



HF0010

Introduktionskurs i Datateknik

F3: World Wide Web?



Introduktionskurs i Datateknik

F1: Vad är en dator?

F2: programmeringsspråk?

L1: Hour of code, alla kan programmera, Diplom!

L2: Visuellt programmering av svårare problem!

F3: Webbprogrammering – JavaScript!

L3: IDE, Web sida & JavaScript!

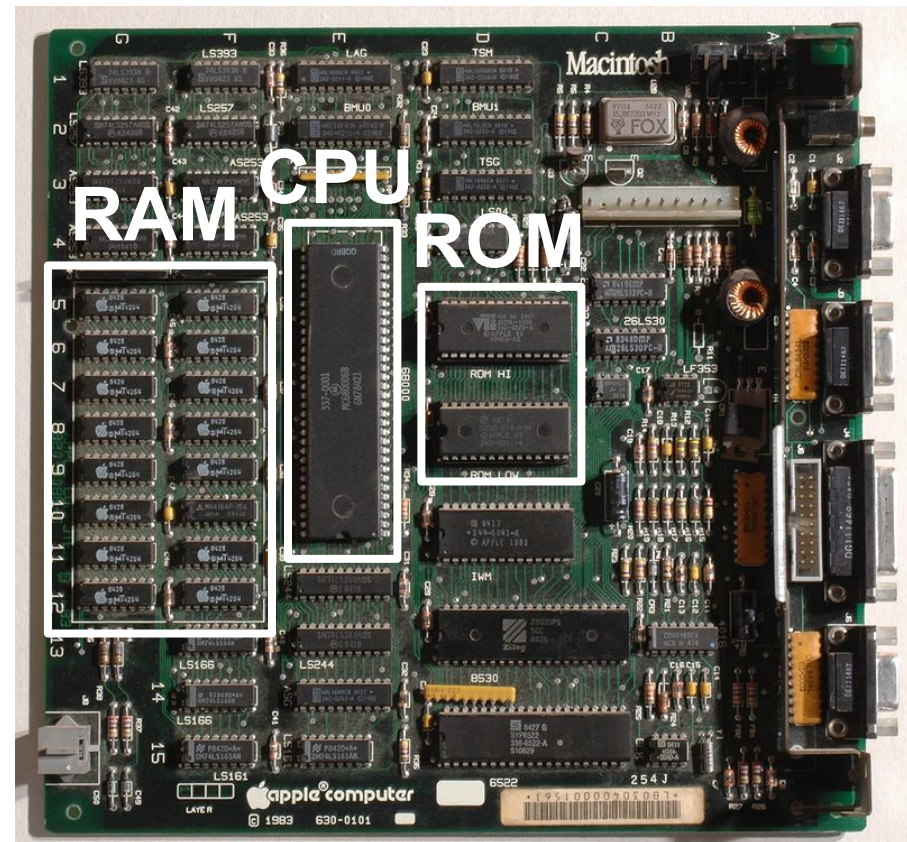
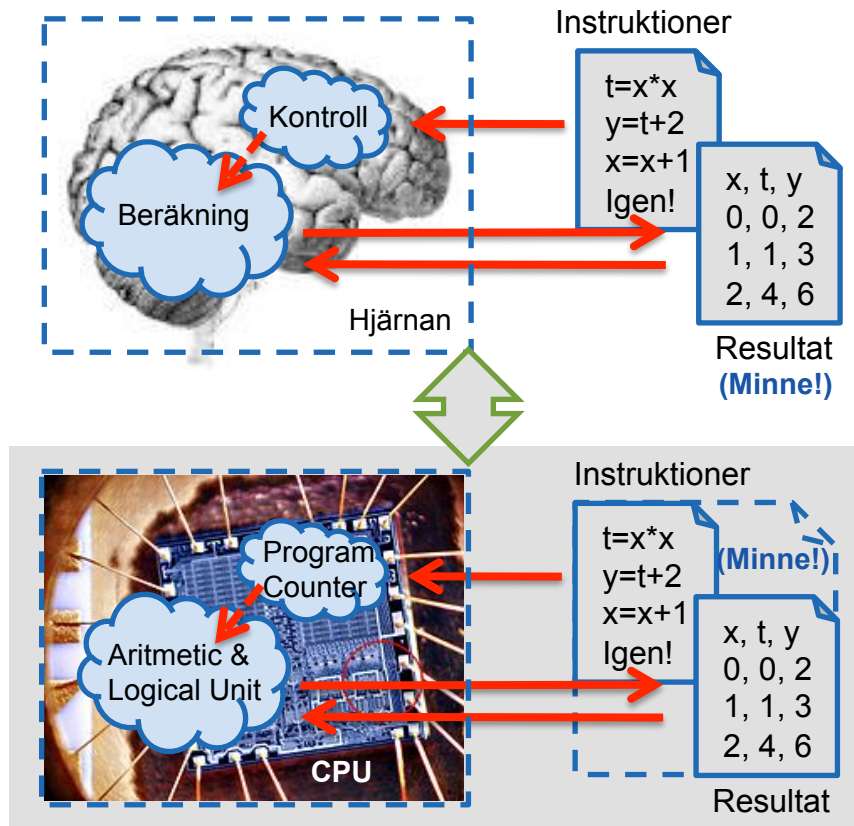


HTML



Vi ska lära oss programmera...

...då måste vi nog kunna två saker...

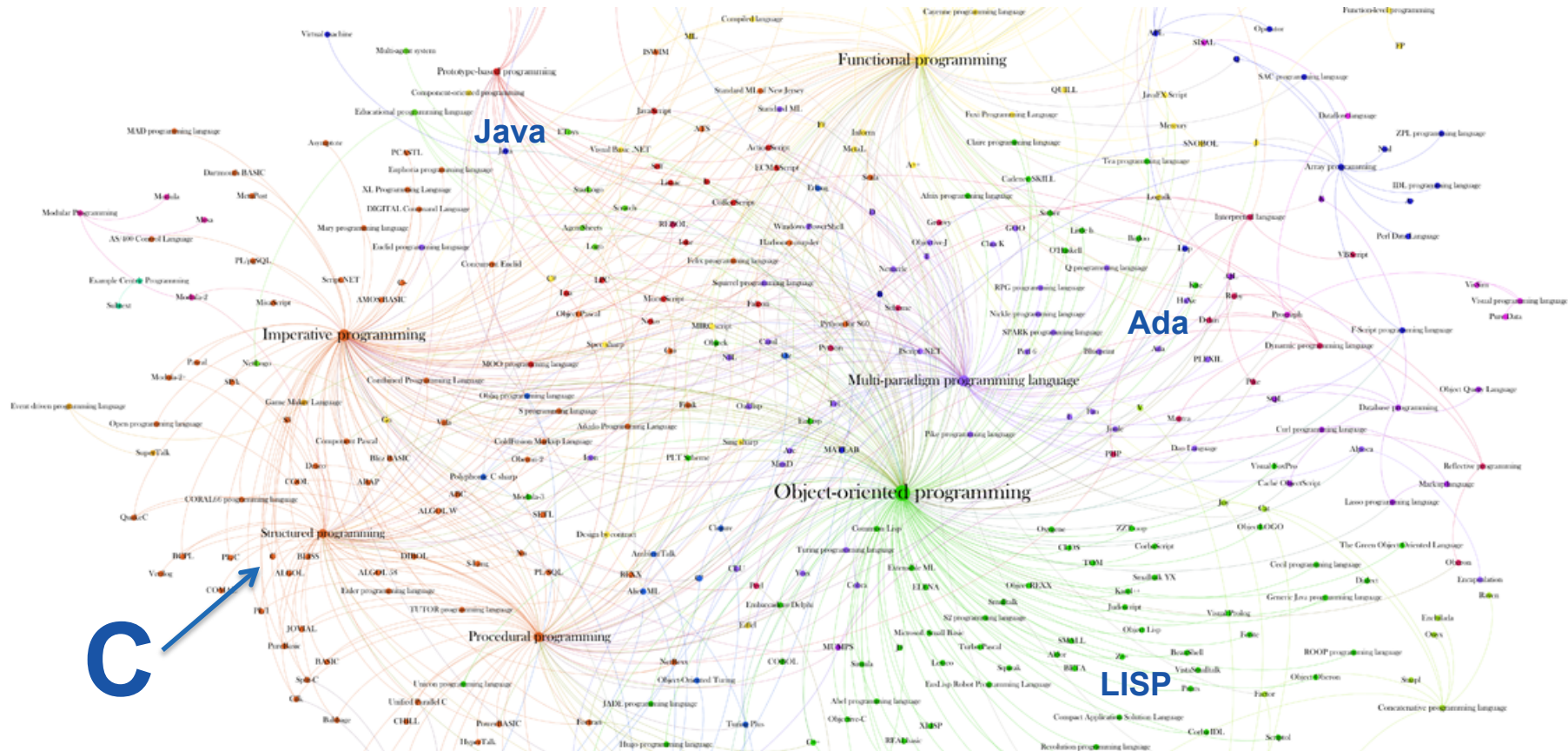


...ett programmeringsspråk och hur det används!



1b

Ja det gör det, men välj rätt!



Generell, maskinnära, processororienterad, funktionell...



1c

Syntax, semantik och reserverade ord!

```
while ( closed ( 'A' , number + 1 ) ) ;
```

Vad är "ord"
(tokens)
i språket!

Syntax

Identifiers
*[A-Za-z_][A-Za-z0-9_]**

Literals Operators

(Tokens)

Punctuation marks
,;(){}::white space

(Keywords ≈50!)

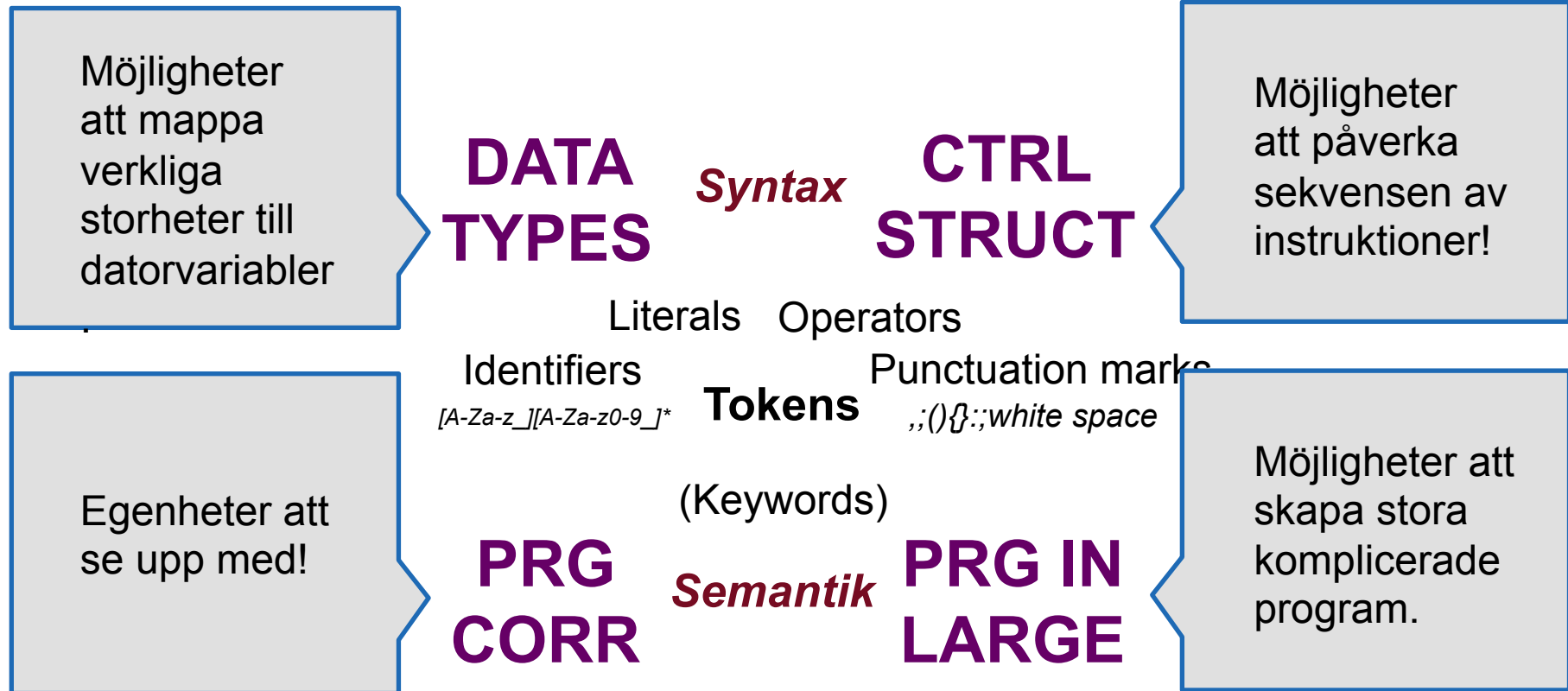
Semantik

Vad är en
mening
i språket!

Antalet mellanslag spelar nästan alltid ingen roll!



Likheter...



Vi börjar med att titta på vad data typer är...



2a

Hur använder man (dator-)språket?

Arbetsintensiv Monoton

Konsten att översätta en **verklig process** till datorns värld...

Farlig Felkänslig



Väg 4 tomater och beräkna medelvärdet!



2b

Hur använder man (dator-)språket?

Arbetsintensiv Monoton

Konsten att översätta en **verklig process** till datorns värld...

Farlig Felkänslig



Väg 4 tomater och beräkna medelvärdet!



Nollställ miniräknaren.

Repetera 4 gånger:

Väg nästa tomat...

**...addera vikten till
summan på miniräknaren.**

Dividera summan med 4!



2c

Hur använder man (dator-)språket?

Arbetsintensiv Monoton

Konsten att översätta en **verklig process** till datorns värld...

Farlig Felkänslig



Väg 4 tomater och beräkna medelvärdet!



Nollställ miniräknaren.

**Repetera 4 gånger:
Väg nästa tomat...
...addera vikten till
summan på miniräknaren.**

Dividera summan med 4!



```
int measurement;  
int total = 0;  
  
for (int counter=0; counter<4; counter++) {  
    scanf("%d",&measurement);  
    total+=measurement;  
}  
  
printf("Avrage = %f \n", total/4.0);
```

...det är att kunna programmera, och rätt lika från språk till språk!



Reflektioner från Lab #1...

The screenshot shows a Minecraft game window on the left with a character in a grassy field. On the right is a programming interface with a text box at the top containing the instruction: "Trä är en viktig resurs. Mycket kan tillverkas av det. Gå till trädet och använd kommandot 'förstör block' för att hugga ned det." Below this is a 'bitar' (blocks) palette on the left and an 'Arbetsyta' (workspace) on the right. The workspace shows a sequence of four blocks: 'vid start' (orange), 'gå framåt' (teal), 'gå framåt' (teal), and 'förstör block' (teal). A blue arrow points to the sequence with the text 'Sekvens...'. At the bottom right, a code block contains the following text:

```
moveForward();  
moveForward();  
destroyBlock();
```



Reflektioner från Lab #1...

Varje hus börjar med en vägg. Bygg den första delen av ditt hus genom att lägga "placera"- och "gå framåt"-kommandona i upprepningsloopen.

bitar

Arbetsyta: Börja om Visa kod

gå framåt

placera björkplankor ▾

sväng vänster ↺ ▾

sväng höger ↻ ▾

upprepa 4 ▾ gånger
gör

vid start

upprepa 4 ▾ gånger

gör placera björkplankor ▾
gå framåt

Iteration...

```
for (var count = 0; count < 4; count++) {  
  placeBlock("planksBirch");  
  moveForward();  
}
```

4ggr



Reflektioner från Lab #1...



Kör

Det är bra att planera framåt. Plantera grödor på båda sidor om vattnet så att du inte blir hungrig längre fram.

bitar Arbetsyta: 12 / 11 block **Börja om** **Visa kod**

gå framåt

plantera gröda

sväng vänster

sväng höger

upprepa 6 gånger
gör

sax

vid start

upprepa 6 gånger
gör **plantera gröda**
gå framåt

sväng höger

gå framåt

gå framåt

sväng höger

gå framåt

upprepa 6 gånger
gör **plantera gröda**
gå framåt



Reflektioner från Lab #1...

Det är bra att planera framåt. Plantera grödor på båda sidor om vattnet så att du inte blir hungrig längre fram.

```
for (var count2 = 0; count2 < 2; count2++) {  
  for (var count = 0; count < 6; count++) {  
    plantCrop();  
    moveForward();  
  }  
  turnRight();  
  moveForward();  
  moveForward();  
  turnRight();  
  moveForward();  
}
```

Arbetsyta: [Börja om](#) [Visa kod](#)

vid start
upprepa 2 gånger
gör
 upprepa 6 gånger
 gör
 plantera gröda
 gå framåt
sväng höger
gå framåt
gå framåt
sväng höger
gå framåt

Nästlade iterationer!

Reflektioner från Lab #1...



The screenshot shows a Minecraft game window on the left with a character in a cave. On the right is a block editor interface. At the top right, a text box explains: "Lava döljer sig under några av de här blocken, och du måste täcka den innan du kan gå framåt. Ett 'om'-kommando blir praktiskt här. Lägg till ett 'gå framåt'-kommando på rätt plats för att bryta de här blocken." Below this is a toolbar with "bitar" and "Arbetsyta: 6 / 6 block", and buttons for "Börja om" and "Visa kod". The block editor has a list of blocks on the left: "gå framåt", "sväng vänster", "sväng höger", "förstör block", "upprepa 3", "om lava fram", and "utför". The main workspace shows a Scratch-style script starting with "vid start", followed by a loop "upprepa 7 gånger", then a "gör" block containing "förstör block", a "Selektion" block with "om lava framför" and "utför" (containing "placera kullersten framåt"), and finally "gå framåt". A code window at the bottom right shows the corresponding JavaScript code.

Lava döljer sig under några av de här blocken, och du måste täcka den innan du kan gå framåt. Ett "om"-kommando blir praktiskt här. Lägg till ett "gå framåt"-kommando på rätt plats för att bryta de här blocken.

bitar Arbetsyta: 6 / 6 block Börja om Visa kod

gå framåt
sväng vänster
sväng höger
förstör block
upprepa 3
om lava fram
utför

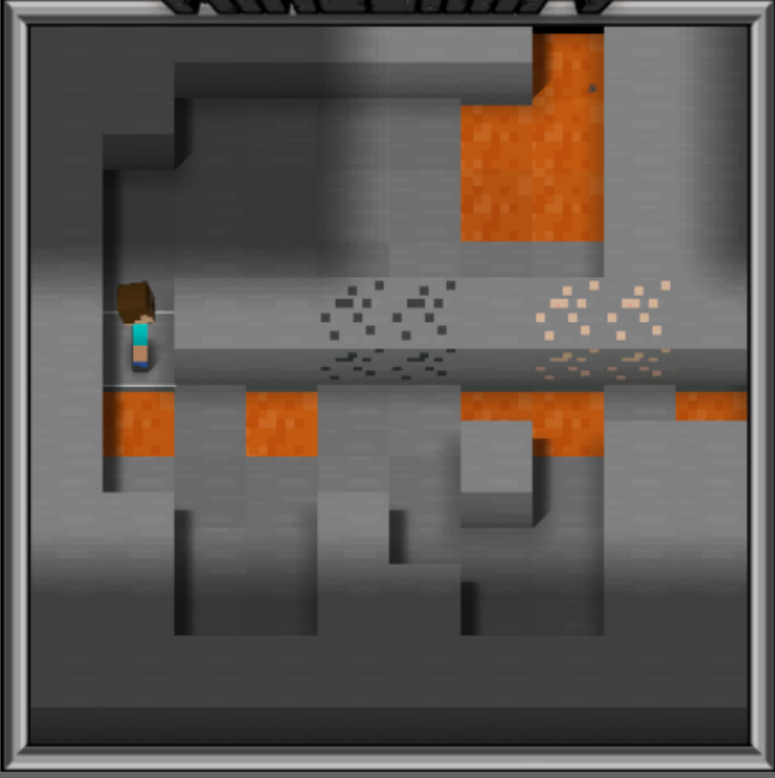
vid start
upprepa 7 gånger
gör
förstör block
om lava framför
utför placera kullersten framåt
gå framåt

Selektion

```
for (var count = 0; count < 7; count++) {  
  destroyBlock();  
  ifLavaAhead(function() {  
    placeBlockAhead("cobblestone");  
  });  
  moveForward();  
}
```



Reflektioner från Lab #1...



Lava döljer sig under några av de här blocken, och du måste täcka den innan du kan gå framåt. Ett "om"-kommando blir praktiskt här. Lägg till ett "gå framåt"-kommando på rätt plats för att bryta de här blocken.

bitar

Arbetsyta: 6 / 6 block Börja om Visa kod

gå framåt

sväng vänster

sväng höger

förstör block

upprepa 3 gånger

om lava framför

vid start

upprepa 7 gånger

gör

förstör block

om lava framför

utför

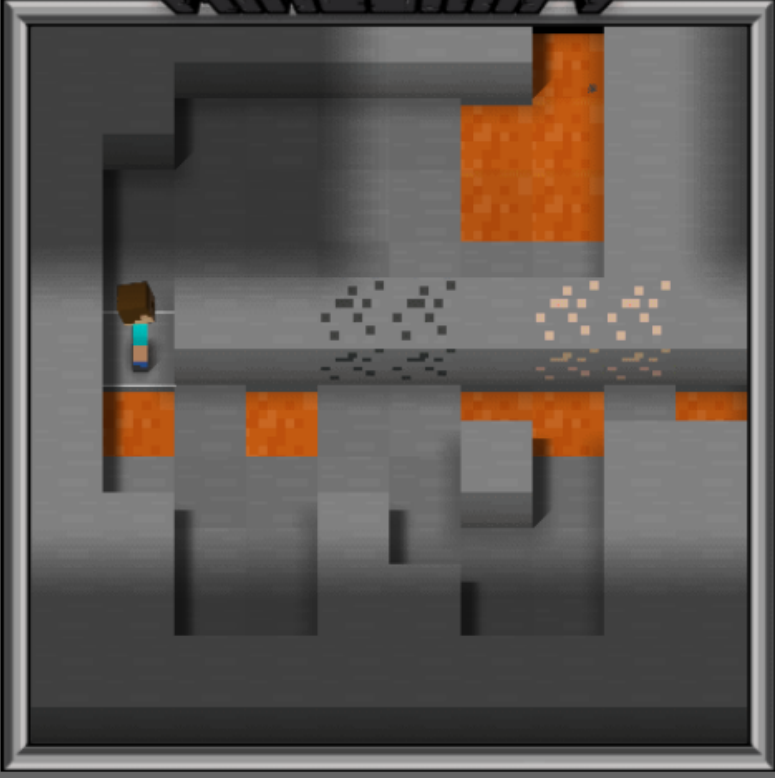
placera kullersten framåt

gå framåt

Sekvens Selektion

```
for (var count = 0; count < 7; count++) {  
  destroyBlock();  
  ifLavaAhead(function() {  
    placeBlockAhead("cobblestone");  
  });  
  moveForward();  
}
```

Reflektioner från Lab #1...



Lava döljer sig under några av de här blocken, och du måste täcka den innan du kan gå framåt. Ett "om"-kommando blir praktiskt här. Lägg till ett "gå framåt"-kommando på rätt plats för att bryta de här blocken.

bitar

Arbetsyta: 6 / 6 block Börja om Visa kod

gå framåt

sväng vänster

sväng höger

förstör block

upprepa 3
gör

om lava framför
utför

vid start

upprepa 7 gånger

gör

förstör block

om lava framför

utför

placera kullersten framåt

gå framåt

Iteration
Sekvens
Selektion

```
for (var count = 0; count < 7; count++) {  
  destroyBlock();  
  ifLavaAhead(function() {  
    placeBlockAhead("cobblestone");  
  });  
  moveForward();  
}
```



Förstår du det här blir resan lätt!

HI1026 Proj
HI1007 Java
HI1024 C
HI1030 DB
HI1029 DS
HI1031 Dist
HE1034 Nät
HE1033 Nät
HI1032 Nät
HE1037 Nät
HI1025 OS



GUI



CLI

Program
#1

Program
#2

*

.html

Browser

IoT
Prg

t.ex Java

KOMPILATOR

t.ex bytekode

OPERATIV
OS
SYSTEM

Virtuell Maskin

Standard Bibliotek

Filsys. Nätverk

Skärm tgb

Processer Minne

Kärna

Resurshantering

RT
OS

IoT
Prg

Assemblerkod

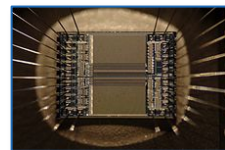
ASSEMBLER

Maskinkod



HE1028 uC
HE1041 uC

Maskinkoden utgör ett abstrakt gränssnitt som döljer bort hur den underliggande hårdvaran är uppbyggd...



...av ett antal IC-kretsar med miljontals grindar...

HE1026 Dig



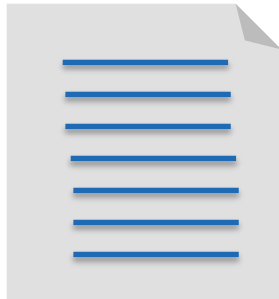
HE1030 Ana

...som i sin tur består av miljardtals transistorer!
(som trots allt inte kan göra mycket mer än 1+1!)

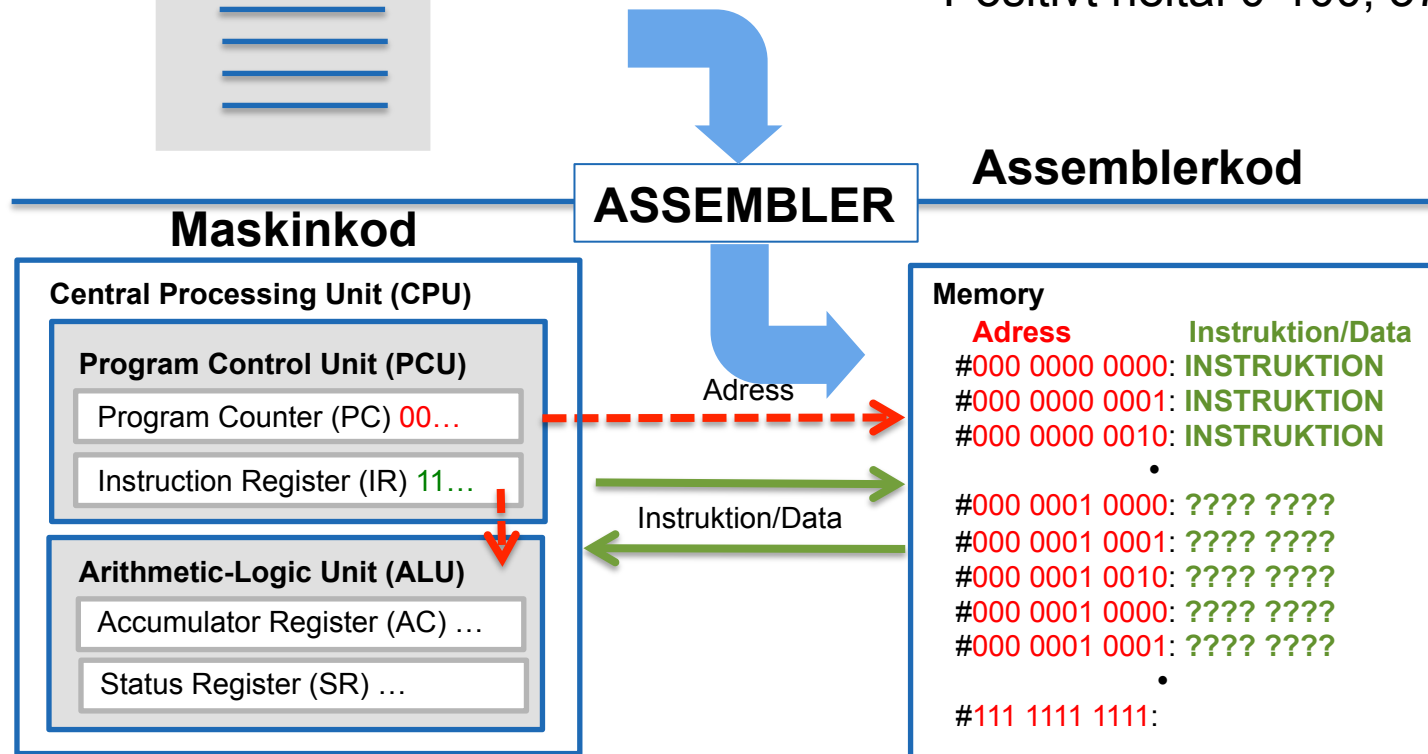


Järnkoll?

Feberdatalogger!



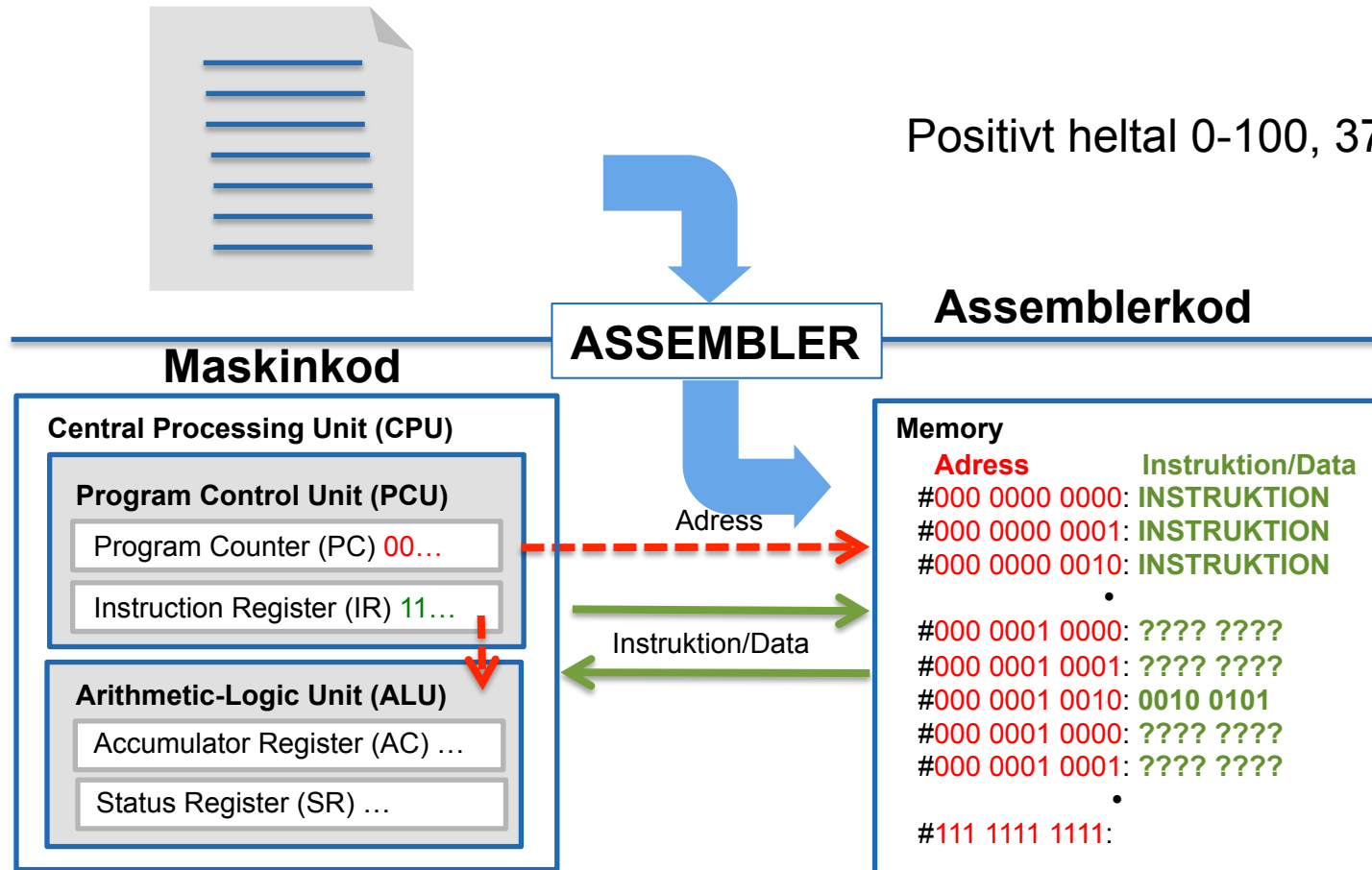
Positivt heltal 0-100, 37





Järnkoll?

Feberdatalogger!

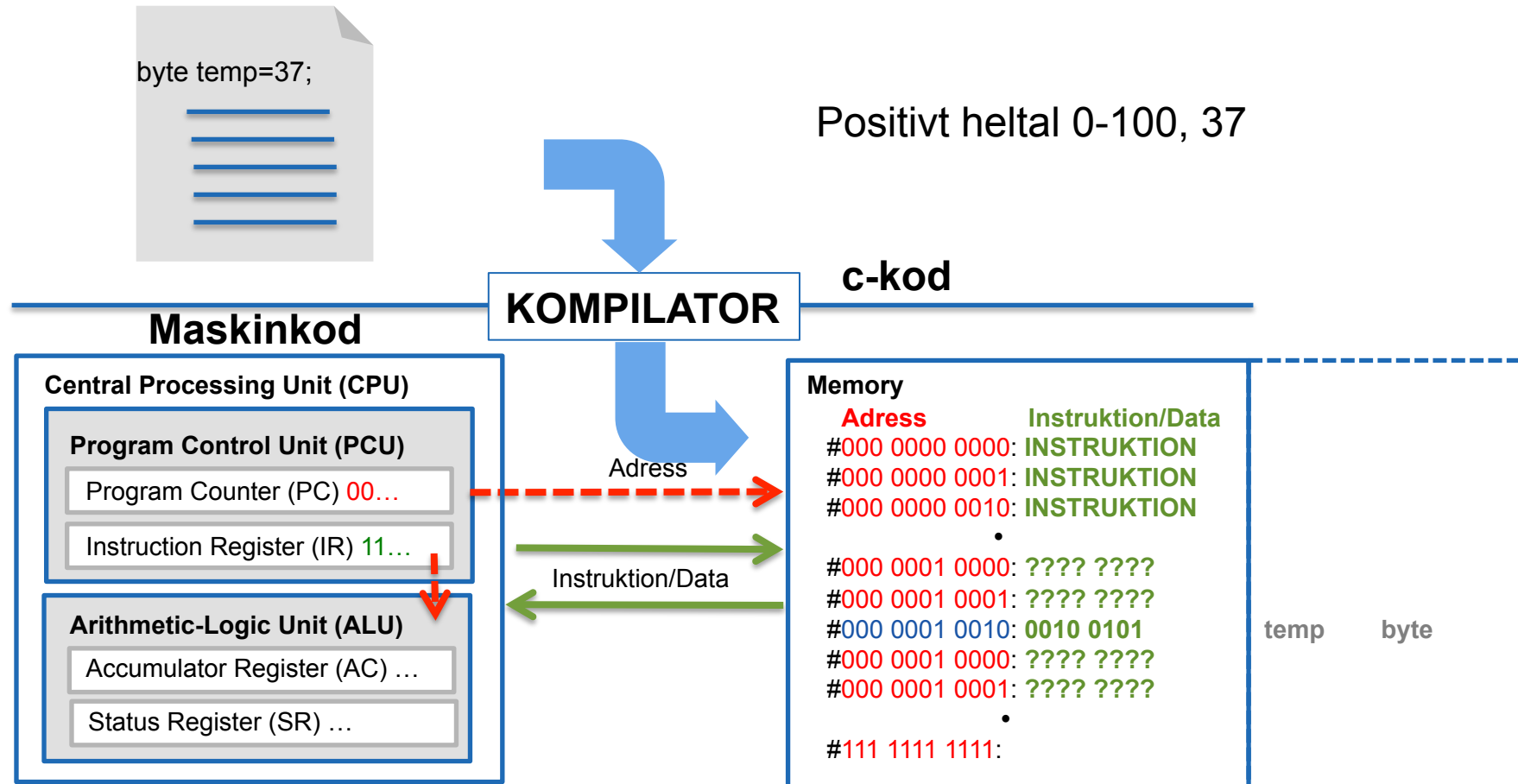


"Vi" måste komma ihåg att temperaturen finns lagrad i minnes-cell 34 och att bitmönstret ska tolkas som ett positivt heltal!



Järnkoll?

Feberdatalogger!

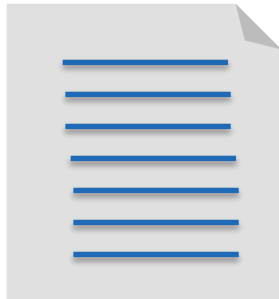


Kompilator väljer en ledig minnescell som "vi" därefter kan nå via variabelnamnet temp, för tal som är mellan 0 och 255.

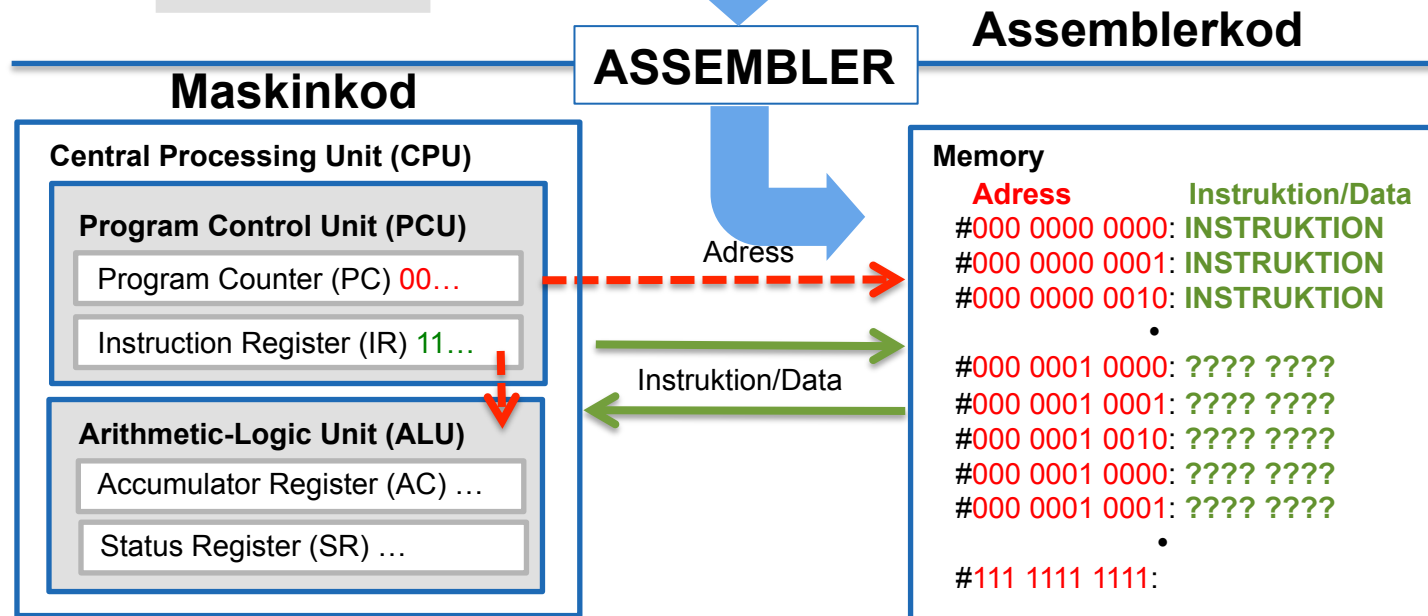


Järnkoll?

Tempdatalogger!

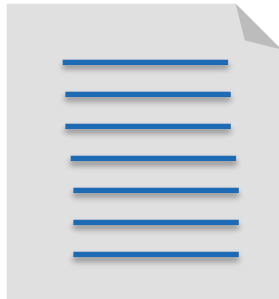


Positivt heltal 0-1000, 800





Järnkoll?



ASSEMBLER

Maskinkod

Central Processing Unit (CPU)

Program Control Unit (PCU)

Program Counter (PC) 00...

Instruction Register (IR) 11...

Arithmetic-Logic Unit (ALU)

Accumulator Register (AC) ...

Status Register (SR) ...

Adress

Instruktion/Data

Little-endian vs Big



Järnkoll?

short temp=800;

KOMPILATOR

Maskinkod

Central Processing Unit (CPU)

Program Control Unit (PCU)

Program Counter (PC) 00...

Instruction Register (IR) 11...

Arithmetic-Logic Unit (ALU)

Accumulator Register (AC) ...

Status Register (SR) ...

Adress

Instruktion/Data



Introduktionskurs i Datateknik

F1: Vad är en dator?

F2: programmeringsspråk?

L1: Hour of code, alla kan programmera, Diplom!

L2: Visuellt programmering av svårare problem!



F3: Webbprogrammering – JavaScript!

L3: IDE, Web sida & JavaScript!



HTML





Kursmål

Sidor märkta med den här symbolen innehåller begrepp som är extra viktiga för dina fortsatta studier!



Programutvecklingsmetodik!

Vad är Internet?

Hypertext Markup Language (HTML).

Web server v.s. Web browser.

Hypertext Transfer Protocol (HTTP).

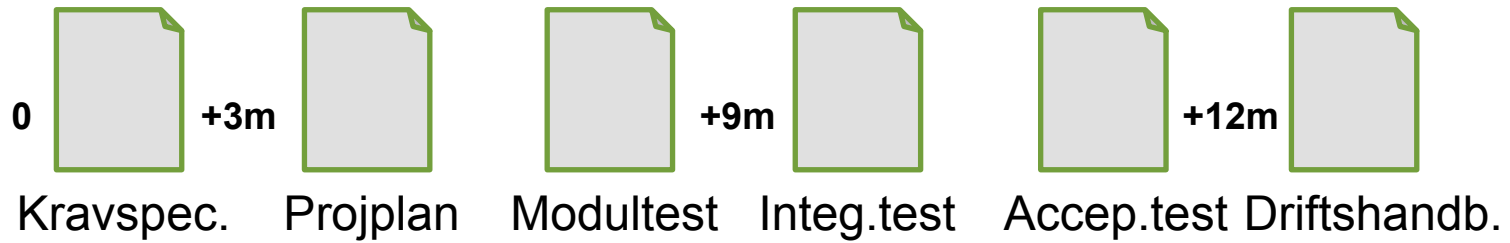
JavaScript.

Client/Server v.s. Peer-to-peer!



Programutvecklingsmetodik...

Vattenfalls
modellen

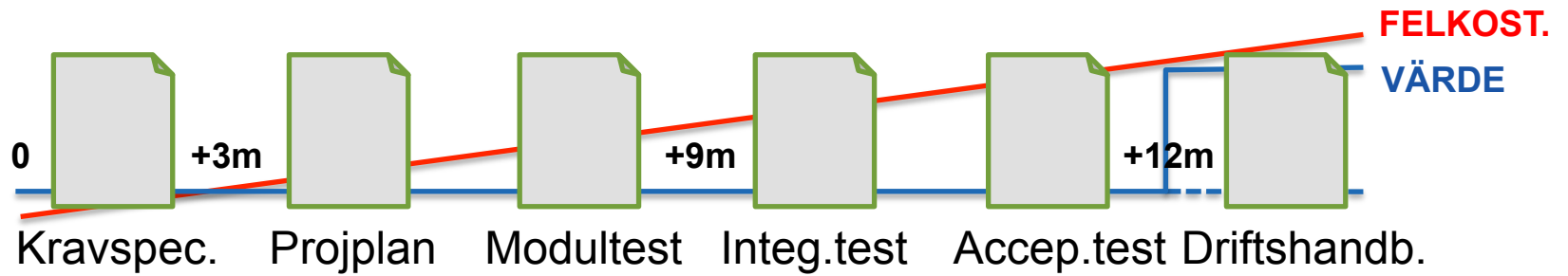


IDE!



Programutvecklingsmetodik...

Vattenfalls
modellen

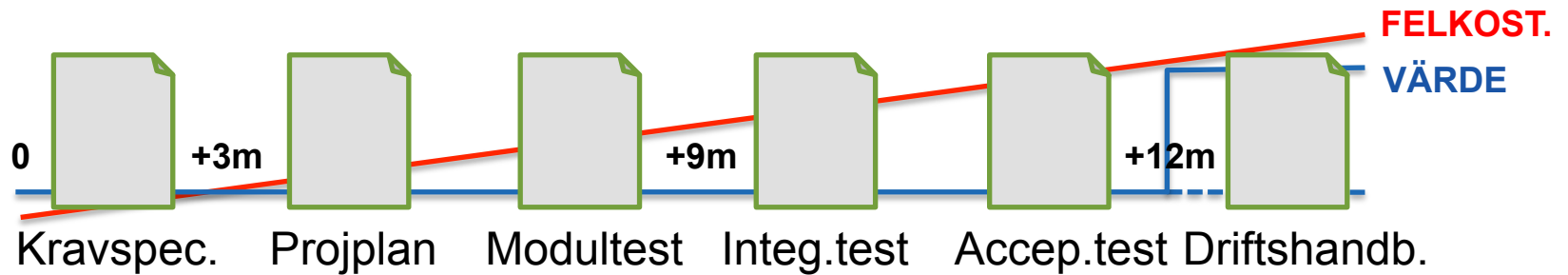


IDE!



Programutvecklingsmetodik...

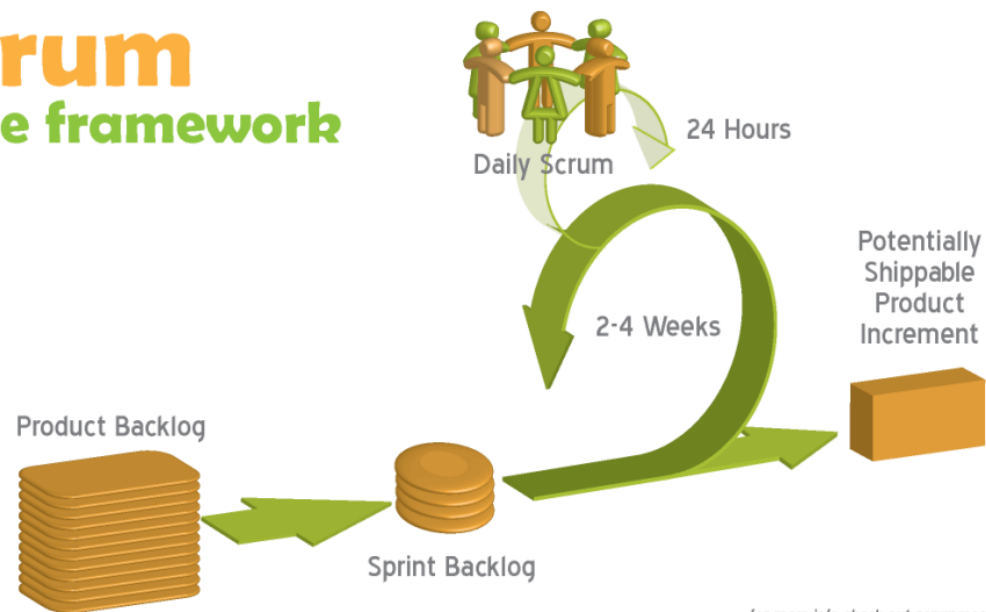
Vattenfalls
modellen



IDE!

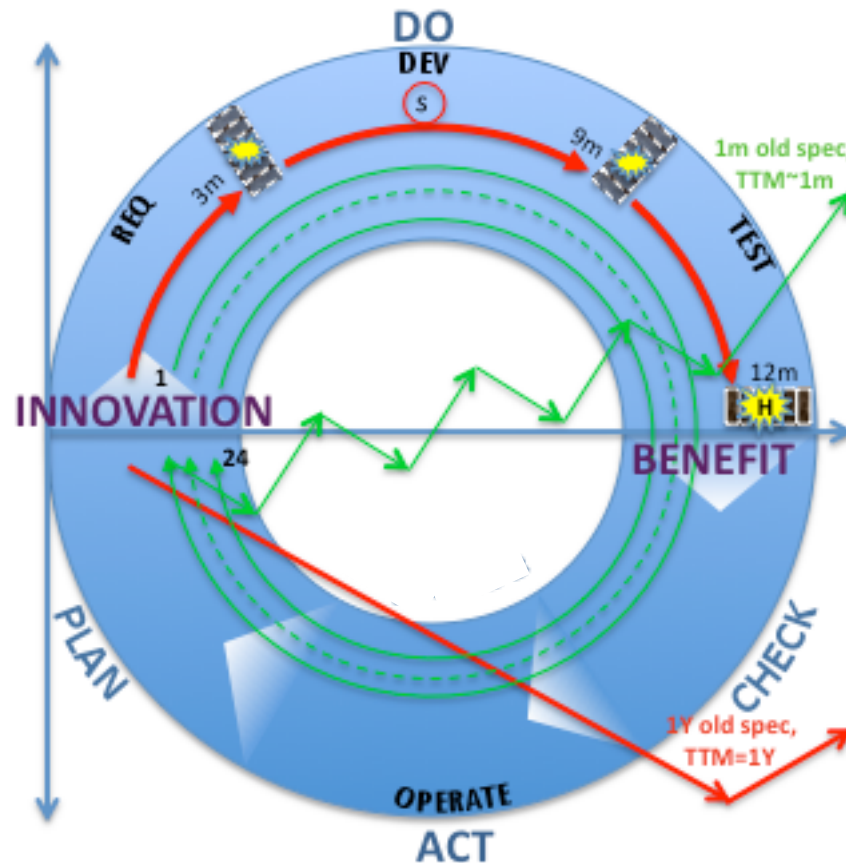


Scrum the framework



for more info check out scrummaster.com.au

Programutvecklingsmetodik...



Vattenfall

- + Ekonomistyrning
- Svåruppskattat
- Överlämningar
- Rätta fel kostsamt
- "Gammal" produkt

Agil/Scrum

- + "Tidigt" värde
- + Förstå eftersom
- + Inga överlämningar
- Kostnadsuppskattning

Ful Stack Developer!

DevOps



Integrated Development Environment (IDE)



HOME

Agile project management
as never seen before.

WORKING WITH SUBVERSION

CVS

1 Checkout

Your repository is the central spot for all of your project's files. To work on them, you checkout a local working copy to your Mac. There can be many working copies per repository.

2 Edit

3 Commit

4 Update

Repository

Working Copy

CHECKOUT



Code::Blocks

*Code::Blocks - The IDE
with all the features you
need, having a consistent
look, feel and operation
across platforms.*



Infrastructure as Code

Chef turns infrastructure into code. With Chef, you can automate how you build, deploy, and manage your infrastructure. Your infrastructure becomes as versionable, testable, and repeatable as application code.

Virtualization, Masterdata



Vad är Internet?

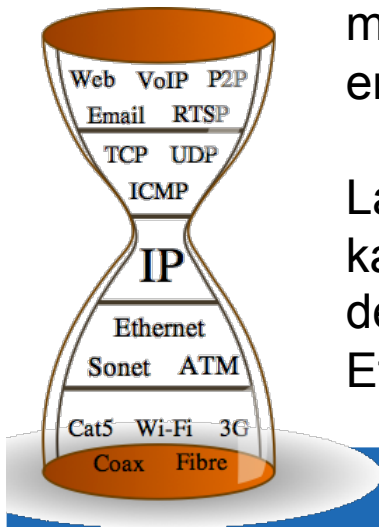


Termen internet är inte konstigare än att den beskriver flera sammankopplade datornät, nät av nät, internet!

Under 1970-talet började datornät bli vanligt förekommande men det var svårt att koppla ihop dem av tekniska skäl.

Robert Kahn och Vinton Cerf arbetade med problemet och skapade en lösning där det fanns ett nytt kommunikationslager, med unika adresser för alla datorer, som gjorde det relativt enkelt att knyta samman olika nätverk. Första näten -1982.

Lagret kallades Internet Protokoll (IP), och de unika adresserna kallades för IP-adresser. Om vi talar om det världsomspännande nät av ip-nät som vi t.ex. surfar via skriver vi Internet (stort I). Ett företagsinternt ip-nät kallas för ett intranet.



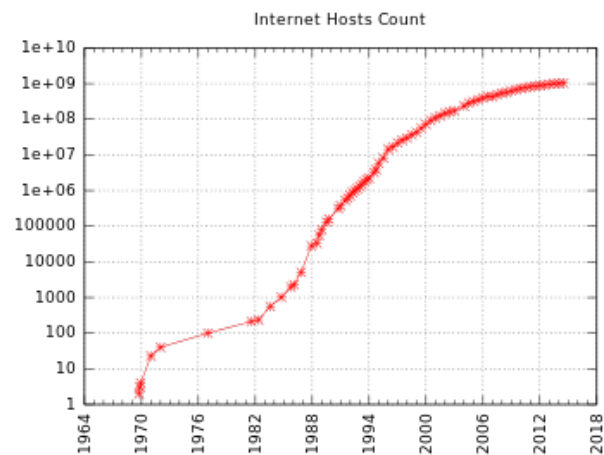
En ip-adress består av 4 tal (0..255) med punkter mellan talen.

130.237.84.8

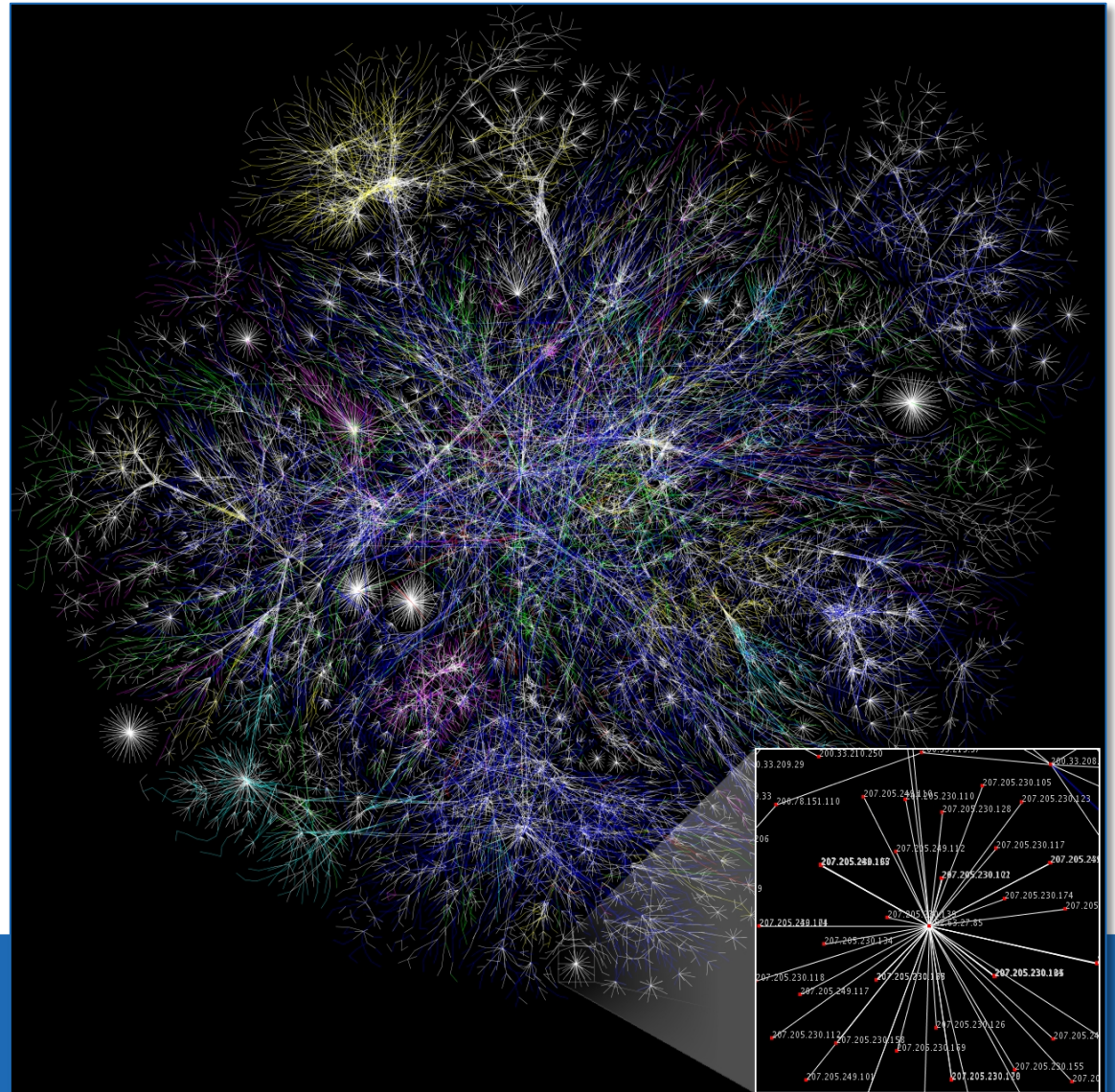


Internet

Via Internet går det i dag att nå över 1 miljard datorer men från början var det få. E-post och filöverföring var tidiga tjänster...



...men det var först när Berners-Lee uppfann WWW -91 som det tog riktig fart!





Vad var det Tim uppfann?!

URL

Protokoll

::/

Host

:

Port

/

Path

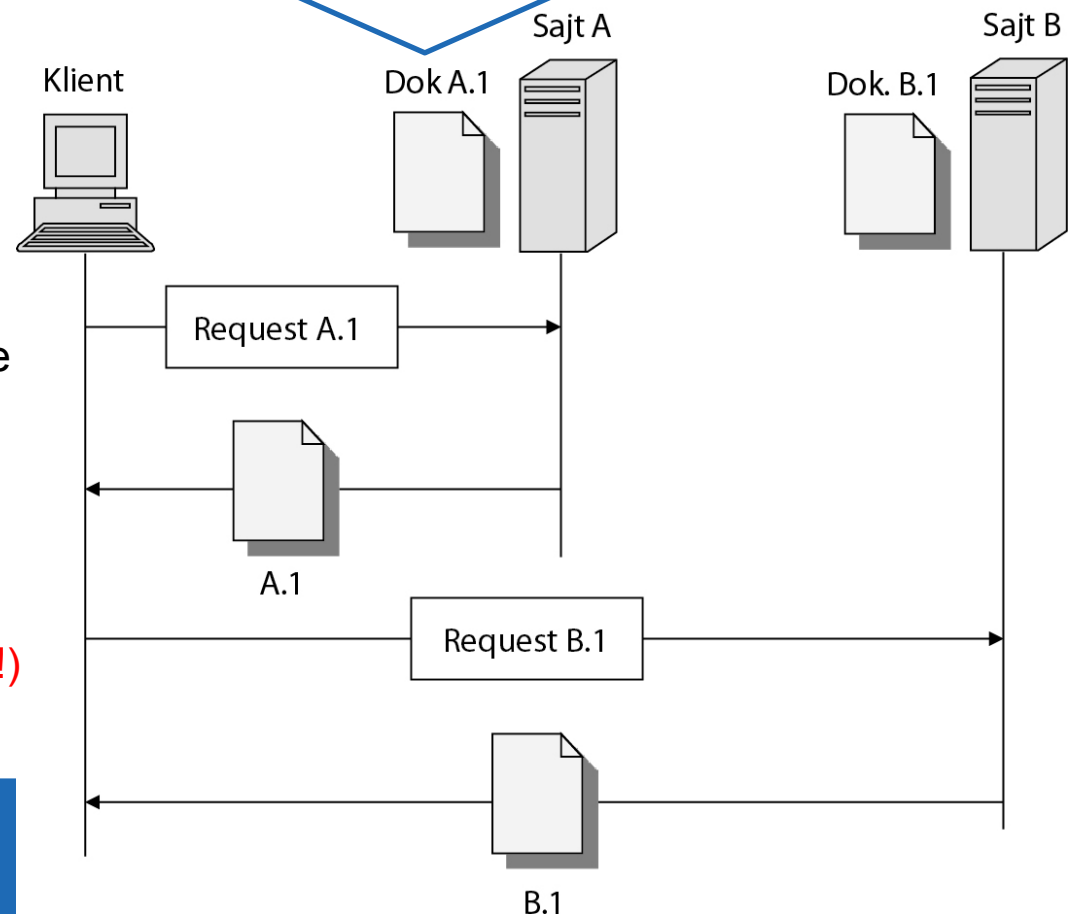
HTTP = Hypertext Transfer Protocol

Hur en klient (Web browser) hämtar en **specifik fil** från en Web server!!!

HTML = Hypertext Markup Language

Hur innehållet i den filen ska vara **uppmärkt** för att klienten ska kunna presentera innehållet!

(OBS: Inte ett programmeringsspråk!)





Att "märka upp" en text?!

A Sample Document

Rubrik

Wikipedians

June 23, 2007

Brödtext

This tiny sample showcases $\text{\TeX}$ and $\text{\LaTeX}$ for Wikipedia. More comprehensive examples, such as L^amport's `sample.tex` (see [1]) are available with any standard $\text{\TeX}$ distribution, or via www.tug.org. Only $\text{\TeX}$ nicians need [2].

1 Documents, Sections and Paragraphs

Kapitel

A $\text{\LaTeX}$ source file begins with a class declaration, followed by a preamble section, and the body of the document. The preamble contains settings and definitions. The body can contain sections, subsections, paragraphs, lists, equations, footnotes¹, figures and tables, and also commands for automatically-generated structures such as a table of contents, bibliography, index and glossary.

Commands such as `"\section"` introduce a new section, whilst an empty line delimits paragraphs. Other surplus white space is usually ignored.

2 Features

Both $\text{\TeX}$ and $\text{\LaTeX}$ allow for accents, and excel at typesetting mathematical equations, both in-line or displayed on a line by itself. For instance, an article on quadratics may need

$$ax^2 + bx + c = 0 \implies x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a},$$

Redan på -60-talet fanns det sätt att "märka upp" text så att ett tryckeri kunde trycka texten snyggt! Metoderna var olika, krångliga och främst för vetenskapliga texter...

...Tim skapade ett litet praktiskt mängd av "**taggar**" för vanlig text...

<h1>Rubrik nivå 1</h1>

<h2>Rubrik nivå 2</h2>

<i>Kursiv stil</i>

Det var sen upp till web-läsaren hur den skulle presentera innehållet! T.ex. kunde en tabell se helt olika ut i två olika läsare...



Nuvarande standard HTML5

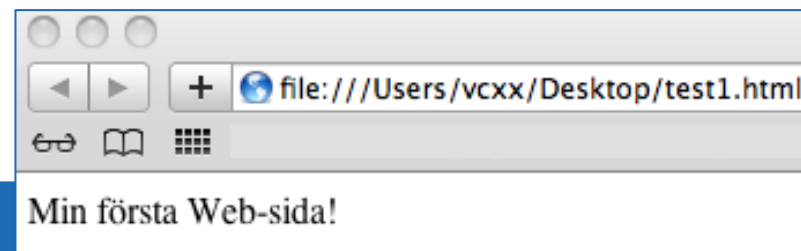


En minimal "uppmärkt" text ser ut så här!

Hela sidan! — {
Kontroll {
Innehåll {
—

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
  </head>
  <body>
    Min första Web-sida!
  </body>
</html>
```

Det här är en HTML5 uppmärkt sida!





Några till taggar...

`<h1>h1 tag</h1>`

`<h2>h2 tag</h2>`

`<h3>h3 tag</h3>`

`<h4>h4 tag</h4>`

`<h5>h5 tag</h5>`

`<h6>h6 tag</h6>`

`<p>Nytt stycke (paragraph)</p>`

"Extra"
mellanslag
& radslut
ignoreras!

UL – Unordered lists:

Example:

- Apples
- Oranges
- Bananas

UL Code:

```
<ul>  
  <li>List item 1</li>  
  <li>List item 2</li>  
</ul>
```

OL – Ordered Lists:

Example:

1. Breakfast
2. Lunch
3. Dinner

OL Code:

```
<ol>  
  <li>List item 1</li>  
  <li>List item 2</li>  
</ol>
```

h1

h2

ul

h2

ol

PIC Microcontrollers

Identification

- ISBN-13:978-0-13-119404-5
- ISBN-10:0-13-119404-6

Chapters

1. Introduction to Computing
2. The PIC Microcontrollers
3. Architecture
4. Branch, Call & Skip

LAB #3!
....TEST....



CSS & JavaScript

Dagens Web-sidor kräver att utvecklaren är insatt i tre områden!

- HTML – Som märker upp och specificerar innehållet på sidan.
- CSS – Som specificerar sidans utseende (layout).
- JavaScript – Som skapar sidans interaktiva egenskaper.

Två korta exempel...



CSS

- Före HTML5 fanns det taggar/attribut som indikerade hur den som skapat sidan ville att den skulle visas, men de var inte standardiserade eller heltäckande!
- Med CSS går det på pixelnivå att styra allt som visas!
- CSS-informationen ligger vanligen i en separat fil...
- ...men den kan skrivas i samma fil som innehållet.

PIC Microcontrollers

Identification

- ISBN-13:978-0-13-119404-5
- ISBN-10:0-13-119404-6

Chapters

1. Introduction to Computing
2. The PIC Microcontrollers
3. Architecture
4. Branch, Call & Skip



`<ul style="font-weight: bold">`



PIC Microcontrollers

Identification

- ISBN-13:978-0-13-119404-5
- ISBN-10:0-13-119404-6

Chapters

1. Introduction to Computing
2. The PIC Microcontrollers
3. Architecture
4. Branch, Call & Skip



JavaScript

- De första webbsidorna var helt statiska men snart skapades ett enkelt programmeringsspråk, JavaScript, som gjorde det möjligt att skapa webbsidor som interagerade med läsaren!
- JavaScript (som inte ska blandas ihop Java) kan:
 - Ändra innehållet på en sida,
 - Ändra attribut på en sida,
 - Ändra CSS på en sida,
 - Kontrollera att formulär är korrekt ifyllda!

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <script>
      function show() {
        alert(Date());
      }
    </script>
  </head>
  <body onClick="show()">
```





Webbutveckling...

- Vi ska snart tala om Programutveckling men du kommer säker att brottas med Webbutveckling i någon form när du börjar arbeta.
- Webbutveckling sker i dag alltid med hjälp av någon WYSIWYG-utvecklingsmiljö men som du har förstått är det bra att förstå hur man "märker upp" innehåll, lite känsla för layout samt viss insikt i programmering för interaktivitet.
- Den som är intresserad bör ta del av W3Schools utmärkta webbplats där det går att lära sig allt om HTML, CSS & JavaScript!

w3schools.com

THE WORLD'S LARGEST WEB DEVELOPER SITE

TUTORIALS ▾

REFERENCES ▾

EXAMPLES ▾





WIKI

Även om det är lätt att skapa en webbsida, vilket i dag alltid görs i någon WYSIWYG-editor, så krävs en viss kunskap för att det ska bli bra.

-04 myntades begreppet Web 2.0 för en ny typ av webbinnehåll där fokuset låg på användargenererat innehåll och användbarhet. Sociala medier tog fart och världen fick Bloggar, Tweets och Wikis.

En Wiki är ett otroligt bra verktyg för all form av dokumentation, individuellt eller i grupp!!!



WIKIPEDIA

English

The Free Encyclopedia

4 853 000+ articles

Español

La enciclopedia libre

1 172 000+ artículos

Deutsch

Die freie Enzyklopädie

1 806 000+ Artikel

日本語

フリー百科事典

962 000+ 記事

中文

自由的百科全书

814 000+ 條目



Русский

Свободная энциклопедия

1 213 000+ статей

Français

L'encyclopédie libre

1 614 000+ articles

Italiano

L'enciclopedia libera

1 193 000+ voci

Polski

Wolna encyklopedia

1 106 000+ haseł

Português

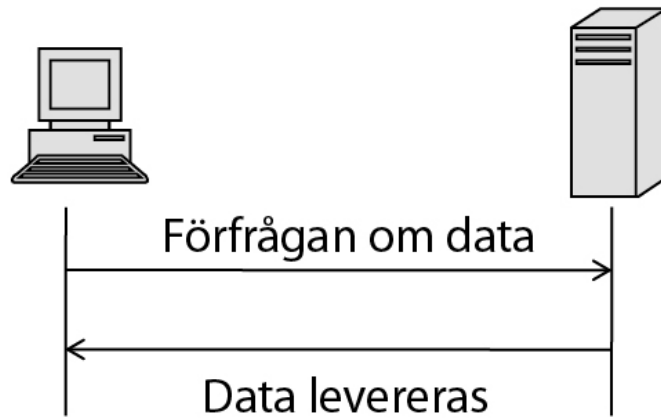
A enciclopédia livre

871 000+ artigos

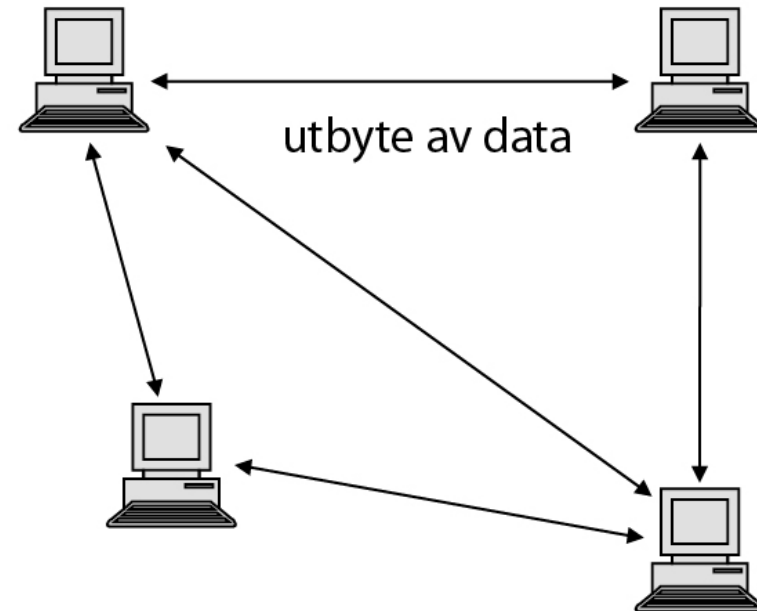
Tips: Använd en Wiki direkt i alla projekt!

Client/Server v.s Peer-to-peer...

(a)



(b)





Programmering...

1. Vi såg i föreläsning #1 att t.o.m dagens datorer inte förstår något annat språk än sitt eget maskinspråk...
2. I föreläsning #2 tittade vi på programmeringsspråk. De var lättare att förstå men begränsade (på olika sätt). För att programmera krävs det en ordbehandlare och en kompilator!
3. Det går att skriva "programmet" i nästan vilken ordbehandlare som helst, och sedan kompilera programmet men i dag använder alla en integrerad utvecklingsmiljö som underlättar alla steg i processen...
4. I kursen Grundkurs i programmering ska ni använda utvecklingsmiljön CodeLite som introduceras här, och som ni kommer att få installera i lab #3!

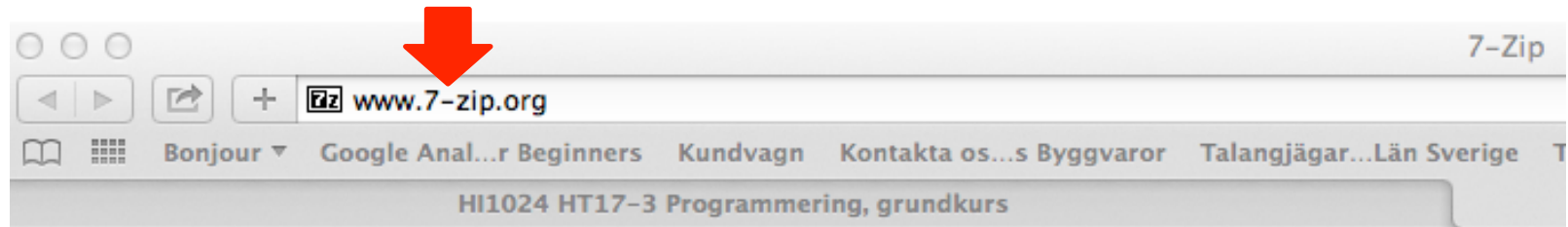


**A Powerful IDE for
C/C++
Supporting all major
compilers**



Installera CodeLite...

STEG 0: ENBART FÖR PC!



7-Zip is a file archiver with a high compression ratio.

Download 7-Zip 16.04 (2016-10-04) for Windows:

Link	Type	Windows	Size
Download	.exe	32-bit x86	1 MB
Download	.exe	64-bit x64	1 MB

7-Zip

License

Enbart för pc!

HÄMTA OCH INSTALLERA "PACKA UPP" PROGRAMMET 7-ZIP!



Installera CodeLite...

!!!Virusskydd!!!
MAC: Systeminställningar/Säkerhet & Integritet

STEG 1: Hämta CodeLite!

CodeLite IDE Products Download Screenshots GitHub Wiki Forums Blog wxCrafter [gitter](#) [join chat](#)

A Free, open source, cross platform C,C++,PHP and Node.js IDE

CodeLite 10.0 - Stable Release released on Jan 05, 2017 [What's New?](#)

CodeLite 10.0 for Windows 64 bit Installer Direct Link GitHub	sha1sum: 378ed383631a25e64df9ec5f7e2709195fcd30d3
CodeLite 10.0 for Windows 32 bit Installer Direct Link GitHub	sha1sum: d83fc6b0c6f185d0407e6dbdf11098dcb7825079
CodeLite 10.0 App Bundle for OSX 10.10 Direct Link GitHub	sha1sum: de2c94045971f8e0a9f3219ccc47a2d83cc22ce5

För Mac & pc!

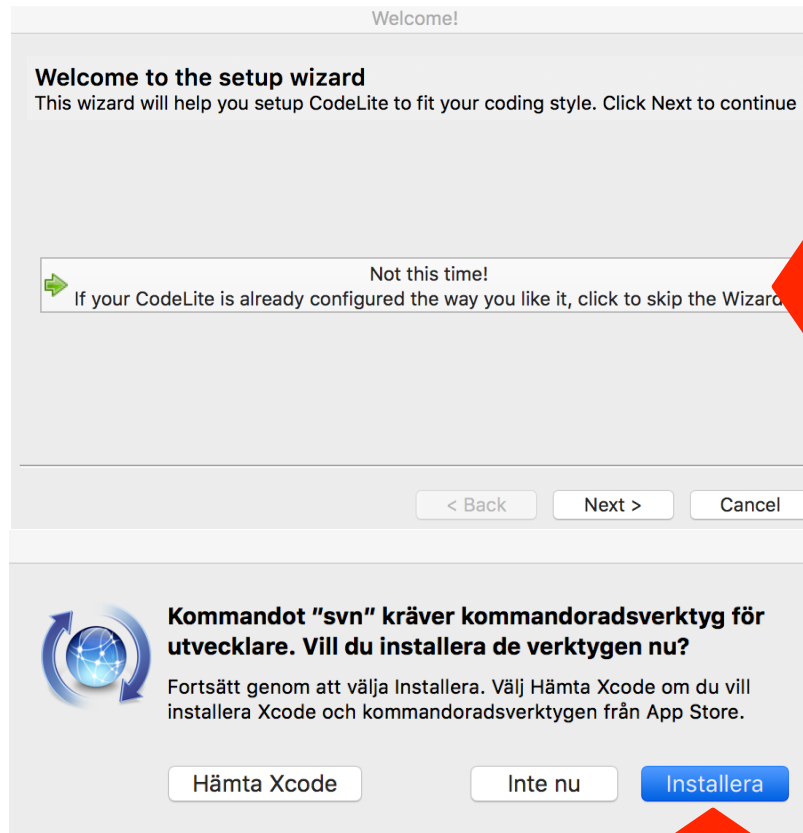
HÄMTA CodeLite...



Installera CodeLite...

STEG 2: Installera CodeLite!

Enbart Mac...



...välj skip...

...välj installera - klart!

För Mac

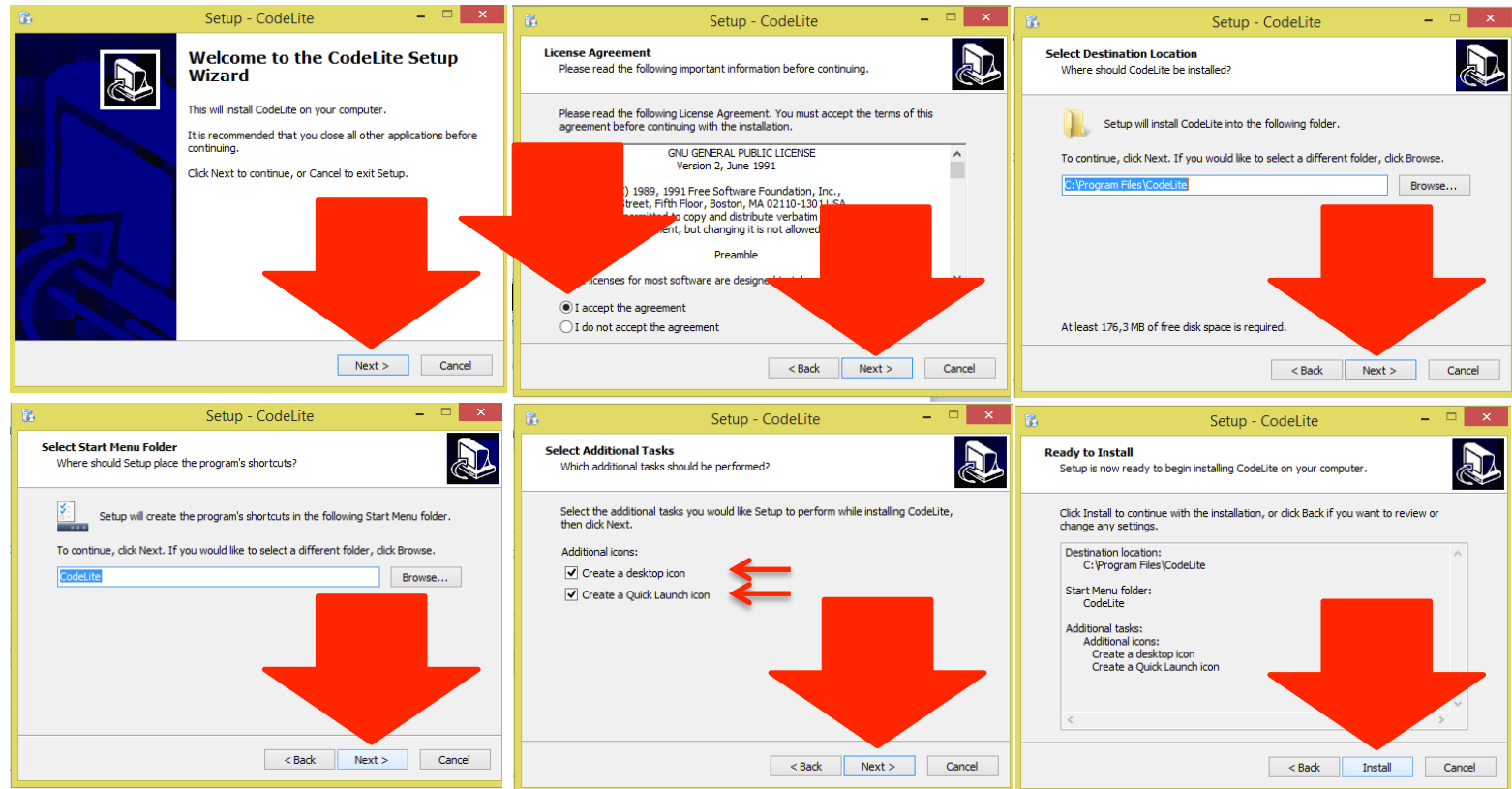
...Installera CodeLite, 2 steg!



Installera CodeLite...

Enbart PC...

STEG 2: Installera CodeLite!



(nu börjar den installera...)

För PC

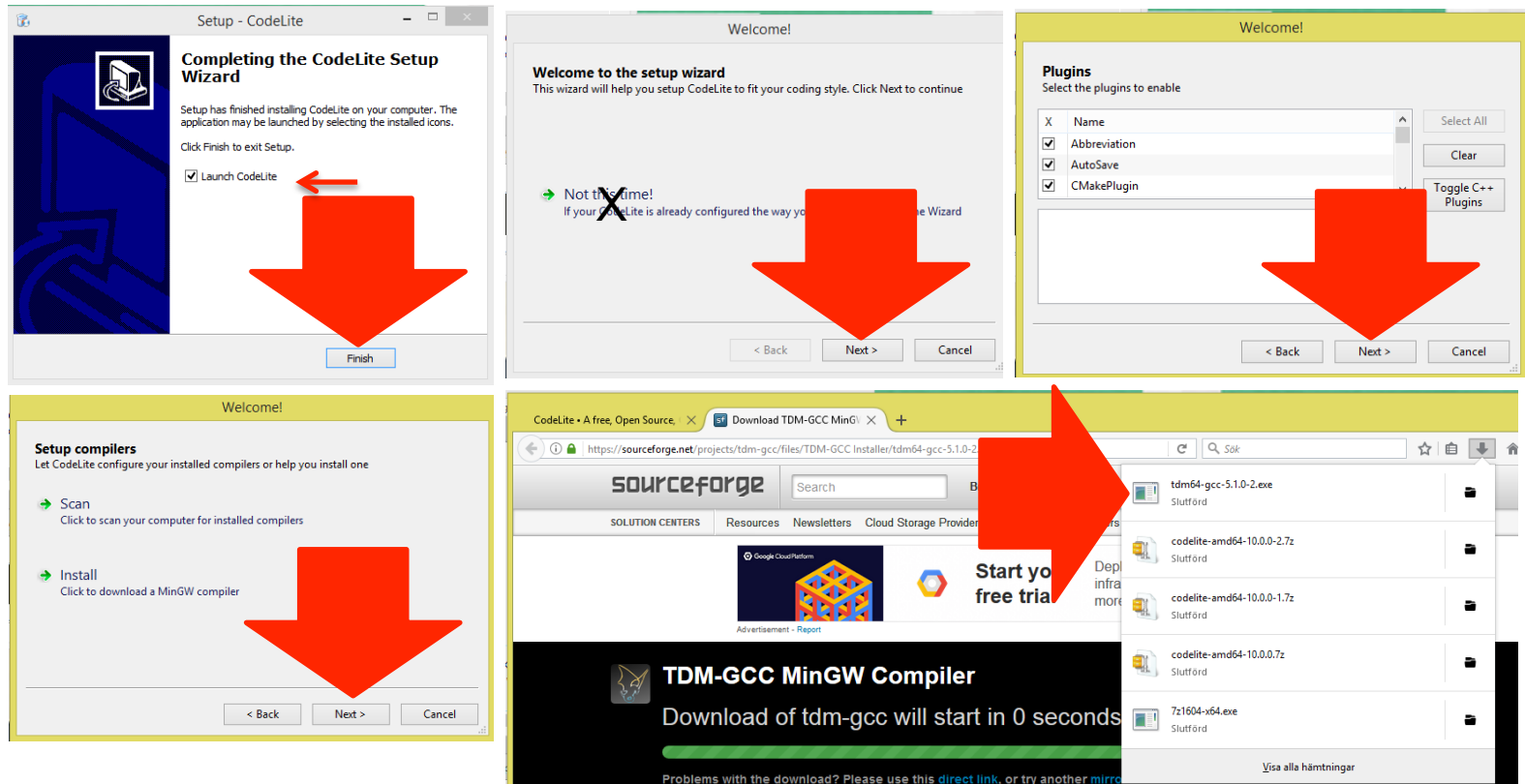
...Installera CodeLite, de första 6 stegen...



Installera CodeLite...

STEIG 2: Installera CodeLite!

Enbart PC...



(Så snart nerladdningen av MinGW kompilatorn är klar, starta installationen av den (om det inte sker automatiskt)!)

För PC

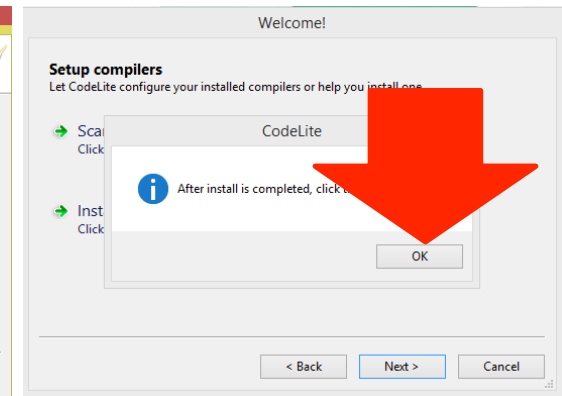
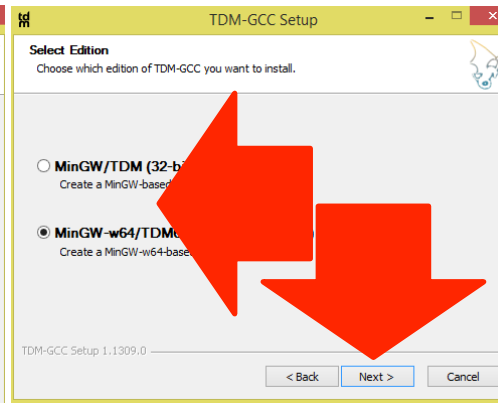
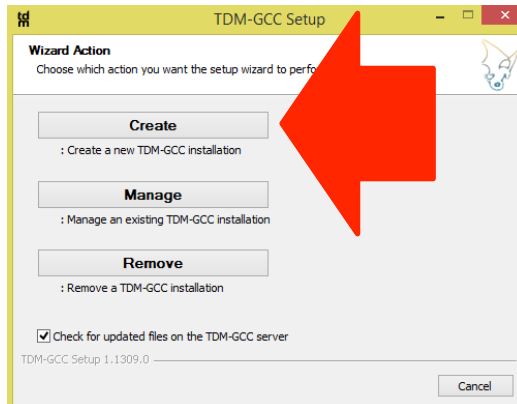
...Installera CodeLite, 5 stegen till...



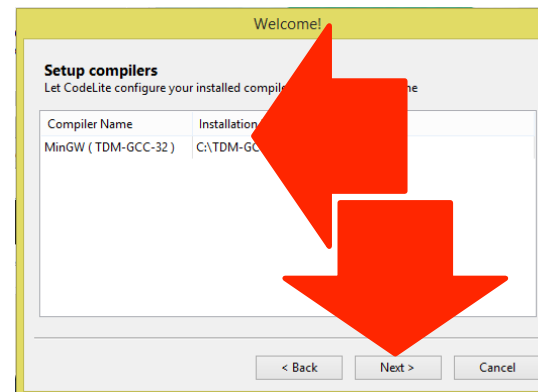
Installera CodeLite...

Enbart PC...

STEG 2: Installera CodeLite!



Tryck på Scan!



Next, Next Next...

Klar!

För PC

...Installera CodeLite, de sista X stegen!



STEIG 3: Sökväg kompilator!

Installera CodeLite...

VÄNTA MED DETTA!
GÄLLER INTE ALLA
VER. AV WINDOWS

Enbart PC...

Högerklicka på **computer** (den här datorn?) och välj **properties**.

Välj **advanced system and settings**.

Välj **Environment variables** (miljövariabler).

Välj **path**-variablen och välj **edit**.

Var noga så att du inte tar bort något som redan finns.

Vi ska bara lägga till.

Lägg till ett ; sist i raden och klipp sedan in sökvägen till gcc:.

I min installation:

;C:\TDM-GCC-64\bin

Öppna nu kommandotolken (cmd) och prova att skriva gcc.

Om den klagar över att den saknar input-fil har vi lyckats:

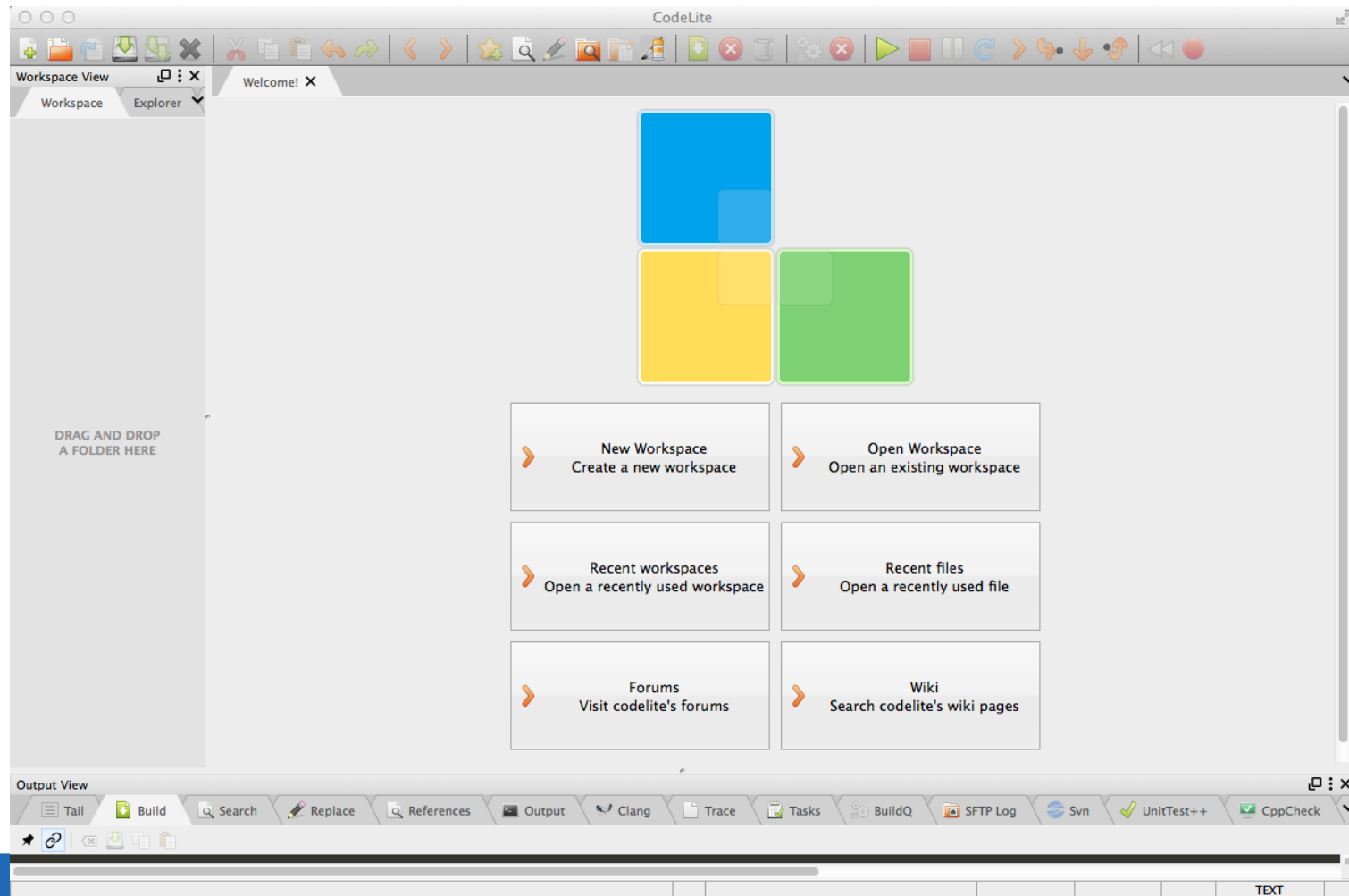
gcc: fatal error: no input files compilation terminated.

För PC

...gör detta steg om du inte kan kompilera, nästa exempel!



Använda CodeLite...



Starta CodeLite... ...det borde se ut ungefär som på bilden!



Använda CodeLite...

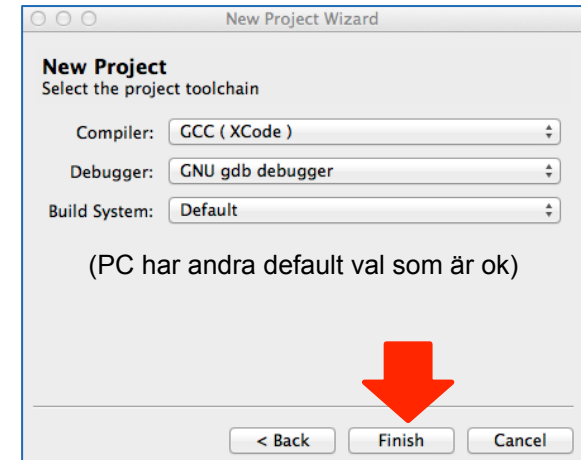
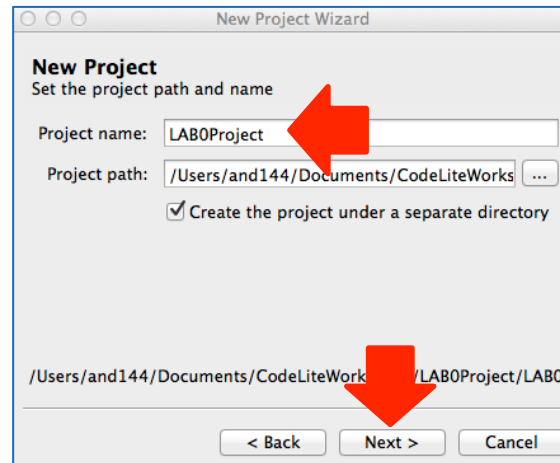
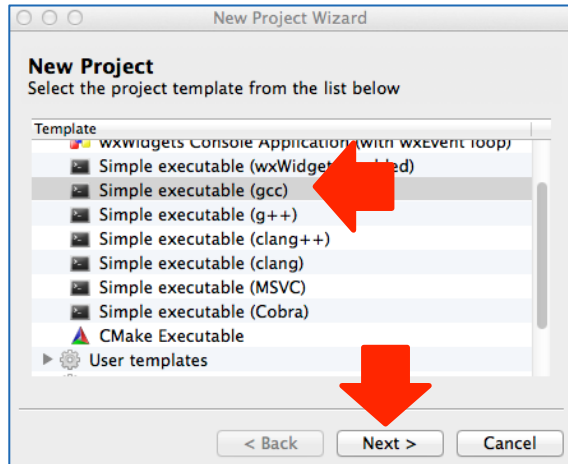
The screenshot shows the CodeLite IDE interface. A large red arrow labeled '1' points to the 'New Workspace' button in the central workspace creation menu. A red arrow labeled '2' points to the 'C++' option in the 'Select the workspace...' dialog. A red arrow labeled '3' points to the 'OK' button in the same dialog. A red arrow labeled '4' points to the 'Workspace Name' field in the 'New Workspace' dialog, which contains the text 'CodeLiteWorkspace'. A red arrow labeled '5' points to the 'Create the workspace under a new directory' checkbox, which is checked. A green arrow points from the 'CodeLiteWorkspace' entry in the 'Workspace View' sidebar to the 'OK!' text in the bottom right corner.

CodeLite behöver ett ställe att lagra dina filer...
...tryck på "create a new workspace"!

OK!

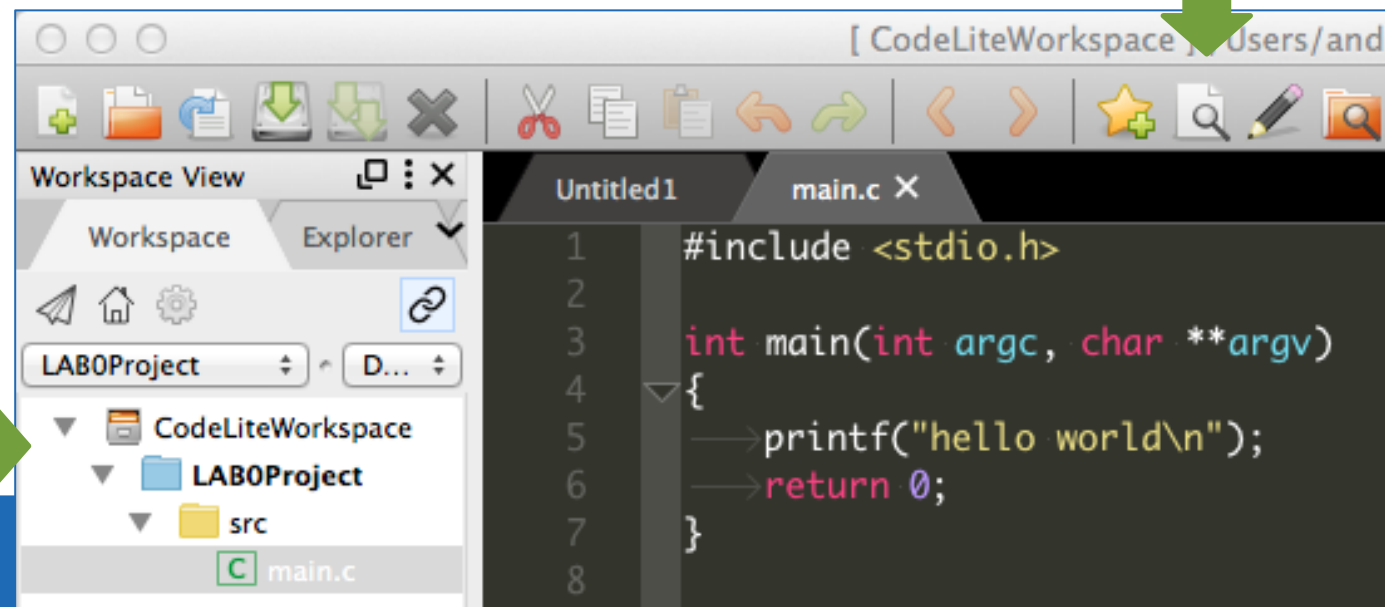


Använda CodeLite...



Välj nu:
File/New/
New project...

...som skapar
ett projekt med
en källkodsfil!



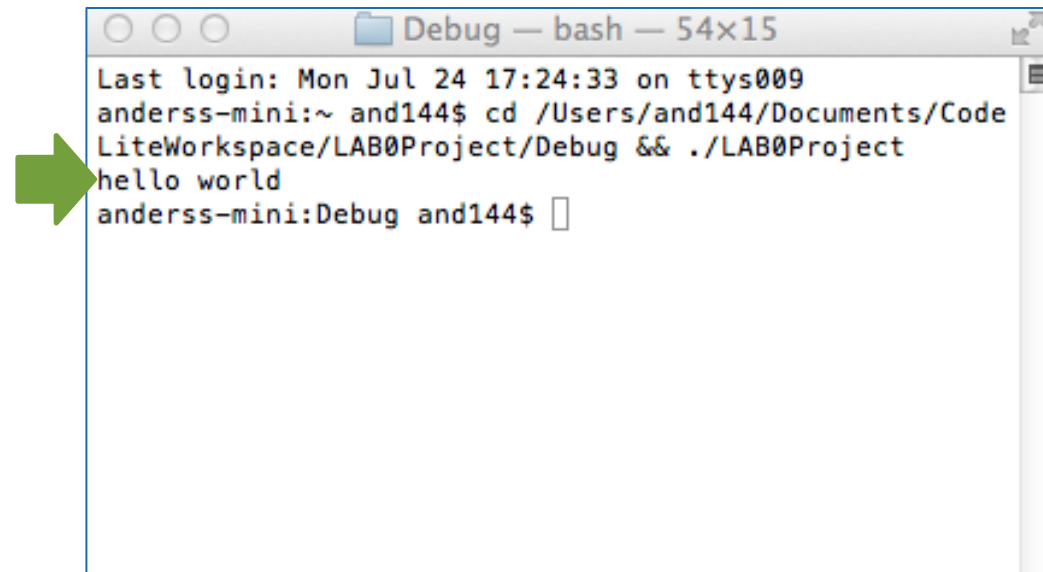


Använda CodeLite

Välj nu från **Build** menyn...

...**Build Project**, och därefter

...**Run**, då ska ett nytt fönster öppnas med bl.a. texten **Hello World!**



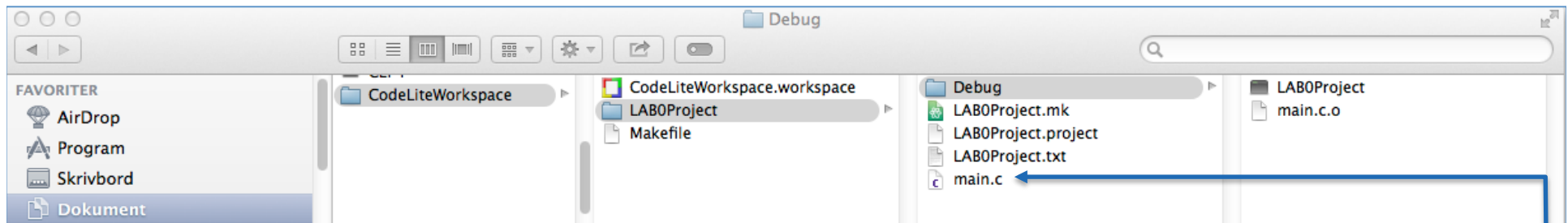
```
Debug — bash — 54x15
Last login: Mon Jul 24 17:24:33 on ttys009
anderss-mini:~ and144$ cd /Users/and144/Documents/Code
LiteWorkspace/LAB0Project/Debug && ./LAB0Project
hello world
anderss-mini:Debug and144$
```



Använda CodeLite (CL)

ÖVERKURS

Som ni kan se så skapar CL en avancerad filstruktur för varje projekt!



Mitt workspace... ..med 1 folder/projekt... ..folder för byggresultat... ..körbar fil m.m
& projektfiler (LAB0Project)
& källkoden main.c

Det finns inget som hindrar,
och det kan vara bra att
kunna, att kompilera i
kommandotolken...
(Tips: F2 om OS)

```
anderss-mini:Debug and144$ ls
LAB0Project      main.c.o
anderss-mini:Debug and144$ cd ..
anderss-mini:LAB0Project and144$ ls
Debug            LAB0Project.project      main.c
LAB0Project.mk   LAB0Project.txt
anderss-mini:LAB0Project and144$ gcc main.c -o main
anderss-mini:LAB0Project and144$ ls
Debug            LAB0Project.project      main
LAB0Project.mk   LAB0Project.txt          main.c
anderss-mini:LAB0Project and144$ ./main
hello  world
anderss-mini:LAB0Project and144$
```



Lab #3: Hello World!

Du ska installera CodeLite IDE, och utveckla ett lite klassiskt c-program!

("Hello World" från föregående sida!)



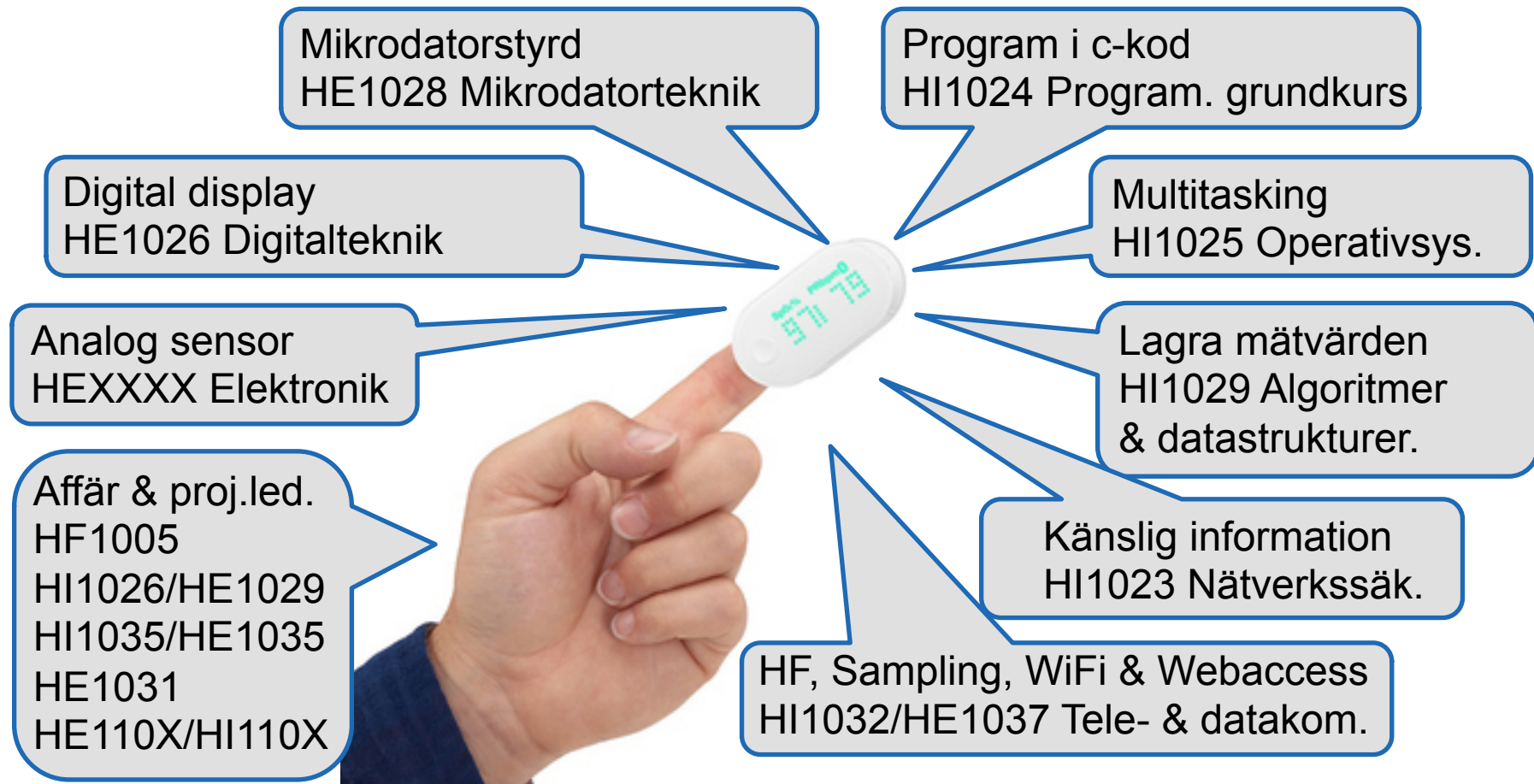
**A Powerful IDE for
C/C++
Supporting all major
compilers**

Grattis!

Du och din dator är nu klara för grundkurs programmering HI1024!



Din Innovation!



Exempel: The pulsmätare!



Rev history

150507 0.1 AC	Skapad.
150518 0.3 AC	Internet-delen nästan klar.
150522 0.5 AC	Första utkastet där allt är med på något sätt.
150823 1.0 AC	Expanderat Wiki-delen med stycke om att uppträda som ingenjör.
160615 1.1 AC	Inkluderat programutvecklingsmetodik som flyttats från föreläsning #2.
160704 2.0 AC	Uppsnyggat för HT16.
170709 3.0 AC	Uppdaterad för ny IDE Codelite.