



## Föreläsning 2 Programmeringsteknik och C DD1316

Mikael Djurfeldt <mdj@kth.se>



## Föreläsning 2 Programmeringsteknik och C

- Datatyp
- Aritmetiska operatörer
- Funktioner
- Omvandling av typer
- Reserverade ord
- Logiska operatörer
- If-sats
- While-sats
- Kommentarer



### Omvandling av typer

Omvandling till typerna str, integer och float görs m.h.a följande funktioner:

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

Exempel:

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

3/32



### Kommentarer

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

- Varför och när skriver man kommentarer i ett program?
  - Programkoden själv räcker inte för att göra koden begriplig för en annan programmerare
  - Efter ett par veckor är man själv en annan programmerare!
  - Dokumentera med kommentarer allt som inte är självklart
  - Funktioner/metoder (senare): Dokumentera vad de gör, deras parametrar och funktionens/metodens returvärde

4/32



### Operatörerna + och \*

- Operatör + används för att konkatenera två strängar.  
"ab"+"ba" → "abba"
- Man kan använda \* följt av ett heltal för att upprepa en sträng ett antal gånger.  
"mam"\*2 → "mammm"

5/32



### Minilabb

Skriv ett program som frågar efter användarens ålder och beräknar och skriver ut examensålder.

```
age = input("Hur gammal är du?")
age = int(age)
examinationAge = age + 4.5
print("Du kommer att vara",
      examinationAge,
      "när du tar examen!")
```

6/32



## Jämförelseoperatorer

Följande är operatorer som används för att jämföra värden. De har ett boolskt värde d.v.s. sant (True) eller falskt (False):

```
==
!=
<
<=
>
>=
```

7/32



## If-sats (villkorssats)

En if-sats används för villkorlig exekvering av en eller flera satser.

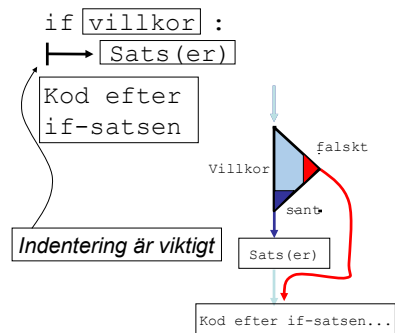
Exempel:

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

8/32



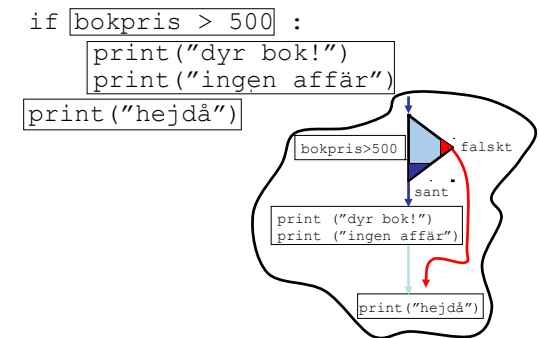
## If-satsens struktur



9/32



## Exempel



10/32



## Exempel

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

11/32



## elif och else

If-satser kan kombineras med elif och else.

Exempel:

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

12/32



## 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

```

13/32

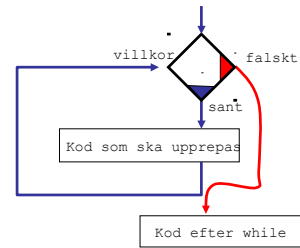


## While-satsens struktur

```

while villkor:
    kod som ska upprepas

```



14/32



## 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

```

15/32



## Logiska operatörer

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

• Jämförelseoperatörer kan kombineras med operatörerna and, or, not

• Exempel:

```

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

```

16/32



## Formatering

a="förkortad diverse:%5.3s"% "diverse"

5 string  
 förkortad diverse: . . . div  
 3

17/32



## Formatering

t="A%11.2f number"%52687.43456

11 float  
 A . . . 52687.43 number  
 2

18/32



## Formatering

```
t="A big%10d number"%52687
```

"digits"

```
A big 10 52687 number
```

19/32



## Funktioner

Exempel:

```
# Compute the square of x
# Parameters:
# x: number
# Returns square of x

def square (x):
    return x * x

y = 3

print ("Kvadraten på", y, "är", square (y))

Kvadraten på 3 är 9
```



## Funktioner

Syntax:

parametrar

```
def funktionensnamn ( ) :
```

Kod som ska exekveras när funktionen anropas

*Indragning är viktigt!*



## Funktioner

- Indata skickas till funktioner via funktionens **formella parametrar** (tex `x` i `def square (x)`)
- Värdet man skickar till funktionen vid ett anrop kallas **argument** (tex `3` i `square (3)`)
- Funktioner returnerar utdata med hjälp av **return-satsen**
- Om en funktion inte har **return-sats** i kroppen kommer funktionen att returnera **None**. (None betyder ingenting i python.)



## Funktioner

- Används för att dela upp ett program i **naturliga** och **återanvändbara** delar
- En funktion tar oftast **indata** och ger **utdata**, men en funktion kan även utföra operationer med bieffekter
- Man kan undvika upprepning av kod genom att använda egna funktioner och parametrar
- Med hjälp av egna funktioner inför man abstraktion i sitt program



## Sammanfattning

- Använd beskrivande variabelnamn
- Var noggrann med datatyper (skilj mellan sträng och tal)
- `input()` används för inmatning
- Planera ditt program innan du börjar skriva kod för det
- If-satser används för att villkorligt köra en eller flera satser

24/32



## 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 innanför varandra kallas det nästlade slingor
- Funktioner används för att dela upp ett program i återanvändbara delar