når bedre alternativer ikke finnes.

a)

Hvilke av de følgende logiske uttrykkene vil gi True om $x=3$, $y=8$ og $z=-3$?

Uttrykk 1: $-5 < z$ and $5 > z$

Uttrykk 2: $\text{not } y == 8$

Uttrykk 3: $x == 8$ or $y == 8$

Uttrykk 4: $\text{not } (x \leq 3 \text{ or } y \geq 9)$

Uttrykk 5: $\text{not } (x**2 \neq 8 \text{ and } y-z == 5)$ or $x+y == y-z$

Dobbelklikk på teksten under og skriv svaret ditt der:

Svar: True:1,3,5 False:2,4

Hint

Om du sliter kan du sjekke ved å skrive et program som gir x, y og z verdiene 3, 8, -3 og bruke `print(<logisk uttrykk>)` .

Husk presedensregler: `not` har større presedens enn `and` , som har større enn `or` .

Du kan teste kode her:

In []:

b)

Nedenfor følger en kodesnutt hvor brukeren bes om å gi inn to tall innenfor et lovlig verdiområde, som forklart av ledeteksten - altså mellom 40-50 eller mellom 70-90. Om ikke begge tallene har fått lovlig verdi, skal koden i else-blokken utføres.

Dessverre er det feil i koden som gjør at den ikke fungerer slik den skal. Din oppgave er å rette opp i disse feilene.

Rett opp feilene under og sjekk at koden fungerer for alle tilfeller

In [6]:

```
print("Gi inn a og b, begge heltall i intervall <40,50> eller <70,90>:" )
a = int(input("Verdi for a: "))
b = int(input("Verdi for b: "))

if ((a>70 and a<90) or (a>40 and not a<=50)) and ((70<b<90) or (b>40 and b<50)):
    print("Tallene er begge i gyldige intervall ^u^")
else:
    print("Minst ett av tallene er utenfor et gyldig intervall :(")
```

Gi inn a og b, begge heltall i intervall <40,50> eller <70,90>:

Verdi for a: 75

Verdi for b: 100

Minst ett av tallene er utenfor et gyldig intervall :(

Hint

Husk parenteser!

c)

Du skal i denne oppgaven fullføre koden gitt nedenfor slik at den fungerer korrekt. Programmet har laget 10 pannekaker, noe som er mer enn det klarer å spise, og ønsker å dele noen av dem med deg. Men om du er grådig og spør om å få flere enn 10, vil du få beskjed om at det ikke er mulig. Om du derimot ønsker å gi programmet pannekaker (skriver inn et negativt tall), vil du også få beskjed om at det ikke er mulig. **Endre på koden under**

In [4]:

```
print("Hei! Jeg har 10 pannekaker jeg ønsker å dele med deg ^u^")
p = int(input("Hvor mange pannekaker ønsker du? "))

if p<0 or p>10:                #Kode mangler her
    print("Beklager, men det er nok ikke mulig")
else:
    r = 10-p
    print("Da blir det",p,"på deg og",r,"på meg :D")
```

```
Hei! Jeg har 10 pannekaker jeg ønsker å dele med deg ^u^
Hvor mange pannekaker ønsker du? 12
Beklager, men det er nok ikke mulig
```

Eksempel på kjøring

```
Hei! Jeg har 10 pannekaker jeg ønsker å dele med deg ^u^
Hvor mange pannekaker ønsker du? 4
Da blir det 4 på deg og 6 på meg :D
```

```
Hei! Jeg har 10 pannekaker jeg ønsker å dele med deg ^u^
Hvor mange pannekaker ønsker du? 100
Beklager, men det er nok ikke mulig
```

d)

I denne deloppgaven skal pannekakeprogrammet utvides. Det skal nå også spørre om personen liker pannekaker og lagre det i en boolsk variabel, som vist i koden under:

```
print("Hei! Jeg har 10 pannekaker jeg ønsker å dele med deg ^u^")
p = int(input("Hvor mange pannekaker ønsker du? "))
s = input("Er du glad i pannekaker? (J/N) ")

if s == 'J':
    elsker_pannekaker = True
else:
    elsker_pannekaker = False
```

Pannekakeprogrammet elsker pannekaker og er lite forståelsesfull ovenfor de som ikke gjør det. Derfor, om `elsker_pannekaker` er `False`, skal det også svare "...ikke mulig" selv om brukeren ber om et antall pannekaker innenfor lovlig intervall $<0,10>$.

Din jobb er å samle alle disse betingelsene i et logisk uttrykk og skrive dette logiske uttrykk inn i koden nedenfor (der ... står), slik at programmet får rett oppførsel. **Endre på koden under**

In [9]:

```
print("Hei! Jeg har 10 pannekaker jeg ønsker å dele med deg ^u^")
p = int(input("Hvor mange pannekaker ønsker du? "))
s = input("Er du glad i pannekaker? (J/N) ")

if s == 'J':
    elsker_pannekaker = True
else:
    elsker_pannekaker = False

if (p<0 or p>10) or elsker_pannekaker == False:           #Kode mangler her
    print("Beklager, men det er nok ikke mulig")
else:
    r = 10-p
    print("Da blir det",p,"på deg og",r,"på meg :D")
```

```
Hei! Jeg har 10 pannekaker jeg ønsker å dele med deg ^u^
Hvor mange pannekaker ønsker du? -3
Er du glad i pannekaker? (J/N) J
Beklager, men det er nok ikke mulig
```

Eksempel på kjøring

```
Hei! Jeg har 10 pannekaker jeg ønsker å dele med deg ^u^
Hvor mange pannekaker ønsker du? 5
Er du glad i pannekaker? (J/N) J
Da blir det 5 på deg og 5 på meg :D
```

```
Hei! Jeg har 10 pannekaker jeg ønsker å dele med deg ^u^
Hvor mange pannekaker ønsker du? 7
Er du glad i pannekaker? (J/N) N
Beklager, men det er nok ikke mulig
```