

Programmering för alla!

Inspirationsseminarium för lärare i
grundskola och gymnasium

Björn Regnell

Professor

Datavetenskap, LTH, Lunds universitet

lth.se/programmera



<http://www.svt.se/nyheter/sverige/krav-pa-att-elever-lar-sig-programmera>

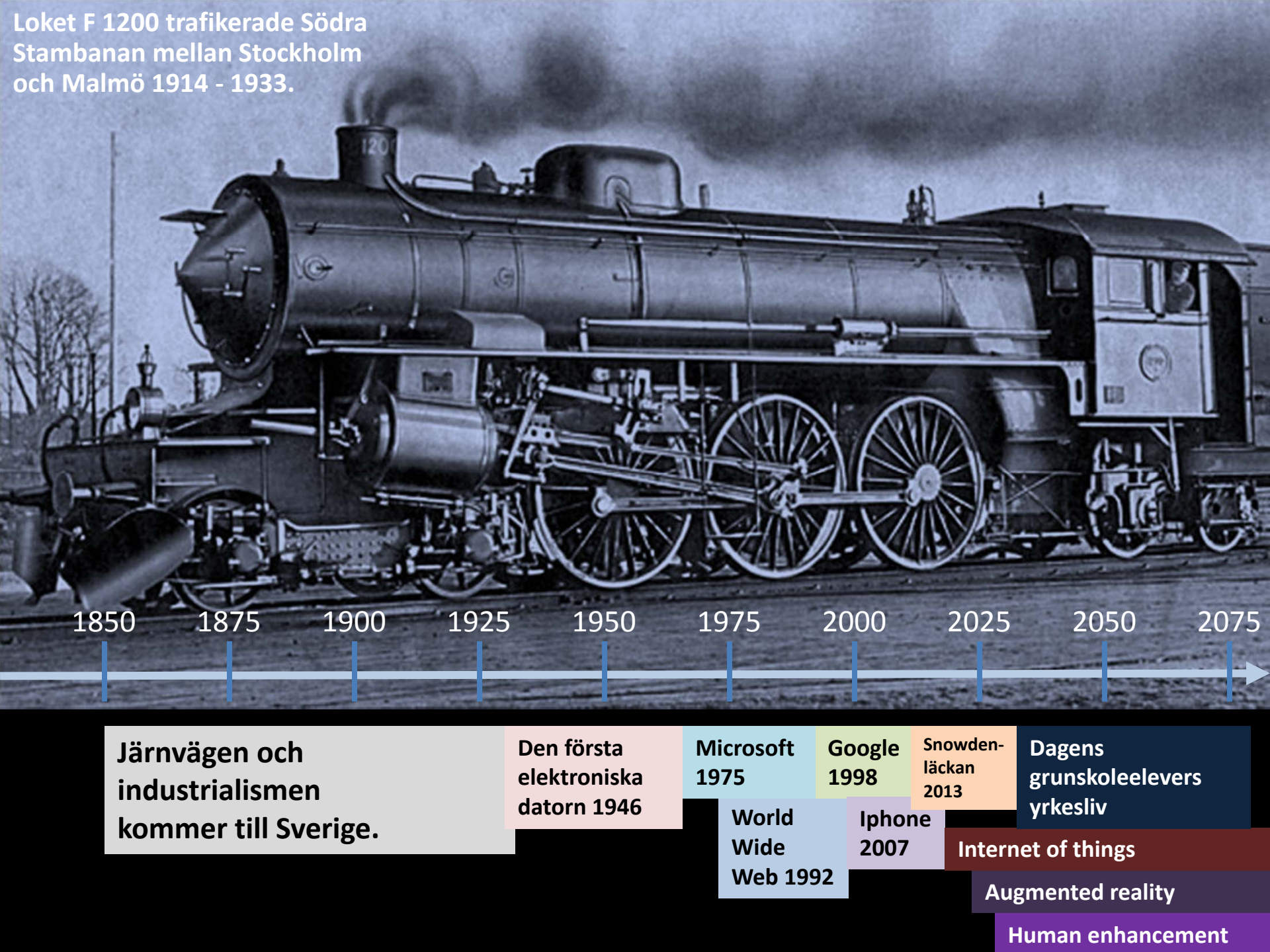


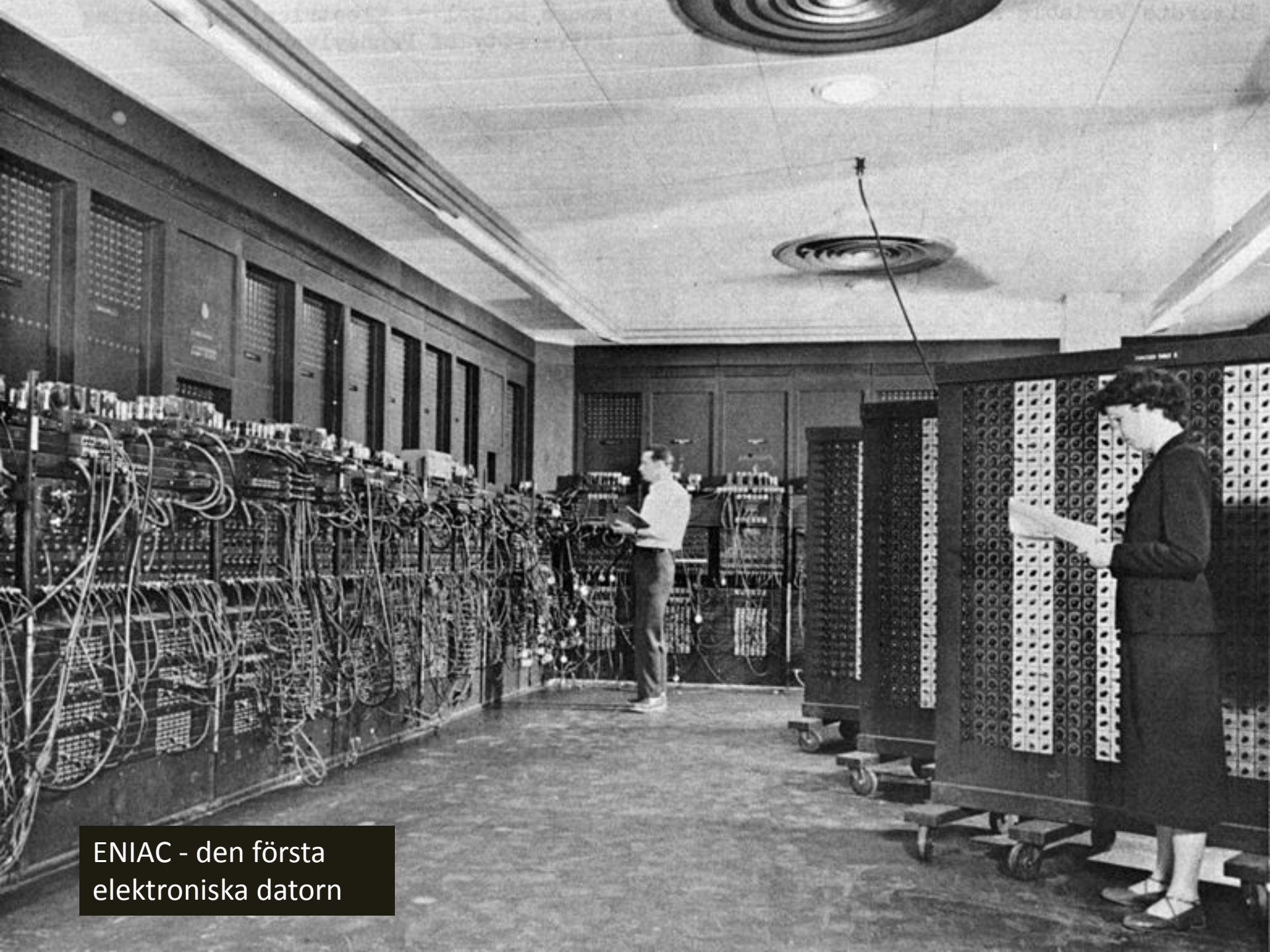
Lärandemål

1. Verktyg och inspiration att komma igång med programmering tillsammans med dina elever
2. Grundläggande förståelse för några viktiga begrepp inom programmering
3. Praktisk kunskap att skriva och starta egna program i en enkel programmeringsmiljö
4. Idéer om hur programmering kan användas i din undervisning
 - Matematik, svenska, engelska, SO, NO, teknik, ...

Varför programmering för alla?

Loket F 1200 trafikerade Södra Stambanan mellan Stockholm och Malmö 1914 - 1933.





ENIAC - den första elektroniska datorn

När jag var liten ...

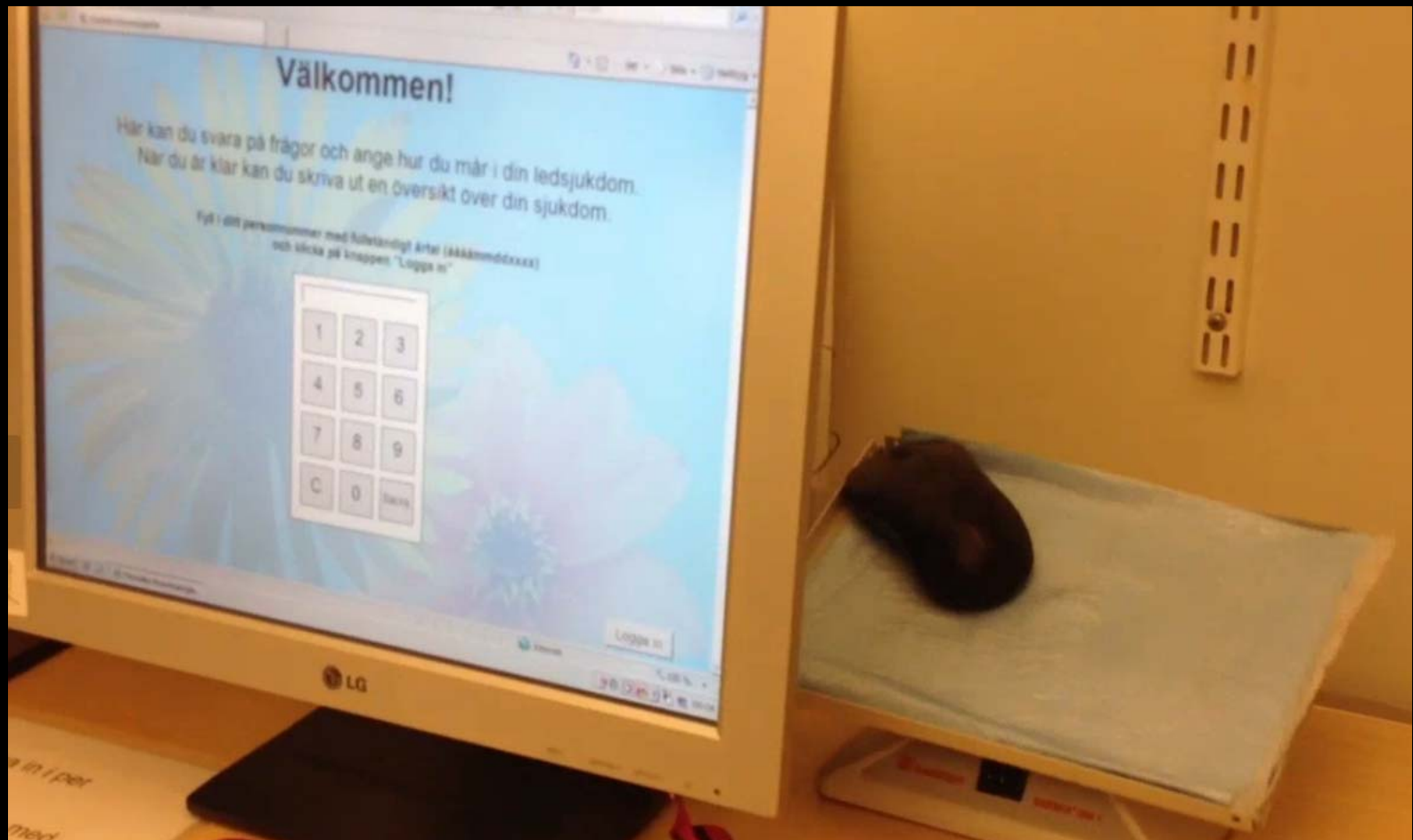


```
10 print "hej"  
20 goto 10
```



- kommunikation
- transport
- byggsektorn
- statsförvaltning
- finanssektorn
- media
- sjukvård
- övervakning
- integritet
- upphovsrätt
- miljö & energi
- sociala relationer
- utbildning

Var finns kod inte?



<http://javlaskitsystem.se/2012/09/dagens-system-varfor-vaggar-doktorn-musen/>



Google



DEMOKRATI & JÄMLIKHET kräver ALLMÄNBILDNING

alla behöver grundkunskaper i
datalogi och programmering för att
aktivt kunna delta i samhällsutvecklingen



Ada Lovelace

(1815 - 1852)

Världens första programmerare



Grace Hopper

Uppfann världens första kompilator 1952

Varför ska **du** lära dig programmera?

- Det är kul att själv skapa kod!
- Om du kan koda är det lättare att förstå datorn.
- Hela världen är beroende av att kod finns och fungerar på önskvärt sätt.
- Alla behöver förstå datorer för att kunna vara med och besluta om vår framtid!

Varför ska **barn** lära sig programmera?

- Mjukvara ligger bakom den största samhällsomvandlingen genom tiderna och det är våra barn som ska forma framtidens samhälle
- Det råder stor obalans mellan olika grupper i samhället vad gäller tillgången till kunskap i datalogi och skolan är en viktig utjämningsfaktor
- Det är stor skillnad på att enbart vara passiv IT-konsument och att aktivt skapa koden som styr IT



Seymour Papert
uppfann 1967 programspråket
LOGO för att barn med hjälp av
programmering ska kunna utveckla
sin problemlösningsförmåga.

Var med och utvecklade den
pedagogiska teorin om
"constructionist learning":
Genom eget skapande konstruerar
eleverna mentala modeller för att
kunna förstå världen omkring dem.

Varför ska **lärare** lära sig programmera?

- Programmering kan användas för att levandegöra och **integrera** olika ämnen, så som matematik, NO, SO och språk.
- Barnen kan ofta mer som användare av mjukvara än många vuxna, men de behöver **hjälp av vuxna** för att komma igång med programmering och att skaffa sig en djupare förståelse av mjukvarans roll i samhället
- Genom programmering kan barn och lärare **tillsammans** utveckla programmeringens roll i lärandet
<http://viprogrammerar.se/>

Exempel på "eviga" färdigheter som utvecklas med datalogi och programmering

- Tänka logiskt
- Tänka steg-för-steg
- Tänka i abstraktioner
- Konstruera abstraktioner
- Skriva exakt – allt du skriver har en effekt
- Fantasi och kreativitet
- Förklara för varandra
- Samarbete

"Computational thinking"
Datalogiskt tänkande

Exempel på pedagogiska möjligheter med programmering i skolan

- De som ännu inte har svenska språket som ett starkt verktyg för att kommunicera kan använda programmering i sin **språkutveckling**
- De som inte har så lätt för skolans fokus på "läs-uppgift-skriv-svaret" får här ett alternativ med metoden "**gör-själv-och-upptäck-vad-som-händer**"
- Programmering visar att de med kunskap att styra maskinen också har makt över människor och den konkreta upplevelsen av att själv koda öppnar upp diskussionen om **kunskapens roll i samhället**

Det finns MASSOR av system
och pedagogiska koncept
för att lära barn programmera.

Vilket ska jag välja???

Om Scala och Kojo

- Scala är ett modernt och kraftfullt programspråk med det allra senaste inom dataforskningen.
- Scala är ett globalt öppen-källkodsprojekt som leds av Prof. Martin Odersky med forskningsmedel från EU.
- Kojo är en utvecklingsmiljö för Scala som gör det lätt att komma i gång med programmering.
- Kojo är ett globalt öppen-källkodsprojekt som leds av Lalit Pant i Indien. LTH är bidragsgivare (pengar, översättning och kodning).

Mål med LTH-projektet

”Programmering för alla!”

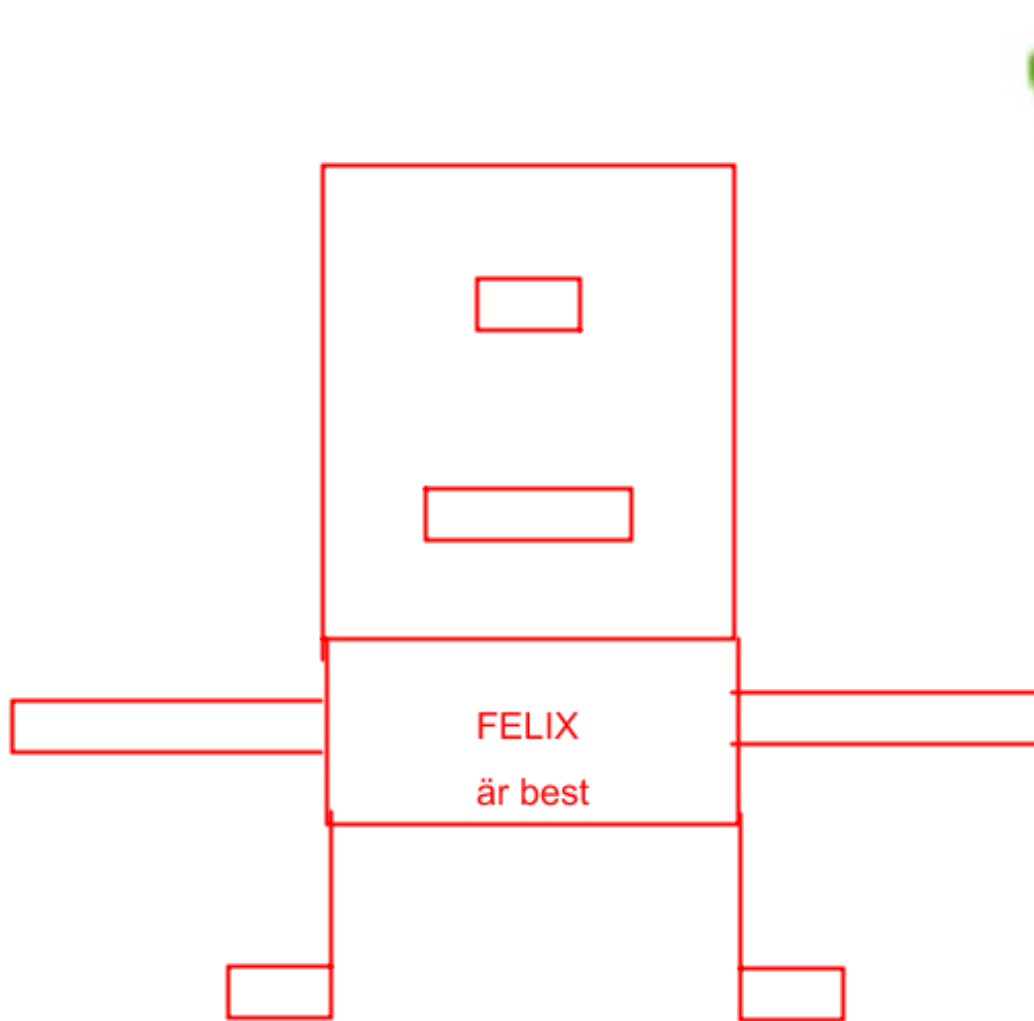
- Så enkel start som möjligt: låg tröskel
- Ett riktigt programmeringsspråk: inget tak
- Roliga programmeringsuppdrag för barn och ungdomar från förskolan och uppåt
- Lärarutbildning som sprider programmeringsglädjen till skolungdomar

LTH:s läromaterial för programmering i skolan

- En serie uppdrag som kan anpassas
- Använda abstraktioner
- Skapa egna abstraktioner

lth.se/programmera

Rita med sköldpaddan i Kojo



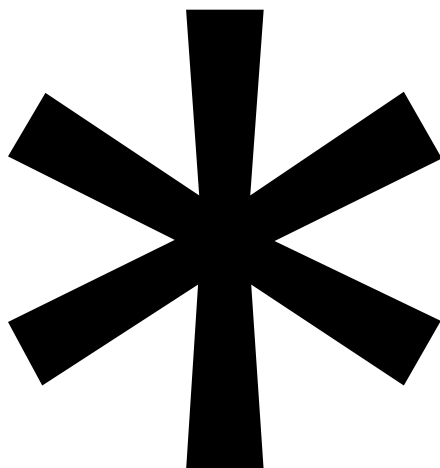
MAMMA ÄR BEST!!!

PAPPA ÄR BEST!!!

RASMUS ÄR BEST!!!

Hur funkar en dator?

42



Vad finns i datorn?

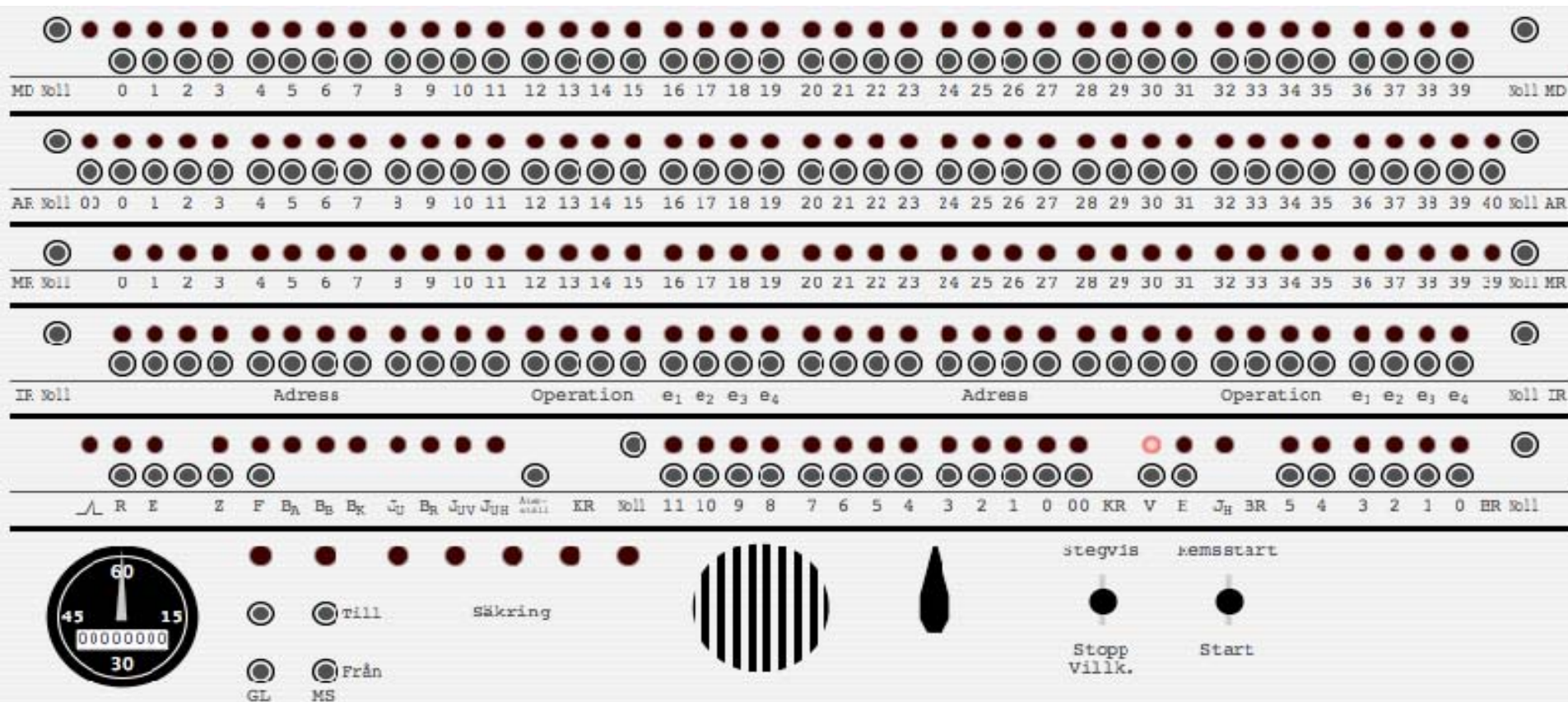
Minnesadress	Innehåll
0	76
1	196
2	82
3	32
4	68
5	73
6	71
7	32
8	80
9	82
10	79
11	71
12	82
13	65
14	77
15	77
16	69
17	82
18	65

Heltal och endast heltal!



SMIL – siffermaskinen i Lund

Sveriges första dator skapades 1953 av Carl-Erik Fröberg. Smil kom tre år senare och konstruerades utifrån samma grundritningar. Datorn kostade 120 000 kronor att bygga, vilket motsvarar cirka två miljoner i dagens penningvärde. Smil krävde ett tio meter långt rum. Minnet var på 40 KB.



Hur funkar en dator?

Minne
sparar
både
data och
program



Läs & Skriv



Processor
bearbetar
data
enligt
program



Skriv ut



**Utdata-
enhet**
data till
omvärlden

Läs in



**Indata-
enhet**
data från
omvärlden

Uppdelning: data / instruktioner

- Heltal används för att representera data: bokstäver, bilder, musik, filmer, etc
- Heltal används för att representera instruktioner (program) som anger vad datorn ska göra: addera, multiplicera, visa på skärmen, etc
- Datorn hjälper till att översätta (för människor) mer lättbegripliga instruktioner till rätt siffror
- I datorns minne finns BÅDE data OCH instruktioner

Data och program är ettor och nollor

Minne
med data
och
program
som *ettor*
och *nollor*

```
0001010010010
0101001010010
1001010010100
1010111110100
1110100110101
1111000010101
```

Läs
&
Skriv

Processor
bearbetar
ettor och
nollor

CPU
Central
Processor
Unit

0 -> 00000000
1 -> 00000001
2 -> 00000010
3 -> 00000011
4 -> 00000100
5 -> 00000101

...

Skriv ut ettor och nollor

0001011011100

Utdata

Läs in ettor och nollor

11011100

Indata

A -> 01000001
B -> 01000010
C -> 01000011

...



Hur många ettor och nollor?

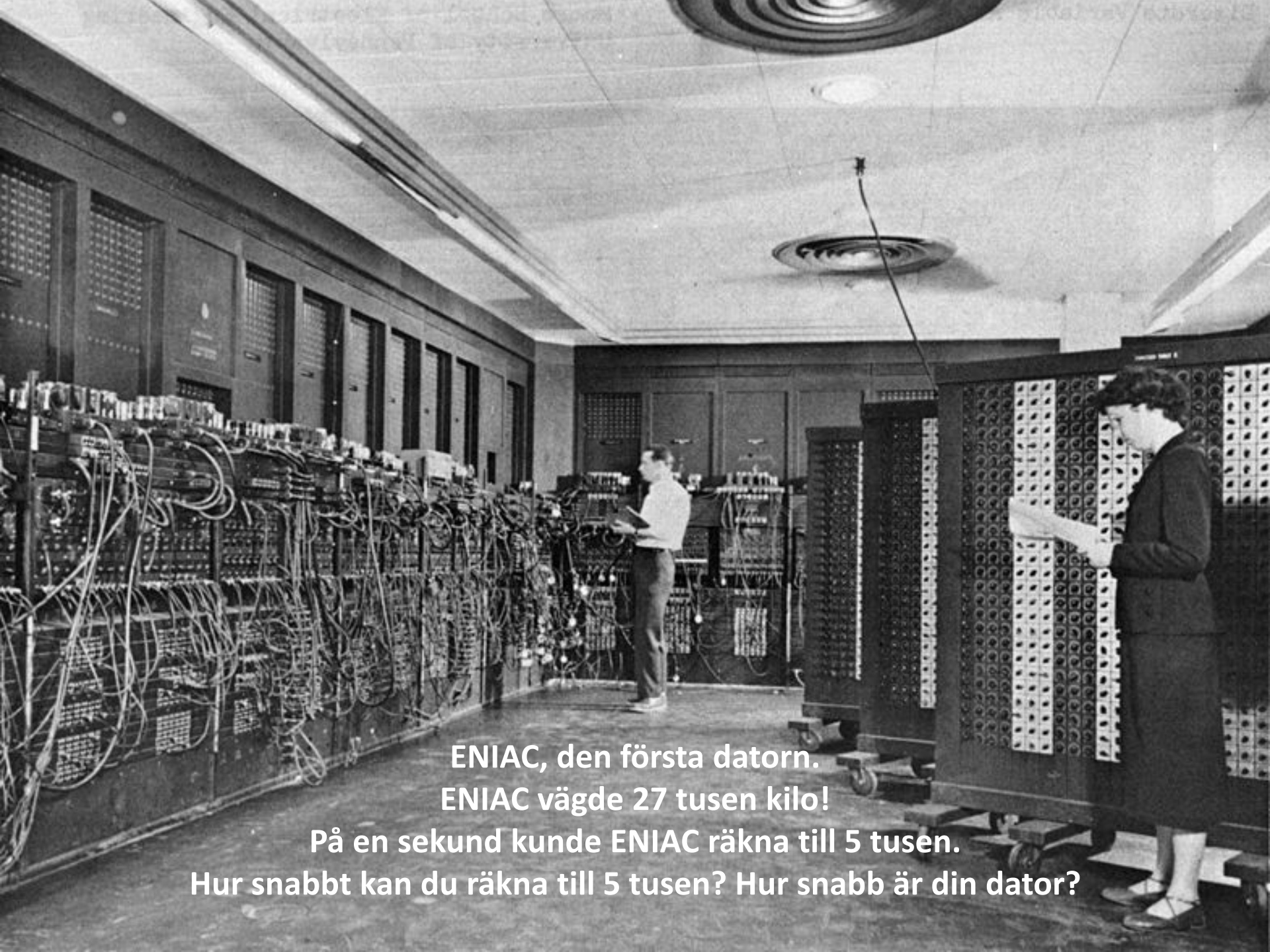


- En binär siffra kallas **bit**
- En **bajt** (eng. byte) förkortat B motsvarar 8 bitar
- En bajt med det binära värdet 01000001 tolkas t.ex. som siffran 65 eller bokstaven A
- 1 MB = 1 miljon B
- 1 GB = 1 miljard B = 1000 MB

Vad är programmering?

Vad är programmering?

- Att programmera är att bestämma vad datorn ska kunna göra.
- Ett program innehåller **koder** som datorn kan förstå.
- Koderna innehåller **instruktioner** som datorn kan utföra och **data** (information) som datorn kan använda när den utför sina instruktioner.
- Ett program är "datorns recept"
- När man gör nya program bygger man på färdiga delprogram som "byggblock" till nya program



ENIAC, den första datorn.
ENIAC vägde 27 tusen kilo!

På en sekund kunde ENIAC räkna till 5 tusen.
Hur snabbt kan du räkna till 5 tusen? Hur snabb är din dator?

Programmering förr och nu

- De första datorerna programmerades genom att människor själva kopplade om sladdar och knappar, en för varje etta eller nolla



Programmering förr och nu

Nu finns program som hjälper människor att översätta program och data till representationer som passar människor (text, bilder, ljud)

Program representeras oftast med ett textuellt språk med en entydig grammatik som översätts till ettor och nollor av ett program som kallas **kompilator**



Källkod

(lättare för människor)



kompiator



Maskinkod

(siffror som styr datorn)

Vad är datorer bra på?

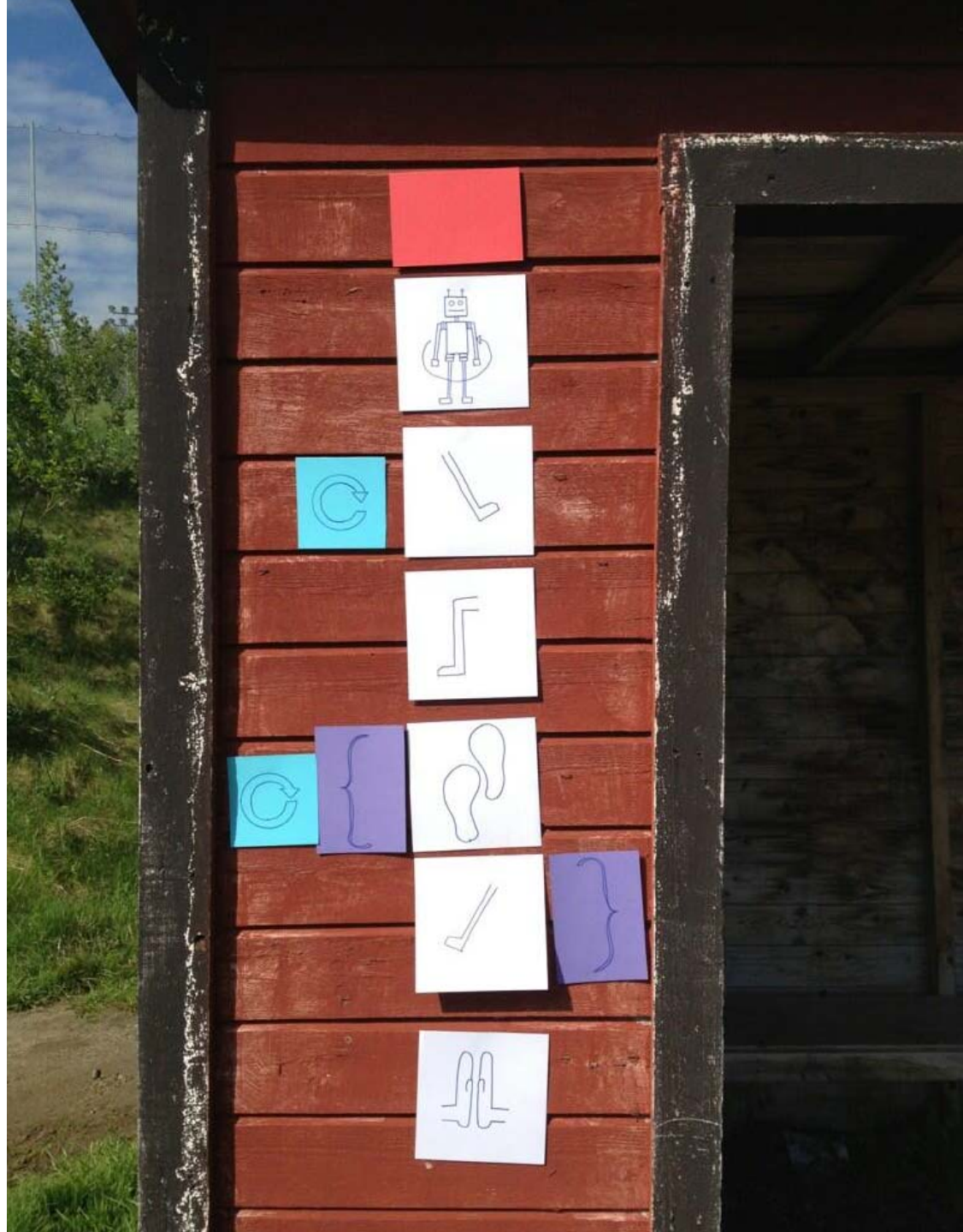
- Datorn är otroligt bra på att räkna snabbt.
- Datorn kan lätt göra samma sak många gånger.
- Datorn är **dålig** på att hitta på själv.
- Om du hittar på något nytt du vill att datorn ska kunna, måste du programmera i detalj varje steg som datorn ska utföra.
- Som tur är finns det redan en mängd färdiga delprogram med användbara funktioner som man kan sätta samman till nya program
- Ett gränssnitt till användbara funktionsbibliotek kallas API (Application Programming Interface)

Några datalogiska grundbegrepp

- **Algorithm:** beskrivning hur man löser ett problem (ett slags [recept](#))
 - **Sekvens:** göra saker steg för steg
 - **Repetition:** upprepa vissa steg många gånger
 - **Alternativ:** välja nästa steg beroende på variablers värde
- **Abstraktion**
 - **Funktion:** en del av ett program som har fått ett namn och som kan återanvändas gång på gång
 - **Parameter:** ger olika värden som påverkar vad funktionen ger för resultat
 - **Variabel:** plats i datorns minne för värden som kan ändras
 - **Objekt:** en del av ett program som sammanför och kapslar in funktioner och variabler
 - **Klass:** kod som beskriver många objekt av samma **typ**
- Uppdragen i Kojo börjar med:
sekvens, repetition, funktion, parameter

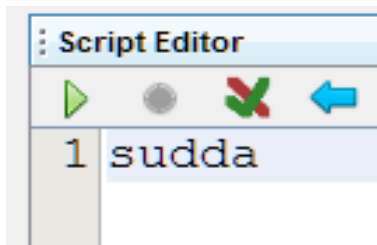
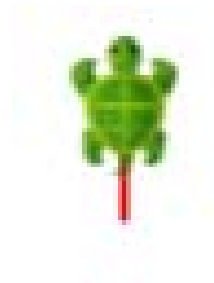
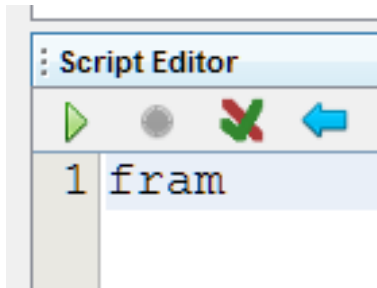
Programming UNPLUGGED

<http://teacherhack.com/programming/dansprogramming/>



Ditt första program!

fram



fram
fram
fram
fram

Paddan kan vrida sig!

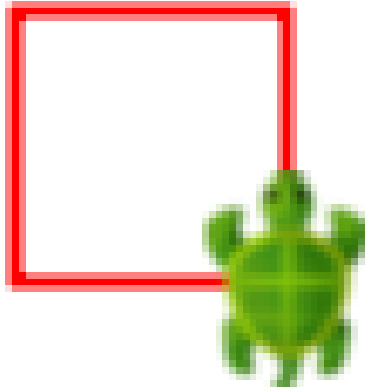
vänster



höger



Rita en kvadrat!



- Tips:
använd **fram** och **vänster**

Nästa steg

- Hemuppgift för lärare:
 - Jobba vidare med [uppdragen](#) gärna tillsammans
 - Gör ett eget nytt uppdrag
 - Prova på elever
 - Skriv en kort reflektion och dela med dig till alla:
 - <http://wiki.kogics.net/forum/c-979648/kojo-teaching-sweden>