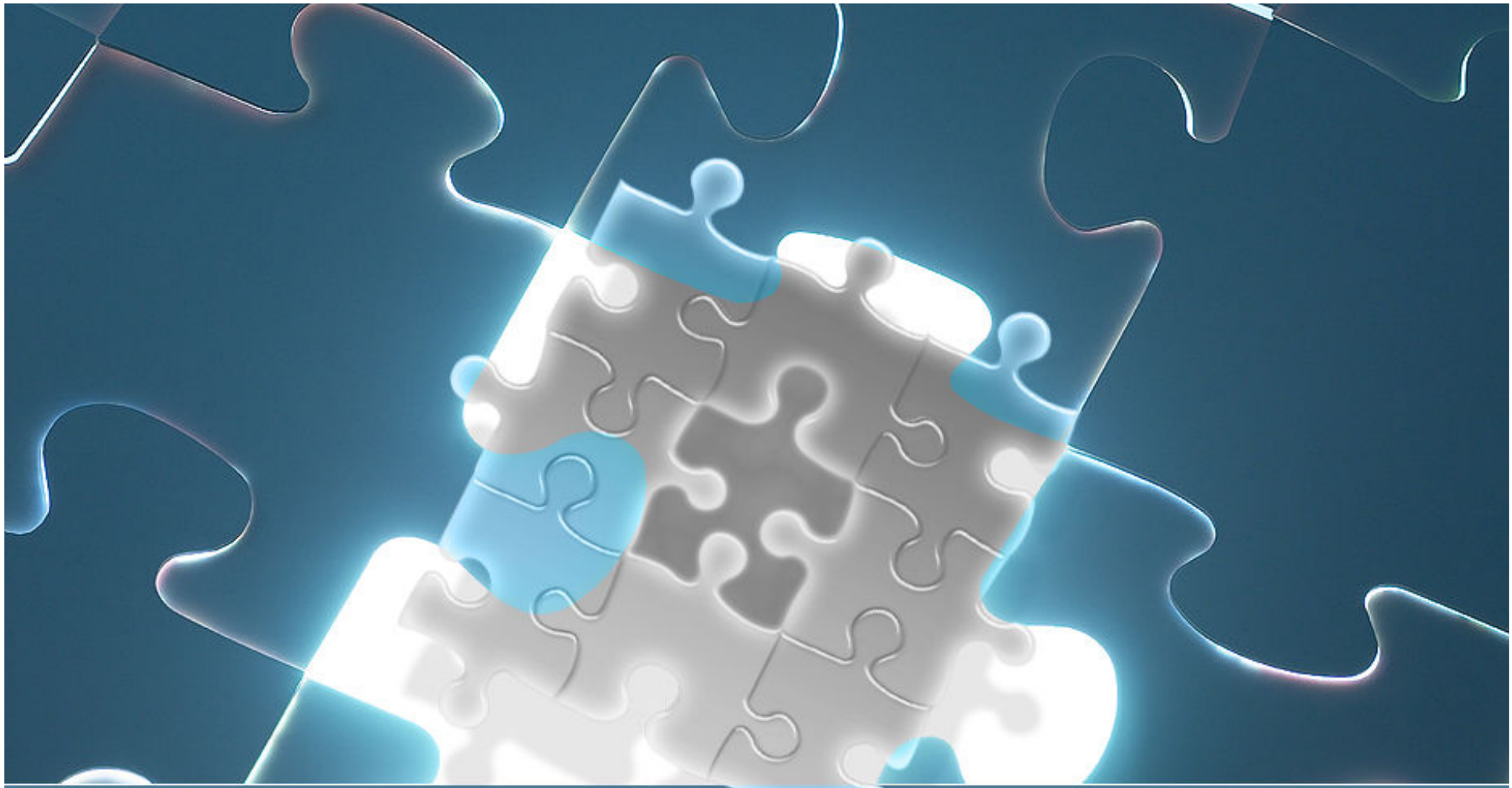


Grundkurs i programmering - intro

Linda Mannila 4.9.2007

Dagens föreläsning

- Allmän kursinformation: mål, syfte, upplägg, examination, litteratur, etc.
- Hur arbetar en dator?
- Hur vi får datorn att förstå oss och lösa de problem vi vill?
- Lite om programmeringsspråk
- Introduktion till Python

Kursen i ett nötskal

- Kräver inga förkunskaper, vi börjar från början
- Innehåll
  - Grundläggande programstrukturer
  - Planering och design av program
  - Problemlösning
  - Testning och felsökning
  - Text- och filhantering
  - Hantering av undantag

Kursens mål

- Efter kursen kommer du att
  - Ha lärt dig att tänka som en informationsbehandlare/datatekniker
  - Förstå idén bakom programmering
  - Kunna planera, designa och skriva program – även ”coola” sådana
  - Klara av att testa och debugga kod
  - Kunna läsa och förstå kod som andra har skrivit

Men du kommer inte att vara en fulländad programmerare... Sorry ☺

Kursens uppläggning

- Föreläsningar
- Räkneövningar
- Skriftlig tent

Föreläsningar

- Period 1 (v. 36-42)
  - Tisdag 8.30 – 10.00
- Period 2 (v. 44-49)
  - Tisdag 8.30 – 10.00
  - Torsdag 8.30 – 10.00
- Sista föreläsningen 4.12
- Auditorium Gamma

Räkneövningar

- Obligatoriska
- Sex stycken á 10 poäng
- Görs i datorklass under handledning
- Lämnas in för korrigering
- För att få tenträtt måste du göra *minst 50% av varje räkneövning* (dvs. totalt 30 poäng)
- Satsa på räkneövningarna
  - Ni lär er massor!
  - Pluspoäng i tenten!

Tent

- Med penna och papper
- Maxpoäng 30, 15 krävs för godkänt
- Kurstenter (OBS! Fel i tryckta utbprog)
  - Nr 1: fre 14.12.07
  - Nr 2: fre 18.01.08
- Pluspoäng från räkne-övningarna till *godkänt* tentresultat

RÖ	Bonus
30-38	0
39-44	1
45-50	2
51-60	3

Kurssida

http://www.abo.fi/~linda.mannila/proggk07

Down to business... föreläsning #1

Vad är en dator?

- ”a machine that stores and manipulates information under the control of a changeable program”

Vad är en dator?

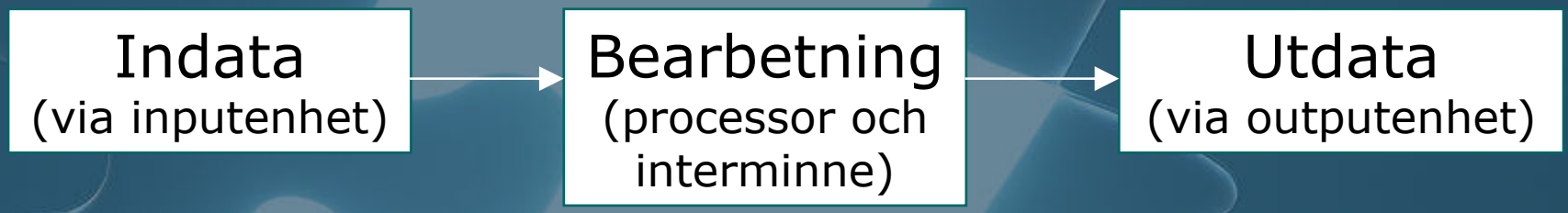
- ”a machine that stores and *manipulates information* under the control of a changeable program”
  - Indata → Utdata
  - Inte unikt för datorer (räknemaskin, bensinpump, mikrovågsugn, ...)

Vad är en dator?

- ”a machine that stores and manipulates information under the control of a *changeable program*”
  - Skiljer datorer från bensinpumpar etc
    - Datorer kan göra olika saker beroende på vilket program de kör
    - En bensinpump kan inte mitt i allt börja fungera som en Nintendo Gameboy eller en mikrovågsugn som en texteditor → byggda för att utföra en enda uppgift

Vad är en dator?

- *Processorn* (CPU) bearbetar information (data)
- *Internminnet* (RAM) lagrar program och data
- *Input/output*-enheterna (IO) kommunicerar med omvärlden, t.ex. tangentbord, skärm, hårddisk, skrivare, etc.



Vad är en dator?

- *Hårdvara*: Fysiska delar i datorn (skärm, CPU, etc.), elektronik och sladdar
- *Mjukvara*: Serie av instruktioner till CPU som styr hårdvaran så att den löser något problem. Program.
- Mjukvaran styr hårdvaran

Programmering

- Att skapa mjukvara
- Kräver en förmåga att se större helheter samtidigt som man är petnoga med alla detaljer

Varför lära sig programmera?

- Väsentlig del av informationsbehandling/ datateknik → viktig för alla som vill bli IT-proffs
- Men även nyttigt för andra
  - Icke-programmerare känner sig ofta utan kontroll då det kommer till datorer
  - Programmerare har kontroll
- Utmanande men väldigt roligt!
- Utvecklar andra färdigheter, t.ex. problemlösning
- Industrin behöver programmerare

Programmeringsspråk

- Datorn förstår enbart ettor (1) och nollor (0)
- En dator förstår *inte* tvetydigheter eller underförstådda saker
- En dator gör *ingenting* automatiskt!
- Programmerare (vi) måste instruera datorn till allt
- Kommunikationen av instruktioner måste vara precis och exakt (som ettor och nollor)

Programmeringsspråk

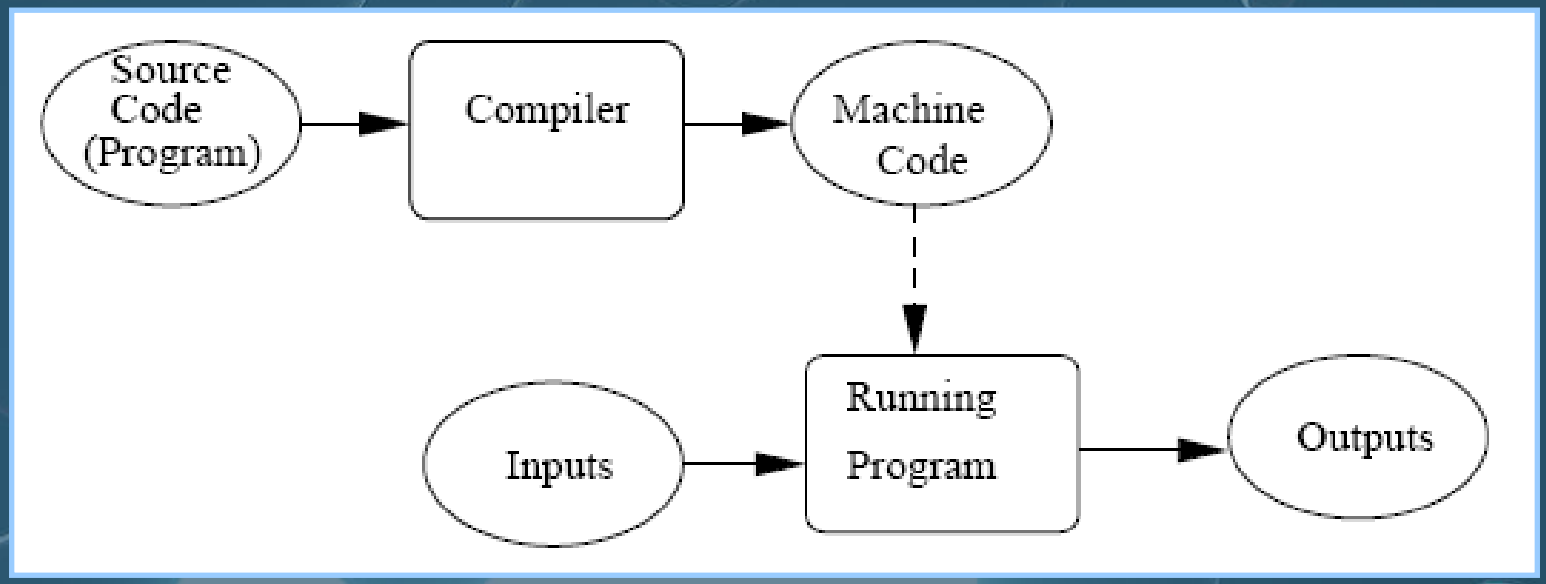
- Naturligt språk?
- Programmeringsspråk
  - Exakt notation
  - Syntax: format
  - Semantik: mening, betydelse

Olika typer av programmeringsspråk

- Maskinspråk
  - Bara ettor och nollor
  - Maskinberoende
- Assemblerspråk (lågnivåspråk)
  - Maskinnära
  - Maskinberoende
- Högnivåspråk
  - Måste översättas för att datorn skall förstå koden
  - Maskinoberoende
  - Kompilerade eller tolkade

Kompilerade språk

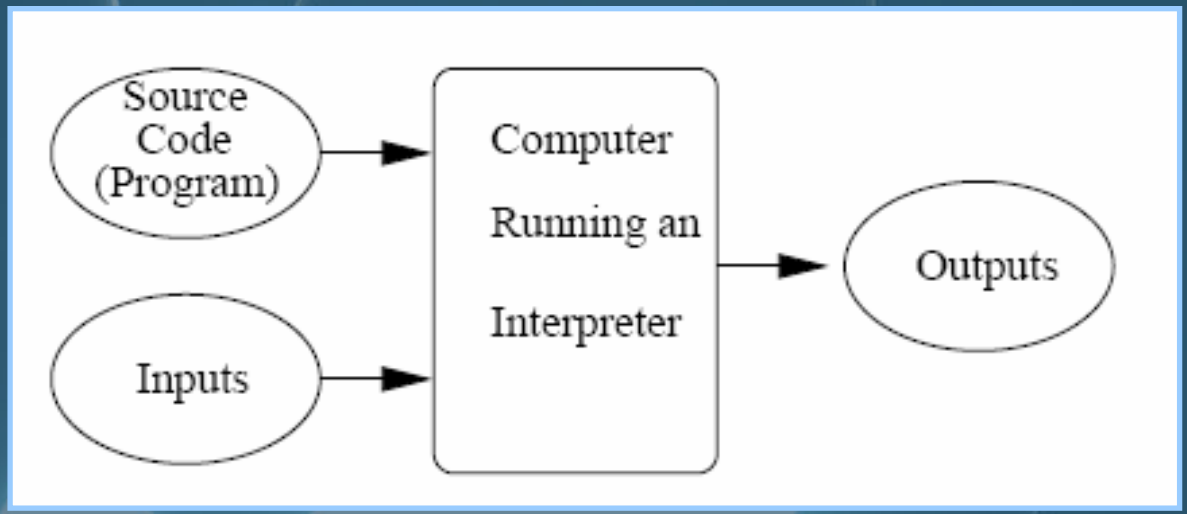
- Kompilator
  Ett program som översätter ett program i ett högnivå-språk till motsvarande program i maskinspråk för en given dator



Tolkade språk

- Tolk

  Ett program som ”låtsas” att datorn förstår högnivåspråk. Analyserar och exekverar källkoden en instruktion i taget – inte hela programmet på en gång som en kompilator.



Kompilerade vs. tolkade språk

- Kompilerade språk
  - Program översätts fullständigt på en gång
  - Ett kompilerat program kan köras om och om igen utan att man behöver blanda in kompilatorn eller källkoden
  - Snabbare program
- Tolkade språk
  - Program översätts steg för steg
  - Tolken och källkoden behövs varje gång programmet skall köras
  - Flexiblare programutveckling, program kan köras interaktivt

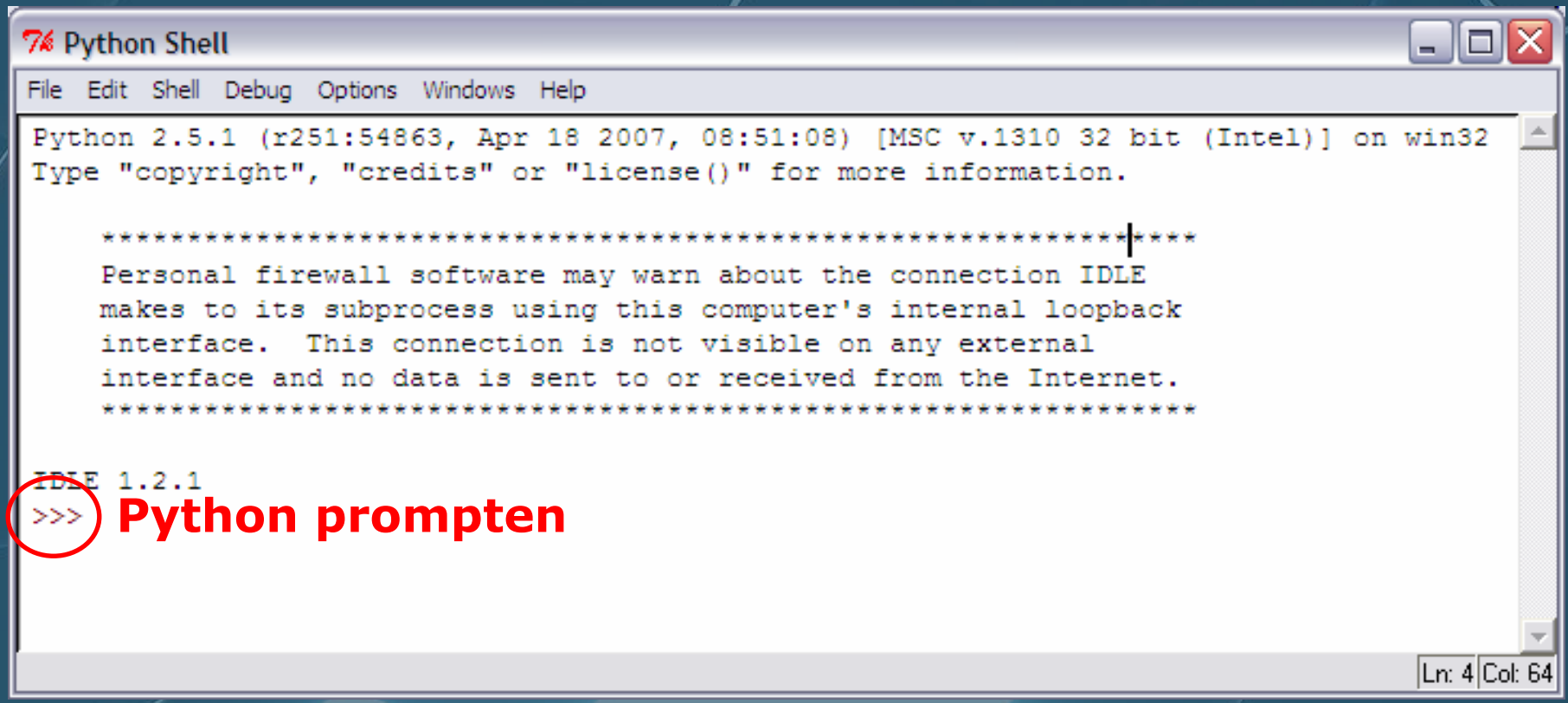
Programmeringsspråk

- Fortran (1950-talet)
- COBOL (1959)
- Basic (1964)
- Pascal (1970)
- C (1970-talet)
- C++ (1983)
- Python (1991)
- Java (1995)

Python

- Tolkat skriptspråk
- Allt mer populärt i industrin
  - Google, Yahoo, NASA, Compaq, Philips,...
- Förinstallerat på Mac och Linux (kolla versionen!)
- Ladda ner från http://www.python.org/download
- Senaste versionen 2.5.1

IDLE - Integrated DeveLopment Environment



Python prompten

Interaktivt läge

- Perfekt för att testa enskilda satser / korta programsnuttar
- Python prompten >>>
  - anger att interaktivt läge körs
- Exempel...
  - `>>> print "Hello World"`
  - `>>> print 2+3`
  - `>>> print "2+3 =", 2+3`

Skriptläge

- Kod som skrivs i det interativa läget lagras inte
  - allt försvinner då tolken stängs
  - måste skrivas in på nytt om vi vill köra samma program igen
- För att spara program skrivs de i en textfil (.py) som sparas
  - modul / skript
- Kan sedan köras om och om igen, när som helst
- Exempel...

Inför nästa vecka

- Installera Python
- Sök upp kursböckerna på nätet
- Provkör Python...

Provkörning

Starta Python i interaktivt läge och kör följande satser – en i taget.
Vad blir resultatet? Varför?

a) `print "Att programmera är kul"`
b) `print 3.0`
c) `print "15 i kvadrat är" , 15 * 15`
d) `print 2.0 + 3.0`
e) `print "2" + "3"`
f) `print 2 * 3`
g) `print 2 ** 3`
h) `print 2 / 3`
i) `print 2.0 / 3.0`