



Kapitel 2: Programmera i Python

Övning 1: Villkorssatser

Vi skall här titta lite mer på villkorssatser, alltså if-satser.

Mål:

- Att kunna använda villkorssatser.
- Att kunna villkorssatsers syntax i Python

I alla programmeringsspråk finns det något sätt att välja olika vägar genom programmet beroende på något villkor. Ett sådant sätt är med hjälp av villkorssatser i formen om A så B. Om villkoret A är sant så utförs B. Oftast finns det också något sätt att lägga till vad som skall göras om A inte är sant. Vi har då en sats i formen om.. så .. annars... I Python ser det ut som i rutan till höger.

```
if villkor:  
    gör-om-sant  
else :  
    gör-om-falskt
```

Exempel:

Ett biluthyrningsföretag har följande erbjudande: Hyrespriset är 660 kr för de första 70 kilometerna plus 25 kr för varje extra körd kilometer.

Vi ska göra ett program som frågar efter antal kilometer och sedan skriver ut motsvarande pris.

Starta ett nytt dokument och lägg ett Python-program.

Börja med att med hjälp av input-satsen frågan om antalet kilometer och ge värdet till en variabel k.

Kommandot **input()** returnerar alltid en sträng, men vi vill ha ett tal för k. Detta kan göras genom att ändra typen till ett heltal med funktionen **int()**. (Här antar vi att antalet kilometer är ett heltal)

De första två raderna i programmet kan nu se ut så här.

```
k=input("Antal hela kilometer:")  
k=int(k)
```

Lärartips: Om antal kilometer inte måste vara ett heltal så kan vi använda **float(k)** istället för **int(k)**.



För att beräkna rätt pris måste programmet först testa om antalet kilometer är mer eller mindre än 70. Du kan använda if-strukturen för det.

Du hittar den med meny > 4: inbyggda > 2: Kontroller > 2: if...else..

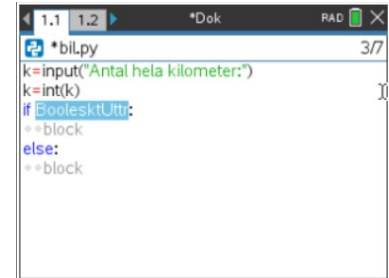
I mallen till höger ska villkoret skrivas direkt efter if där texten BooleskUttr står.

Det första blocket innehåller de kommandon som skall köras om villkoret gäller.

Kommandon som skrivs i det andra blocket körs om villkoret är falskt.

Alla kommandon i samma block måste börja med samma antal mellanslag. Det är så Python skiljer på olika nivåer i programmet.

Du ser också ett kolon i slutet av if-raden och på else-raden. Detta markerar för Python att det nu kommer ett block av kod.



```
*bil.py 3/7
k=input("Antal hela kilometer:")
k=int(k)
if BooleskUttr:
  ==block
else:
  ==block
```

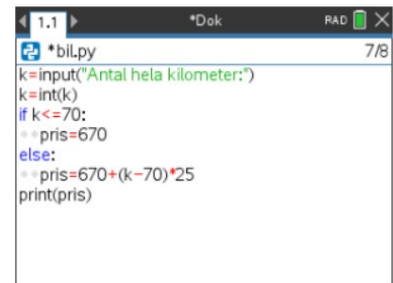
Vi kan nu fylla i resten av if-mallen. Man skriver \leq som $<=$.

Resultatet av beräkningen måste naturligtvis också skrivas ut.

Du kan köra programmet med Ctrl + R.

När du kör programmet öppnas konsolen.

Om du sedan vill köra programmet igen kan du göra det från konsolen med Ctrl + R igen.



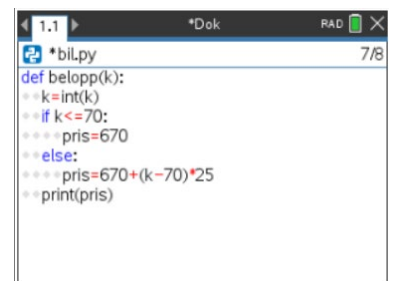
```
*bil.py 7/8
k=input("Antal hela kilometer:")
k=int(k)
if k<=70:
  ==pris=670
else:
  ==pris=670+(k-70)*25
print(pris)
```

Tips: En rad inte är rätt indragen så kan man placera kursorn i början på raden och sedan trycka på tab. Skift-tab gör det motsatta och tar istället bort mellanslag.

Du kan också sätta in beräkningarna i en funktion. Vi tar bort input-satsen och låter k vara en parameter till funktionen istället.

I exemplet till höger har funktionen getts namnet "belopp".

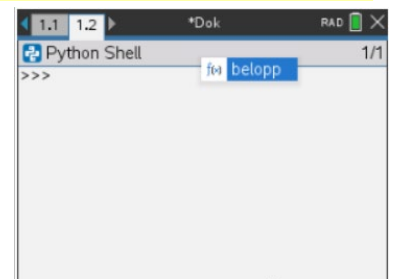
Var uppmärksam på syntaxen: Funktionsblocket är indraget i två mellanslag medan blocken i if-satserna nu måste vara indragna med två extra mellanslag.



```
*bil.py 7/8
def belopp(k):
  ==k=int(k)
  ==if k<=70:
  ==++ pris=670
  ==else:
  ==+++ pris=670+(k-70)*25
  ==print(pris)
```

När du kör det här programmet händer ingenting vid första anblicken. Men om du nu trycker på var-tangenten (eller Meny > A Variabler > 1 Variabler: Program för senaste körning) så ser du funktionen och du kan välja den.

Om du ger parametern ett värde och trycker enter så får du det beräknade priset.



```
Python Shell 1/1
>>> belopp
```

Tips: Man behöver inte använda menyn. Man kan helt enkelt skriva in funktionsnamnet. I datorprogramvaran så kan man använda Ctrl-L