



Föreläsning 1

Programmeringsteknik och C

DD1316

Mikael Djurfeldt <mdj@kth.se>



Program

```
# Essensen i programmering är att bryta ned
# ett problem i naturliga delar där de minsta
# delarna till slut kan skrivas som satser
# i det programmeringsspråk vi använder

def stekLök ():
    print ("Stek lök i smör")

def stekKöttfärs ():
    print ("Stek köttet")

def koka ():
    print ("Blanda och koka")

stekLök ()
stekKöttfärs ()
koka ()
```



Föreläsning 1

Programmeringsteknik och C DD1316

- Programmeringsteknik – introduktion
- Kursinformation
- Introduktion till UNIX
- Introduktion till Emacs
- Introduktion till Python



Programmering

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

5/32



Italiensk köttfärsrås

- 1) Stek 1-2 hackade lökar i en stor klick smör i medelvarm panna i c:a 4 min (tills löken börjar bli gul).
- 2) Lägg löken i en gryta.
- 3) Stek 400 g köttfärs i smör i het panna tills köttet börjar få brun, knispig, knäckig yta (detta är viktigt för smaken!). Om fryst köttfärs, lägg hela biten i pannan och skala successivt av lager som tinat upp. Om färsk, lägg hela klumpen i mitten av pannan och dela upp den stegvis. (Delar man omedelbart upp hela klumpen över pannan blir temperaturen för låg och kötsaften försvinner ur köttet och kokar bort.)
- 4) Tag pannan av värmen.
- 5) Salta köttet rikligt och dra 70 varv över köttet med en vitpepparkvam.
- 6) Sprid en stor matsked mjöl över köttet och blanda.
- 7) Lägg köttfärsen i grytan.
- 8) Pressa ut nästan en hel tub Arditas tomatpuré i grytan.
- 9) Häll vatten i grytan. Vid omrörning ska innehållet ha en något grölig konsistens.



Programspråk

- Programspråk används av människor för att skriva datorprogram på ett mer begripligt och lätthanterligt sätt (än maskinkod)
- Programspråk tolkas eller översätts till maskinkod av maskinkodsprogram (som i sin tur kan vara *skrivna* i ett programspråk)
- En instruktion som är skriven i ett programspråk måste följa språkets regler för att kunna tolkas eller omvandlas till maskinkod



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?

7/32



Variabelnamn

- Använd beskrivande variabelnamn som talar om vilken roll variabeln har
Bra: `nTurns`, `n_turns`, `numberOfTurns`
Dåligt: `pluttilutt`, `zz45`
- Undvik *alltför* långa namn
- Var konsekvent vid val av variabelnamn (tex antingen genomgående "camelCase" eller "under_score")
- Försök att följa språkets tradition för val av variabelnamn (Ex: I Python brukar ett variabelnamn inledas med gemen)
- Python är ett skiftlägekänligt (case sensitive) språk

10/32



Syntax och semantik

- **Syntax** är programspråkets struktur, dvs vilka element som får förekomma och i vilken ordning.
- Exempel på syntaktiska fel:

```
print "Hej"
print ("Hej)
```
- **Semantik** är innebörden hos ett stycke programkod, tex en sats (i Python ungefär "programrad") eller ett program.
- Exempel på semantiskt fel:

```
print ("Hälften av x är", x/3)
```

8/32



Reserverade ord

- En del ord får inte användas som variabelnamn eftersom de har särskild betydelse för Python. Dessa kallas **reserverade ord**:

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

11/32



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 = "Mikael"
length = 1.87
```



Datatyper

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



Inläsning

- Inläsning från tangentbordet görs m.h.a funktionen

```
input()
```

Datorn läser inmatade tecken tills användaren trycker på "Enter" och resultatet blir en sträng med dessa tecken

- Om man ger en textsträng som argument till input skrivs denna ut som prompt innan input väntar på inmatning:

```
input("Vad heter du?")
```

argument

13/32



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