

Föreläsning 2

Programmeringsteknik

DD1310

- python introduktion
- Utskrift
- Inläsning
- Variabler
- Datatyp
- Aritmetiska operatorer
- Omvandling av typer
- Reserverade ord
- logiska operatorer
- If-sats
- While-sats
- kommentarer

Översikt

Programmering

Utskrift

Syntax

Variabel

Inläsning

Omvandling

Kommentarer

Operatorer + och *

Minilabb

Jämförelse op.

If,elif, else

While-sats

Logiska operatorer

Sammanfattning

OLI-resultat

- Variabler och utskrifter
 - 141 : >80%
 - 6: 50%-79%
 - 8: 40%-50%
 - 11: <40%
 - 3: 0%
- Inmatning och styrstrukturer
 - 90 : >80%
 - 1: 50%-79%
 - 0: 40%-50%
 - 6: <40%
 - 73 : 0%
- While-slingor: 80: >80% och 90: 0%

Översikt

Programmering

Utskrift

Syntax

Variabel

Inläsning

Omvandling

Kommentarer

Operatorer + och *

Minilabb

Jämförelse op.

If,elif, else

While-sats

Logiska operatorer

Sammanfattning

Programmering

- Programmering betyder att instruera en dator
- Ett program är alltså en sekvens av instruktioner till en dator
- Datorer kan endast förstå maskinkod d.v.s kombination av ettor och nollor
- Att skriva maskinkod d.v.s instruktioner i form av kombination av ettor och nollor är oerhört svårt och tidskrävande

Översikt

Programmering

Utskrift

Syntax

Variabel

Inläsning

Omvandling

Kommentarer

Operatorer + och *

Minilabb

Jämförelse op.

If,elif, else

While-sats

Logiska operatorer

Sammanfattning

Programspråk

- Programspråk är ett verktyg som används för att kunna omvandla mer begripliga och lätthanterliga instruktioner till maskinkod
- En instruktion som är skrivet i ett programspråk måste följa språkets regler för att kunna omvandlas till maskinkod

Översikt

Programmering

Utskrift

Syntax

Variabel

Inläsning

Omvandling

Kommentarer

Operatorer + och *

Minilabb

Jämförelse op.

If,elif, else

While-sats

Logiska operatorer

Sammanfattning

Utskrift på skärmen

1

```
print("Hej")  
print("Hur är det?")
```

Hej

Hur är det?

2

```
print("Hej",end="")  
print("Hur är det?")
```

HejHur är det?

Översikt

Programmering

Utskrift

Syntax

Variabel

Inläsning

Omvandling

Kommentarer

Operatorer + och *

Minilabb

Jämförelse op.

If,elif, else

While-sats

Logiska operatorer

Sammanfattning

Syntax

- Syntax betyder språkets struktur.
 - Följande är en massa svenska ord:
labbarna, i, tid, gör, till, bonus, samla, datorprovet, ! , och
 - Kan man bygga en vettig mening med orden?
 - Kan en maskin gissa det utan att missuppfatta meningen?

Översikt

Programmering

Utskrift

Syntax

Variabel

Inläsning

Omvandling

Kommentarer

Operatorer + och *

Minilabb

Jämförelse op.

If,elif, else

While-sats

Logiska operatorer

Sammanfattning

Syntax

- Ovanstående gäller också för olika programspråk bl.a. Python:

Följande rad blir datorn förvirrad av p.g.a syntaxfel:

```
"Python är enkelt!"() print
```

För att datorn ska förstå, ska man skriva som nedan:

```
print("Python är enkelt!")
```

Översikt

Programmering

Utskrift

Syntax

Variabel

Inläsning

Omvandling

Kommentarer

Operatorer + och *

Minilabb

Jämförelse op.

If,elif, else

While-sats

Logiska operatorer

Sammanfattning

Variabel

Med hjälp av variabler kan man lagra ett värde i datorns minne så att programmet kan komma åt det lagrade värdet vid senare tillfällen i programmet.

Exempel:

```
age = 20
```

```
name = "Vahid"
```

```
length = 1.70
```


Översikt

Programmering

Utskrift

Syntax

Variabel

Inläsning

Omvandling

Kommentarer

Operatorer + och *

Minilabb

Jämförelse op.

If,elif, else

While-sats

Logiska operatorer

Sammanfattning

Variabelnamn

- Använd så korta och beskrivande variabelnamn som möjligt
- Var konsekvent vid val av variabelnamn
- Försök att följa språkets tradition för val av variabelnamn
- Python är ett skiftlägekänslig (case sensitive) språk

Översikt

Programmering

Utskrift

Syntax

Variabel

Inläsning

Omvandling

Kommentarer

Operatorer + och *

Minilabb

Jämförelse op.

If,elif, else

While-sats

Logiska operatorer

Sammanfattning

Reserverade ord

En del ord får inte användas som variabelnamn, för att de har en viktig betydelse i python och därför kallar vi dem ***reserverade ord***:

<u>is</u>	<u>elif</u>	<u>import</u>	<u>global</u>
<u>if</u>	<u>from</u>	<u>pass</u>	<u>class</u>
<u>in</u>	<u>return</u>	<u>except</u>	<u>continue</u>
<u>def</u>	<u>and</u>	<u>else</u>	raise
<u>del</u>	<u>or</u>	<u>continue</u>	assert
<u>for</u>	<u>while</u>	<u>break</u>	exec
<u>try</u>	<u>not</u>	<u>finally</u>	<u>lambda</u>

Översikt

Programmering

Utskrift

Syntax

Variabel

Inläsning

Omvandling

Kommentarer

Operatorer + och *

Minilabb

Jämförelse op.

If,elif, else

While-sats

Logiska operatorer

Sammanfattning

Datatyper

- Datatyp för text (Sträng):
 - str t.ex: "hej", "12"
- Numeriska datatyper:
 - int t.ex: 12
 - float t.ex: 12.0

Översikt

Programmering

Utskrift

Syntax

Variabel

Inläsning

Omvandling

Kommentarer

Operatorer + och *

Minilabb

Jämförelse op.


If,elif, else

While-sats

Logiska operatorer

Sammanfattning

Inläsning

- Inläsning från tangentbordet görs m.h.a inbyggda funktionen `input()` datorn väntar tills användaren tryckt på "Enter" sedan
- `input()` kan ta högst **en** parameter
- Om man använder `input` med parameter så skrivs ut parametern först sedan väntar `input` på inmatning:
- `input("Vad heter du?")`

parameter

Översikt

Programmering

Utskrift

Syntax

Variabel

Inläsning

Omvandling

Kommentarer

Operatorer + och *

Minilabb

Jämförelse op.

If,elif, else

While-sats

Logiska operatorer

Sammanfattning

Omvandling av typer

Omvandling mellan typerna `str`, `int` och `float` görs m.h.a följande funktioner:

- `str(x)`
- `int(x)`
- `float(x)`

Exempel:

```
age_str = input("ange ålder:")  
age = int(age_str)
```

Översikt

Programmering

Utskrift

Syntax

Variabel

Inläsning

Omvandling

Kommentarer

Operatorer + och *

Minilabb

Jämförelse op.

If,elif, else

While-sats

Logiska operatorer

Sammanfattning

Kommentarer

```
# This is a comment  
print("Learning Python is easy!")  
Learning Python is easy!
```

Varför och när skriver man
kommentarer i ett program?

Översikt

Programmering

Utskrift

Syntax

Variabel

Inläsning

Omvandling

Kommentarer

Operatorer + och *

Minilabb

Jämförelse op.

If,elif, else

While-sats

Logiska operatorer

Sammanfattning

operatorerna + och *

- operatoren + används för att konkatenera två strängar.

"ab" + "ba" → "abba"

- Man kan använda * följd av ett heltal för att upprepa en sträng ett antal gånger.

"mam" * 2 → "mammam"

Översikt

Programmering

Utskrift

Syntax

Variabel

Inläsning

Omvandling

Kommentarer

Operatorer + och *

Minilabb

Jämförelse op.

If,elif, else

While-sats

Logiska operatorer

Sammanfattning

Mini labb

Skriv ett program som frågar efter åldern och sedan beräknar examensåldern och till slut skriver ut examensåldern.

```
# 1. fråga efter ålder
```

```
# 2. läs in ålder
```

```
# 3. addera angiven ålder med 4.5
```

```
# 4. skriv ut resultatet
```


Översikt

Programmering

Utskrift

Syntax

Variabel

Inläsning

Omvandling

Kommentarer

Operatorer + och *

Minilabb

Jämförelse op.

If,elif, else

While-sats

Logiska operatorer

Sammanfattning

Jämförelse operatorer

Följande är operatorer som resulterar ett boolskt värde d.v.s. sant (`True`) eller falskt (`False`). Med hjälp av följande operatorer kan man skapa villkor-uttryck:

`==`, `is`

`!=`, `not`

`<`

`<=`

`>`

`>=`

Översikt

Programmering

Utskrift

Syntax

Variabel

Inläsning

Omvandling

Kommentarer

Operatorer + och *

Minilabb

Jämförelse op.

If,elif, else

While-sats

Logiska operatorer

Sammanfattning

Exempel

<pre>A = 1 B = 1 A == B True</pre>	<pre>A = "hej" B = "hej" A == B True</pre>
<pre>A = 1 B = 1 A is B True</pre>	<pre>A = "hej" B = "hej" A is B True</pre>

Översikt

Programmering

Utskrift

Syntax

Variabel

Inläsning

Omvandling

Kommentarer

Operatorer + och *

Minilabb

Jämförelse op.

If, elif, else

While-sats

Logiska operatorer

Sammanfattning

If-sats

En If-sats används för att undvika eller påtvinga exekvering av en eller flera satser.

Exempel:

```
if bokpris > 500:  
    print("dyr bok!")  
    print("ingen affär")  
print("hejdå")
```

Översikt

Programmering

Utskrift

Syntax

Variabel

Inläsning

Omvandling

Kommentarer

Operatorer + och *

Minilabb

Jämförelse op.

If, elif, else

While-sats

Logiska operatorer

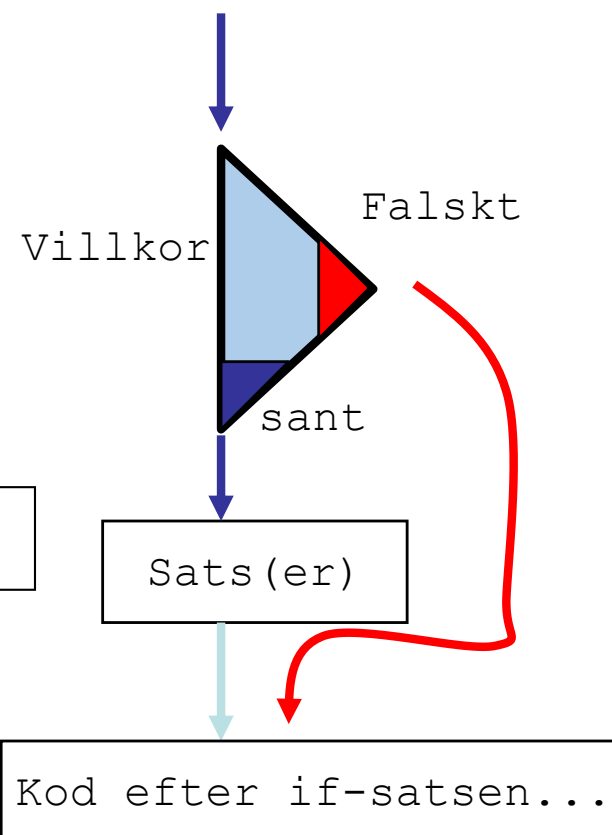
Sammanfattning

If-satsens struktur

```
if villkor:  
    Sats(er)
```

Kod efter
if-satsen

Indentering är viktig!!!



Översikt

Programmering

Utskrift

Syntax

Variabel

Inläsning

Omvandling

Kommentarer

Operatorer + och *

Minilabb

Jämförelse op.

If, elif, else

While-sats

Logiska operatorer

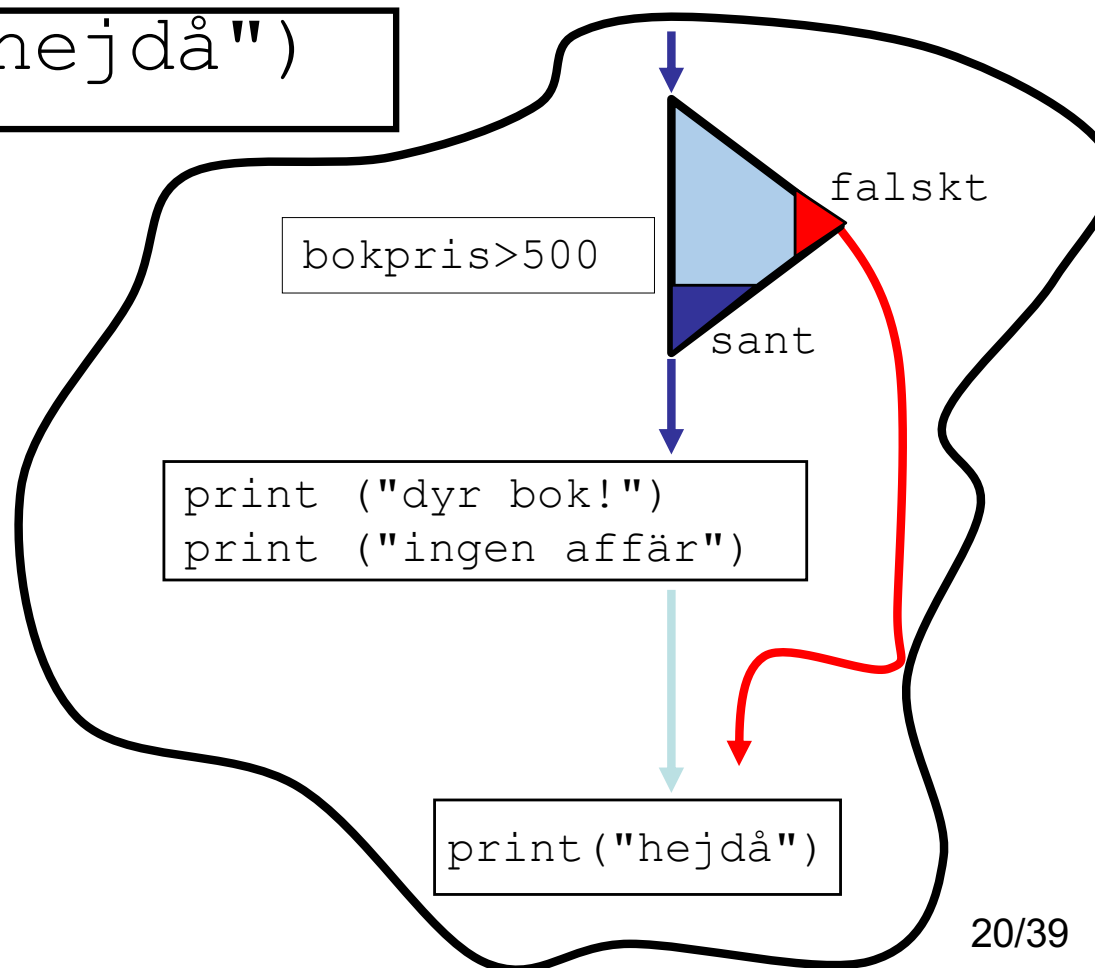
Sammanfattning

Exempel

```
if bokpris > 500 :
```

```
    print("dyr bok!")  
    print("ingen affär")
```

```
print("hejdå")
```



Översikt

Programmering

Utskrift

Syntax

Variabel

Inläsning

Omvandling

Kommentarer

Operatorer + och *

Minilabb

Jämförelse op.

If, elif, else

While-sats

Logiska operatorer

Sammanfattning

Exempel

```
age = input("ålder:")  
age = float(age)  
if age < 20:  
    print ("Åldersgräns är  
20 för att kunna vara  
systembolagets kund")
```

elif och else

`if` -satsen kan kombineras med `elif` och `else` för att åstadkomma lite mer avancerade program.

Exempel:

```
if bokpris > 500:
    print("dyrbok, ingen affär!")
elif bokpris > 300:
    print("dyr men jag behöver boken!")
else:
    print("billig bok, köp snabbt!")
```

Översikt

Programmering

Utskrift

Syntax

Variabel

Inläsning

Omvandling

Kommentarer

Operatorer + och *

Minilabb

Jämförelse op.

If,elif, else

While-sats

Logiska operatorer

Sammanfattning

elif och else

- `elif` och `else` måste användas i samband med `if`.

Översikt

Programmering

Utskrift

Syntax

Variabel

Inläsning

Omvandling

Kommentarer

Operatorer + och *

Minilabb

Jämförelse op.

If, elif, else

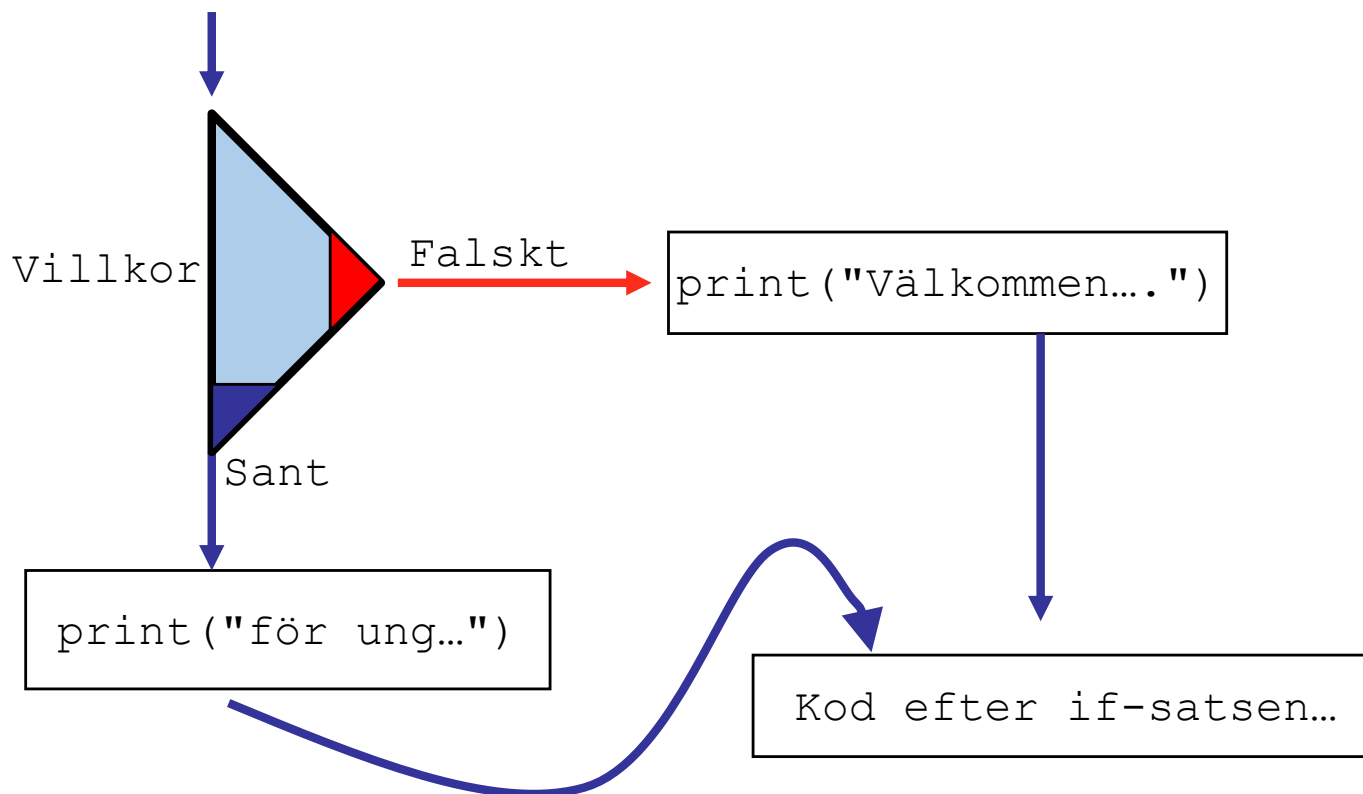
While-sats

Logiska operatorer

Sammanfattning

Exempel

```
if age < 20:  
    print("för ung för att handla här")  
else:  
    print("Välkommen till systembolaget!")
```



while-loop(while-slinga)

- while-slinga används för att exekvera en eller flera satser ett antal gånger.

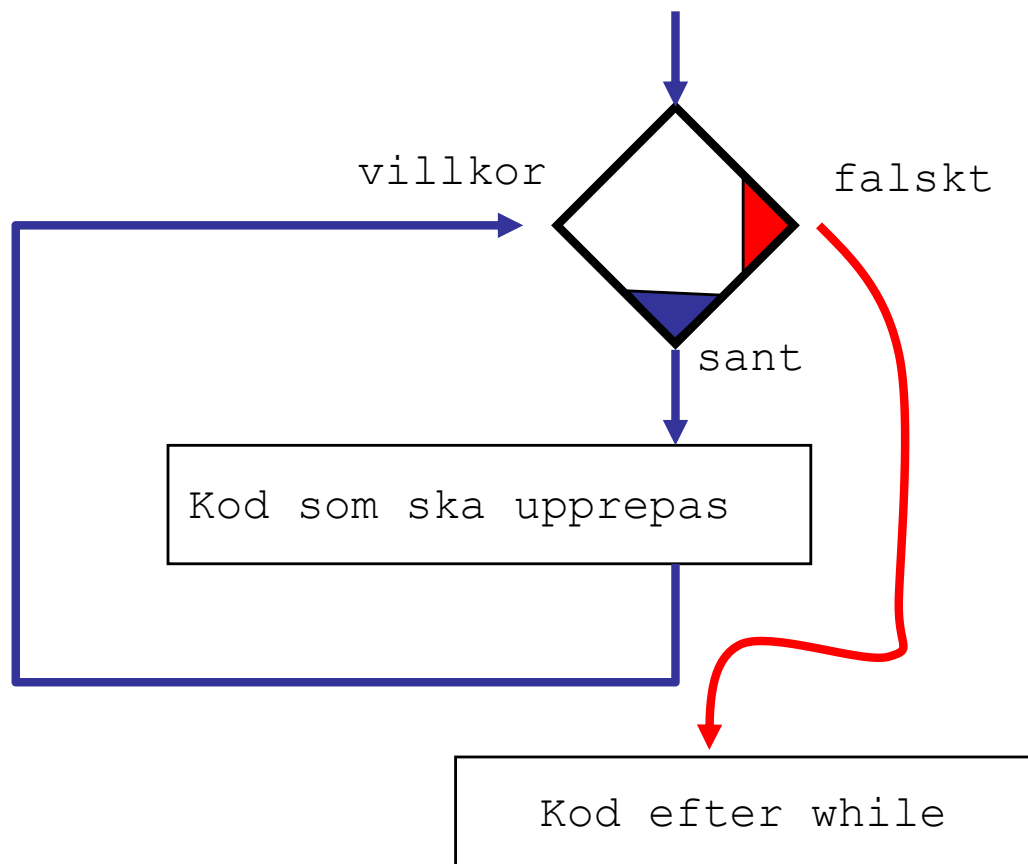
- Exempel:

```
varv = 0
while varv < 3:
    print("Hej")
    varv = varv+1
```

while-satsen struktur

while villkor:

kod som ska upprepas



Översikt

Programmering

Utskrift

Syntax

Variabel

Inläsning

Omvandling

Kommentarer

Operatorer + och *

Minilabb

Jämförelse op.

If,elif, else

While-sats

Logiska operatorer

Sammanfattning

Indentering

Indentering (indragning av kod) har stor betydelse i python.

Hur många gånger skrivs Hej ut av följande program?

```
varv = 0
while varv < 3:
    print ("Hej")
varv = varv + 1
```

Logiska operatorer

- Villkor kan kombineras med operatorerna and, or, not

- Exempel:

```
if pris<1000 and taltid>=100:  
    print("Telefonfynd!")
```

A	B	A and B	A or B	not A
True	True	True	True	False
True	False	False	True	False
False	True	False	True	True
False	False	False	False	True

Översikt

Programmering

Utskrift

Syntax

Variabel

Inläsning

Omvandling

Kommentarer

Operatorer + och *

Minilabb

Jämförelse op.

If,elif, else

While-sats

Logiska operatorer

Sammanfattning

Sammanfattning

- Använd korta men beskrivande variabelnamn
- Var noggrann med datatyper
- `input()` används för inmatning
- Planera ditt program innan du börjar skriva kod för det
- If-satser används för att villkorlig köra en eller flera satser

Översikt

Programmering

Utskrift

Syntax

Variabel

Inläsning

Omvandling

Kommentarer

Operatorer + och *

Minilabb

Jämförelse op.

If, elif, else

While-sats

Logiska operatorer

Sammanfattning

Sammanfattning

- while-slinga (while-sats) används för att upprepa en eller flera satser
- Genom att använda while-satser får man kortare kod, mer genomskådlig kod och ett mer flexibelt program
- När två eller fler while-slingor hamnar i varandra kallas det nästlade slingor