

Kapitel 2: Starta programmering på riktigt

I den första övningen i kapitel 2 får du lära dig att skriva och använda en villkorsinstruktion i Python.

I ett program är det särskilt vanligt att behöva styra genomförandet efter villkor som påverkar de olika variablerna. Ett villkor är en sats som kan vara sann eller falsk. Till exempel: $a = b$ eller $a \geq b$ men också n är ett villkor som verifieras med de värden som tilldelats dessa variabler. I ett program kan ett villkor testas och beroende på om det är sant eller falskt, utföra ett processande eller inte. Detta kallas villkorligt processande.

Lärarkommentar: I Pythonspråket finns det ingen instruktion för att ange slutet på en instruktion med villkor. Det visas av indragen (indenteringen) för instruktionerna A och B. elif är sammandragning av else och If. För att testa likheten mellan två värden i Pythonspråket används två likhetstecken "==".

Exempel: Ett biluthyrningsföretag erbjuder sina kunder följande avtal: Ett weekendpris för 66 euro plus 0,25 euro per kilometer över 70 km. Ditt jobb är att konstruera ett skript som automatiskt beräknar kostnaden för avtalet baserat på den tilltänkta sträckan.

Genomförande:

- Starta Python-appen genom att trycka på tangenten `prgm`.
- Starta sedan ett nytt -skript genom att trycka på f3(New) och döp det till "BILHYRA". Validera genom att trycka på f5-tangenten (OK).
- Startas skriptet genom att definiera en funktion. Tryck då på f1 (Fns...) och tryck sedan på 1:def function():. Namnge funktionen till **kostnad(s)**. s står här för sträckan i km.
- Tryck nu på f1-tangenten (Fns...) och välj sedan Ctl, som står för Kontroll. Här finns nu ett antal val för olika villkorssatser plus ett antal alternativ för s.k. loopar. Välj nu **2:if .. else**. Detta kan översättas till svenska som "om .. annars".

På nästa sida ser vi resultatet.

Övning 1: Instruktioner med villkor
Syfte:

- Skriva och använda en villkorsinstruktion
- Återanvända begreppet funktion i Python

If villkor

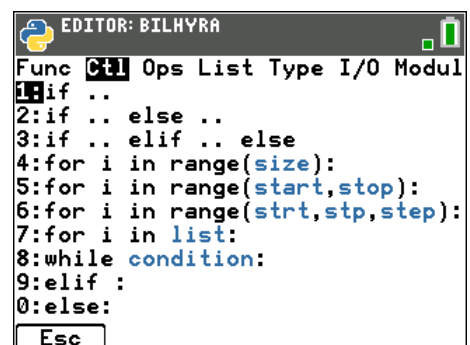
```

Instruktion A
else
Instruktion B
    
```



```

EDITOR: BILHYRA
PROGRAM LINE 0001
def kostnad(s):
    ..
    
```



```

EDITOR: BILHYRA
Func Ctl Ops List Type I/O Modul
1:if ..
2:if .. else ..
3:if .. elif .. else
4:for i in range(size):
5:for i in range(start,stop):
6:for i in range(strt,stp,step):
7:for i in list:
8:while condition:
9:elif :
0:else:
Esc
    
```

Vi ser strukturen på skärmen med nödvändiga indrag för villkorsblocket. I Python-språket så hör kommandon med samma antal indrag ihop. Detta är en del av själva språket. Man får då en bra bild av programstrukturen.

Du kan se intill hur funktionen kan se ut, till exempel. Om sträckan s är 70 eller mindre blir resultatet 66. I det andra fallet beräknar funktionen först antalet kilometer över 70 och sedan kostnaden. I det här exemplet kan du se att ett block kan bestå av flera regler. Du infogar symbolen \leq med f1(Fns...) och sedan Ops eller också kan du trycka på $\boxed{2nd} \boxed{math}$.

Nu kan vi köra programmet för några olika värden på sträckan s . Du får fram kostnadsfunktionen genom att trycka på tangenten \boxed{vars} .

Lärarkommentar: Till höger har vi ett skript som ser lite annorlunda ut än skriptet Bilhyra. Be eleverna att skriva in skriptet i editorn och göra några körningar vid olika värden på sträckan s . Be dem förklara vad det är för beräkningar som utförs. Förklara också användandet av villkorssatsen `if .. elif .. else`.

Lärarkommentar: Hur *plottar* man nu denna kostnadsfunktion på räknaren. Den kallas för en styckvis funktion eftersom den består av olika bitar som är definierade för olika intervall. Vi ska här visa ett sätt att plotta funktionen som handlar om hur räknaren utvärderar jämförelser när man använder s.k. relationsoperatörer. Se nästa sida

Räknaren utvärderar påståenden och ger ett av två möjliga sanningsvärden: 1 = **Sant** och 0 = **Falskt**. När är $A < 2$ sant?

```

EDITOR: BILHYRA
PROGRAM LINE 0005
def kostnad(s):
  if :
  else:

```

```

EDITOR: BILHYRA
PROGRAM LINE 0001
def kostnad(s):
  if s<=70:
    k=66
  else:
    k=66+0.25*(s-70)
  return k

```

```

PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running BILHYRA
>>> from BILHYRA import *
>>> kostnad(65)
66
>>> kostnad(74)
67.0
>>> |

```

```

EDITOR: BILHYRA2
PROGRAM LINE 0001
def kostnad(s):
  if s<=70:
    k=66
  elif 70<s<=150:
    k=66+0.25*(s-70)
  else:
    k=66+0.25*(s-70)+0.40*(s-150)
  return k

```

Räknaren returnerar 1 för sant och 0 för falskt när A är mindre än 2. Se skärmbilden till höger.

Om du delar en funktion med ett påstående divideras funktionen med 1 när påståendet är sant och funktionen divideras med 0 när påståendet är falskt. Nu kan man ju inte dividera med 0 och då kan man heller inte plotta funktionen.

Den funktion vi nu ska plotta kan skrivas så här:

$$\begin{cases} x = 66, & 0 \leq x \leq 70 \\ x = 66 + 0.25 \cdot (x - 70), & 70 < x \leq 150 \\ x = 66 + 0.25 \cdot (x - 70) + 0.40 \cdot (x - 150), & x > 150 \end{cases}$$

Ett sätt att mata in detta i Y=-editorn blir då som i skärmen här bredvid. Här måste man se till att man ställt in visningen som MATHPRINT (tryck på `mode` för att ställa in). I Y-editorn trycker man sedan på `alpha` `y=` för att få inmatning som bråk. Sedan skriver man enligt skärmen till höger

Med en bra fönsterinställning kan plottningen sedan bli så här. Man ser lätt de tre delarna då de är i olika färg både i inmatningsskärmen och plottningsskärmen.

