



# DATALOGISK TÄNKANDE OCH PROGRAMMERING

CETIS  
OKTOBER 2017

*Tom Tits  
Experiment*

# VAD ÄR DET FÖRSTA NI TÄNKER PÅ?

”Datalogiskt tänkande”  
”Programmering”



*Experiment*



*Tom Tits  
Experiment*

## FRÅGOR I FOKUS IDAG

- Varför är digitala kompetenser och programmering viktigt idag?
- Vad är datalogiskt tänkande och programmering?
- Vilka övningar skulle vi kunna göra för att öva Digitala kompetenser och Datalogiskt tänkande i verksamheten?



*Tom Tits  
Experiment*

## SKOLVERKET

- Nationell digitaliseringsstrategi för skolan
- Digitala kompetenser skrivs in i grundskolans samtliga ämnen och programmering i Ma & Teknikämnet
- All undervisande personal F-9 kommer behöva fortbildning i detta.

### Vision att nå 2022

Alla barn och elever har utvecklat adekvat digital kompetens. Skolväsendet präglas av att digitaliseringens möjligheter tas tillvara så att de digitala verktygen och resurserna bidrar till att resultaten förbättras och verksamheten effektiviseras.

# FÖRMÅGOR ATT UTVECKLA I FÖRSKOLAN

- att tänka kritiskt
- att lösa problem
- att vara nyfiken och kreativ
- att ta eget initiativ
- att ge och följa stegvisa instruktioner
- att tänka logiskt och matematiskt
- att kommunicera och samarbeta

Digital kompetens utgörs av i vilken utsträckning man är förtrogen med digitala verktyg och tjänster samt har förmåga att följa med i den digitala utvecklingen och dess påverkan på ens liv. Digital kompetens innefattar:

- kunskaper att söka information, kommunicera, interagera och producera digitalt
- färdigheter att använda digitala verktyg och tjänster
- förståelse för den transformering som digitaliseringen innebär i samhället med dess möjligheter och risker
- motivation att delta i utvecklingen.

Få syn på digitaliseringen i grundskolan-

Ett kommentar material för förskoleklass, fritidshem och

grundskoleutbildning

Skolverket

**ENTREPRENÖRI  
ELLT LÄRANDE**

# NYCKELKOMPETENSER FÖR LIVSLÅNGT LÄRANDE

- Kommunikation på modersmålet.
- Kommunikation på främmande språk.
- Matematiskt kunnande och grundläggande vetenskaplig och teknisk kompetens. (STEM)
- Digital kompetens. (IKT)
- Lära att lära.
- Social och medborgerlig kompetens.
- Initiativförmåga och företaganda. (Entreprenöriella förmågor)
- Kulturell medvetenhet och kulturella uttrycksformer.

Källa: EU



STIFTELSEN FÖR STRATEGISK FORSKNING

*Vårtannat jobb  
automatiseras  
inom 20 år*  
- utmaningar för Sverige



**MEN DET ÄR OMÖJLIGT  
ATT VETA EXAKT VAD SOM  
KOMMER ATT BEHÖVAS**



## YRKEN SOM BLIR KVAR

Yrken som kräver:

- Fingerfärdighet
- Originalitet
- Konstnärlighet
- Social förmåga
- Förhandling
- Förmåga att övertala
- Omtanke om andra människor

har lägst sannolikhet att ersättas.



# DIGITALA KOMPETENSER ÄR VIKTIGT! (FILM)



[http://www.ted.com/talks/mitch\\_re  
snick\\_let\\_s\\_teach\\_kids\\_to\\_code](http://www.ted.com/talks/mitch_renwick_let_s_teach_kids_to_code)

<https://code.org/>

*Tom Tits  
Experiment*

# DEN STORA UTMANINGEN



Det finns en stor klyfta mellan teknik som finns tillgänglig, problemen vi står inför i samhället och vad vi använder teknik att göra.



*Tom Tits  
Experiment*



# DATALOGISKT TÄNKANDE OCH PROGRAMMERING

*Tom Tits  
Experiment*

## FRÅGOR I FOKUS IDAG

- Varför är digitala kompetenser och programmering viktigt idag?
- Vad är datalogiskt tänkande och programmering?
- Vilka övningar skulle vi kunna göra för att öva Digitala kompetenser och Datalogiskt tänkande i verksamheten?



*Tom Tits  
Experiment*

## DATALOGISKT TÄNKANDE

- Att tänka på problem på ett sådant sätt som gör att datorer kan lösa dem.
- Datalogiskt tänkande är sådant som människor gör, inte datorer.
- Det inkluderar logiskt tänkande och förmågan att känna igen **mönster**, tänka i **algoritmer**, bryta ner problem och sortera bort detaljer (**abstraktion**).



## MÖNSTERIGENKÄNNING

- Att se mönster
- Mönsterigenkänning kallas det när man letar efter likheter och mönster för att lösa utmanande problem på ett bättre sätt.
- För att hitta sådana mönster tittar man på saker som är lika (eller nästan lika) för varje problem



*Tom Tits  
Experiment*

# ALGORITMER

- Ett svårt ord för en enkel koncept
- "systematisk procedur som i ett ändligt antal steg anger" - NE

```
if ($(window).scrollTop() > header1_initialDistance) {  
  if (parseInt(header1.css('padding-top'), 10) > header1_initialPadding) {  
    header1.css('padding-top', '' + $(window).scrollTop() - header1_initialDistance);  
  } else {  
    header1.css('padding-top', '' + header1_initialPadding);  
  }  
}  
  
if ($(window).scrollTop() > header2_initialDistance) {  
  if (parseInt(header2.css('padding-top'), 10) > header2_initialPadding) {  
    header2.css('padding-top', '' + $(window).scrollTop() - header2_initialDistance);  
  } else {  
    header2.css('padding-top', '' + header2_initialPadding);  
  }  
}
```

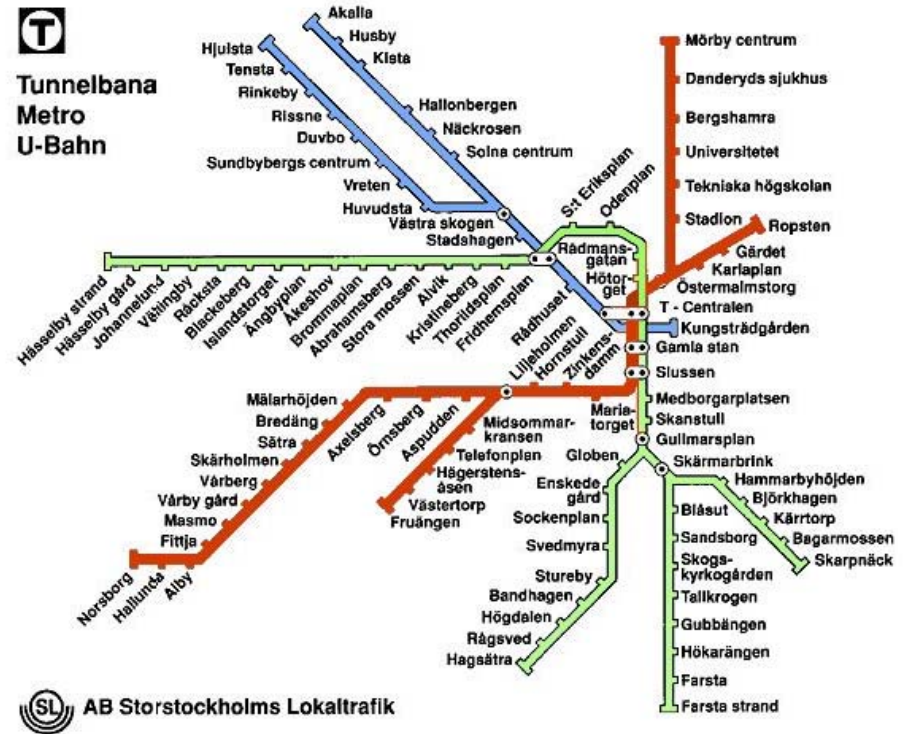


Tom Tits  
Experiment



# ABSTRAKTION

- När man abstraherar något bortser eller döljer man detaljer som är onödiga för den uppgift som ska lösas
- Ett annat ord för detta är att förenkla
- En tunnelbanekarta t ex är en abstraktion av den verkliga, krångliga världen. En kalender är en abstraktion av din tid. Till och med programmeringsspråk är abstraktioner



Tom Tits  
Experiment

## AVDRAMATISERA PROGRAMMERING

- “You don’t have to be a genius to code. Do you have to be a genius to read? – Bronwen, Technical artist at Valve
- Oavsett vad du har för utbildningsnivå är modern teknik oerhört komplicerad, men våra studenter kommer att förstå att all teknik kan reduceras till ett begränsat antal enkla instruktioner som programmerades av någon som dem.





# DATALOGISKT TÄNKANDE OCH PROGRAMMERING

*Tom Tits  
Experiment*

## FRÅGOR I FOKUS IDAG

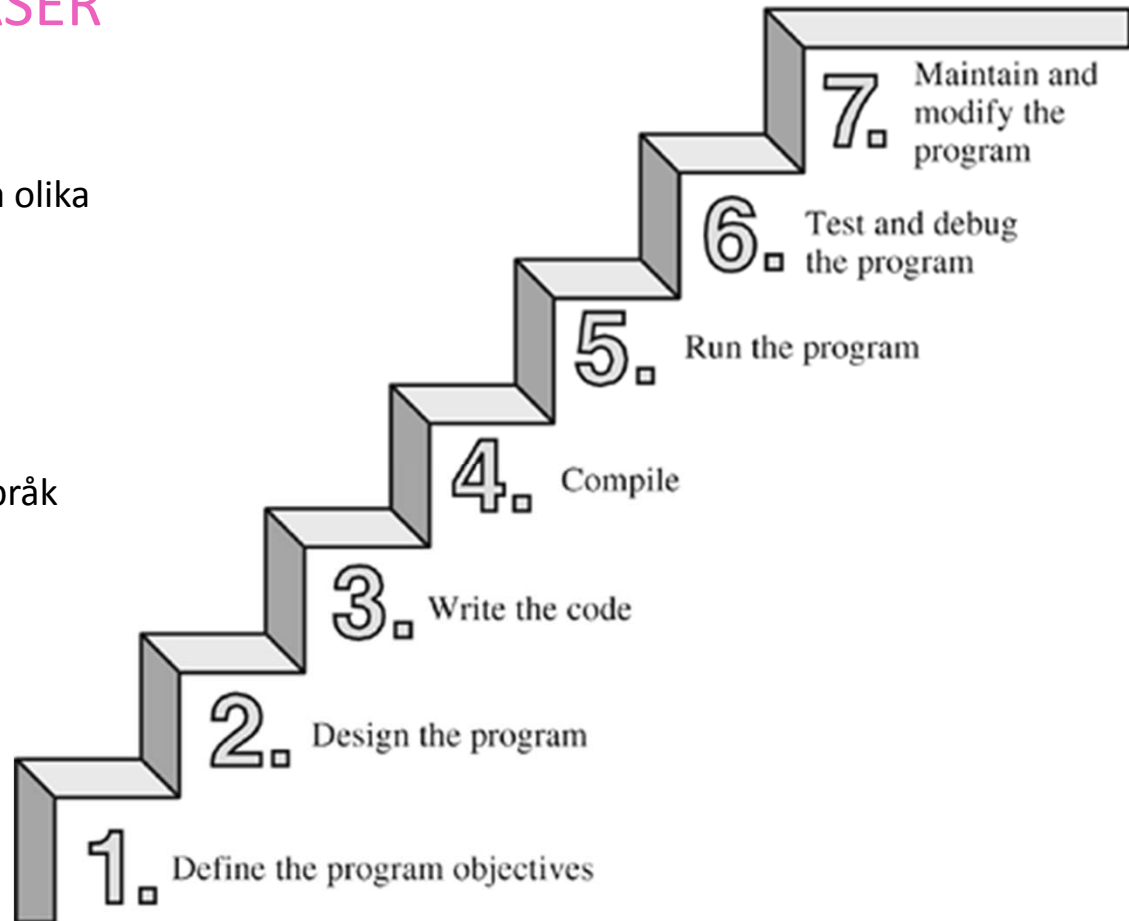
- Varför är digitala kompetenser och programmering viktigt idag?
- Vad är datalogiskt tänkande och programmering?
- Vilka övningar skulle vi kunna göra för att öva Digitala kompetenser och Datalogiskt tänkande i verksamheten?



*Tom Tits  
Experiment*

# PROGRAM SKAPAS I FASER

1. Analysera problemet/ Utvärdera olika lösningsmodeller
2. Designa en lösning
3. Skriv programkoden ("koda")
4. "Översätta" koden till datorns språk
5. Testa programmet
6. Debugga programmet
7. Modifiera programmet



## ATT LÖSA ETT PROBLEM

Gör en beskrivning till en robot som ska tillaga en kopp thé till dig



1. Koka vatten
2. Sätt tepåsen i en kopp
3. Häll vatten i koppen

Att tänka på:

När man programmerar måste man vara oerhört detaljerad och exakt.



1. Fyll en kastrull med vatten
2. Sätt kastrullen på spisplattan
3. Vänta tills vattnet kokar
4. Ta av kastrullen från plattan

Ta fram en tepåse

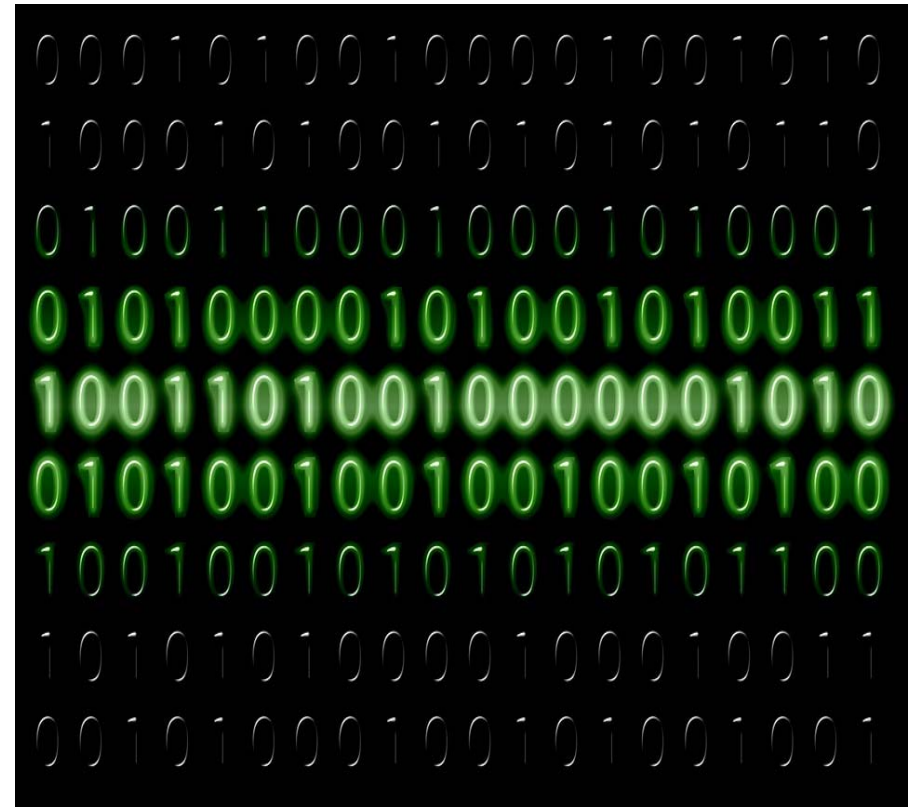
Sätt tepåsen i en kopp

7. Lyft vattenkastrullen
8. Luta kastrullen så att vattnet rinner ner i koppen
9. Vänta tills koppen är full
10. Ställ tillbaka kastrullen på plattan

*Tom Tits  
Experiment*

## ETTOR OCH NOLLOR

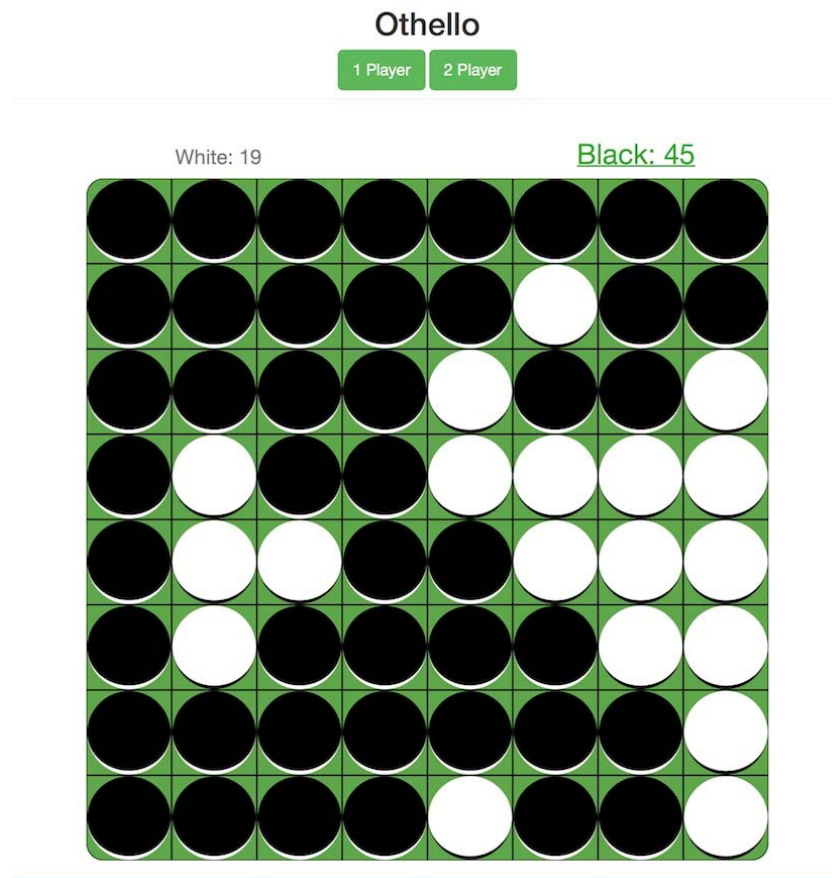
- I datorernas värld sker all kommunikation med ettor (1) och nollor (0)
- Detta kallas binär talrepresentation
- Binär talrepresentation används i praktiken i alla datorer eftersom dessa nyttjar digital elektronik.
- Det binära talsystemet är naturligt och effektivt i de fall där logiskt resonerande överförs till kalkyl. Detta sker i den booleska algebran, där de logiska sanningsvärdena, *falskt* och *sant* representeras av talen 0 respektive 1.



Tom Tits  
Experiment



# HUR FÅR MAN 5-ÅRINGAR ATT FÖRSTÅ BINÄRKOD



Character	Binary Code	Character	Binary Code	Character	Binary Code
A	01000001	a	01100001	!	00100001
B	01000010	b	01100010	"	00100010
C	01000011	c	01100011	#	00100011
D	01000100	d	01100100	\$	00100100
E	01000101	e	01100101	%	00100101
F	01000110	f	01100110	&	00100110
G	01000111	g	01100111	'	00100111
H	01001000	h	01101000	(	00101000
I	01001001	i	01101001	)	00101001
J	01001010	j	01101010	*	00101010
K	01001011	k	01101011	+	00101011
L	01001100	l	01101100	,	00101100
M	01001101	m	01101101	-	00101101
N	01001110	n	01101110	.	00101110
O	01001111	o	01101111	/	00101111
P	01010000	p	01110000	0	00110000
Q	01010001	q	01110001	1	00110001
R	01010010	r	01110010	2	00110010
S	01010011	s	01110011	3	00110011
T	01010100	t	01110100	4	00110100
U	01010101	u	01110101	5	00110101
V	01010110	v	01110110	6	00110110
W	01010111	w	01110111	7	00110111
X	01011000	x	01111000	8	00111000
Y	01011001	y	01111001	9	00111001
Z	01011010	z	01111010	?	00111010
				@	01000000

## TESTA/DEBUGGA/ MODIFIERA

- Det är vanligt att allting inte går helt perfekt första gången – eller även 50 gången.
- <https://youtu.be/lft0TcJVdm4?t=2m37s>
- Det finns många olika sätt att programmera en robot att brygga thé
- <https://youtu.be/yh4JbUXzPiA?t=2m7s>
- [https://youtu.be/OaB\\_y4gYeH8?t=6s](https://youtu.be/OaB_y4gYeH8?t=6s)



## VILLKOR

- När man programmerar så berättar programmeraren för datorn hur den ska reagera när den får ett kommando.
- Villkor gör att man kan få programmet att göra ett av två olika val.
  - OM (IF) någonting händer)
  - SÅ (THEN) ska datorn göra såhär
  - ANNARS (ELSE) gör något annat



# OM-SÅ-ANNARS LEKEN

- Regler
  - I varje omgång finns en programmerare och övriga deltagare är datorer
  - Programmeraren står framför datorerna och ger kommandon. (Programmeraren får ge tre kommandon innan man byter)
  - Programmeraren ger ett kommando:  
OM jag \_\_\_\_ SÅ ska ni \_\_\_\_
  - T ex "OM jag lyfter höger arm SÅ ska ni lyfta höger arm"
- Svårighetsgrader
  - 1: OM jag gör såhär SÅ gör ni likadant
  - 2: OM jag gör såhär SÅ gör ni sådär
  - 3: OM jag gör såhär, SÅ gör ni sådär, ANNARS gör ni något annat



## FÖRSLAG PÅ AKTIVITETER

- Klassiska spel
- Öva problemlösning (Att tala om hur/på olika sätt man kan lösa ett problem)
- Att arbeta med mönster i färg och form
- Beskrivningar och kartor
- Att följa instruktioner och att skapa/ge instruktioner till andra
- Göra hinderbanor åt andra
- Appar som där man kan öva datalogiskt tänkande och programmering
  - <http://www.scratchjr.org/>
- Programmerbara fysiska hjälpmedel/robotar
- Tänk och testa spåret på Tom Tits



## NAMNKOD/HALS BAND

- Använd ASCII-kod för att göra ett namnhalsband
- Använd binära tal för att göra ett armband som berättar när du är född
- Se separat övningsblad





PROGRAMMERA MED  
BLUE-BOT, SCRATCH JR & OZOBOT

*Tom Tits  
Experiment*