KTH ROYAL INSTITUTE
OF TECHNOLOGY
Stockholm
Sweden

DD1301 & DD1337
Understanding Web Pages

Ric Glassey
glassey@kth.se
2015

Grading for DD1301

Grading for DD1301

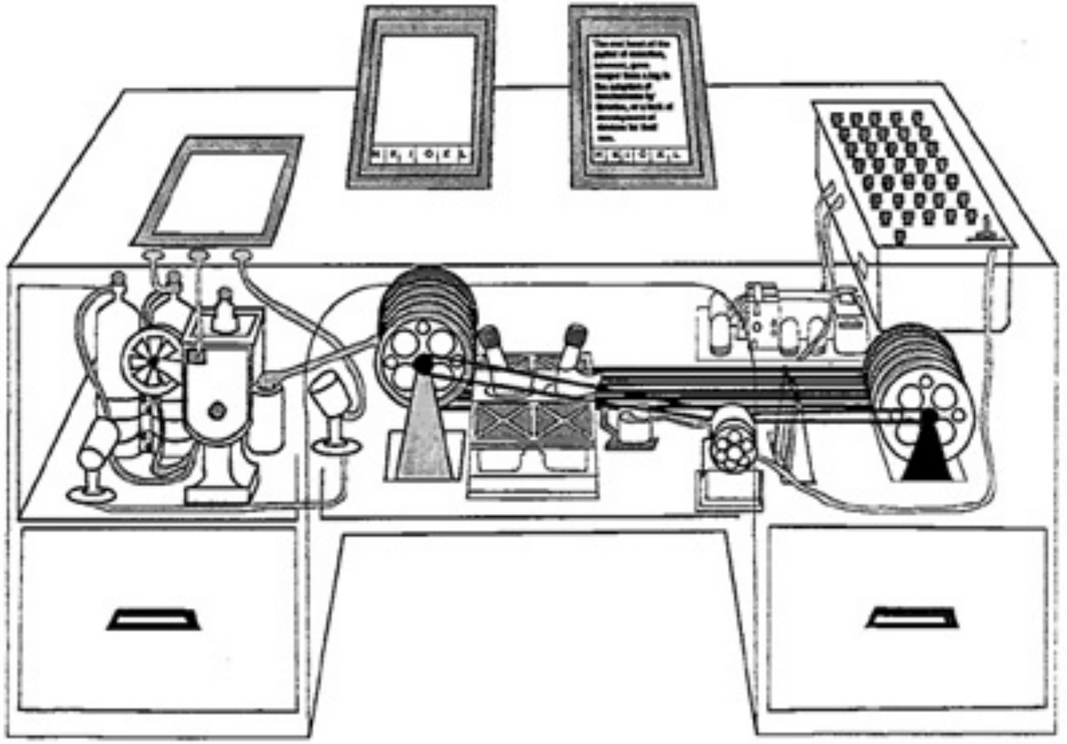
- You must have completed:
  - Introduction to Linux OR Advanced Intro to KTH systems
  - Latex Lab OR HTML exercise

- Lab assistants will check your work

- Grades will appear on RAPP soon after

A brief history of the web…

1945 – Vannevar Bush wrote about the Memex in “As we may think”



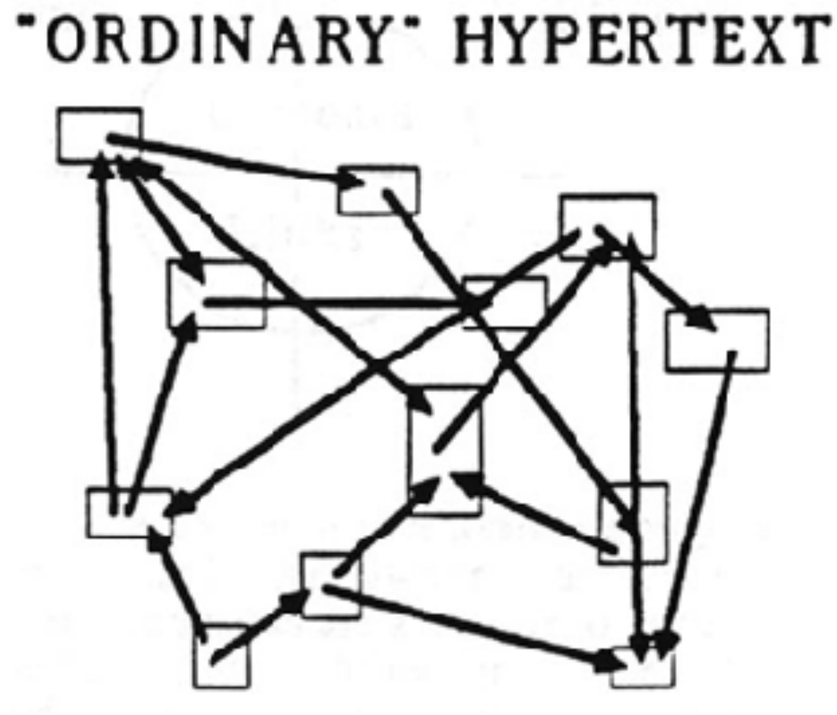
AS WE MAY THINK

A TOP U. S. SCIENTIST FORESEES A POSSIBLE FUTURE WORLI
IN WHICH MAN-MADE MACHINES WILL START TO THINI

By VANNEVAR BUSH

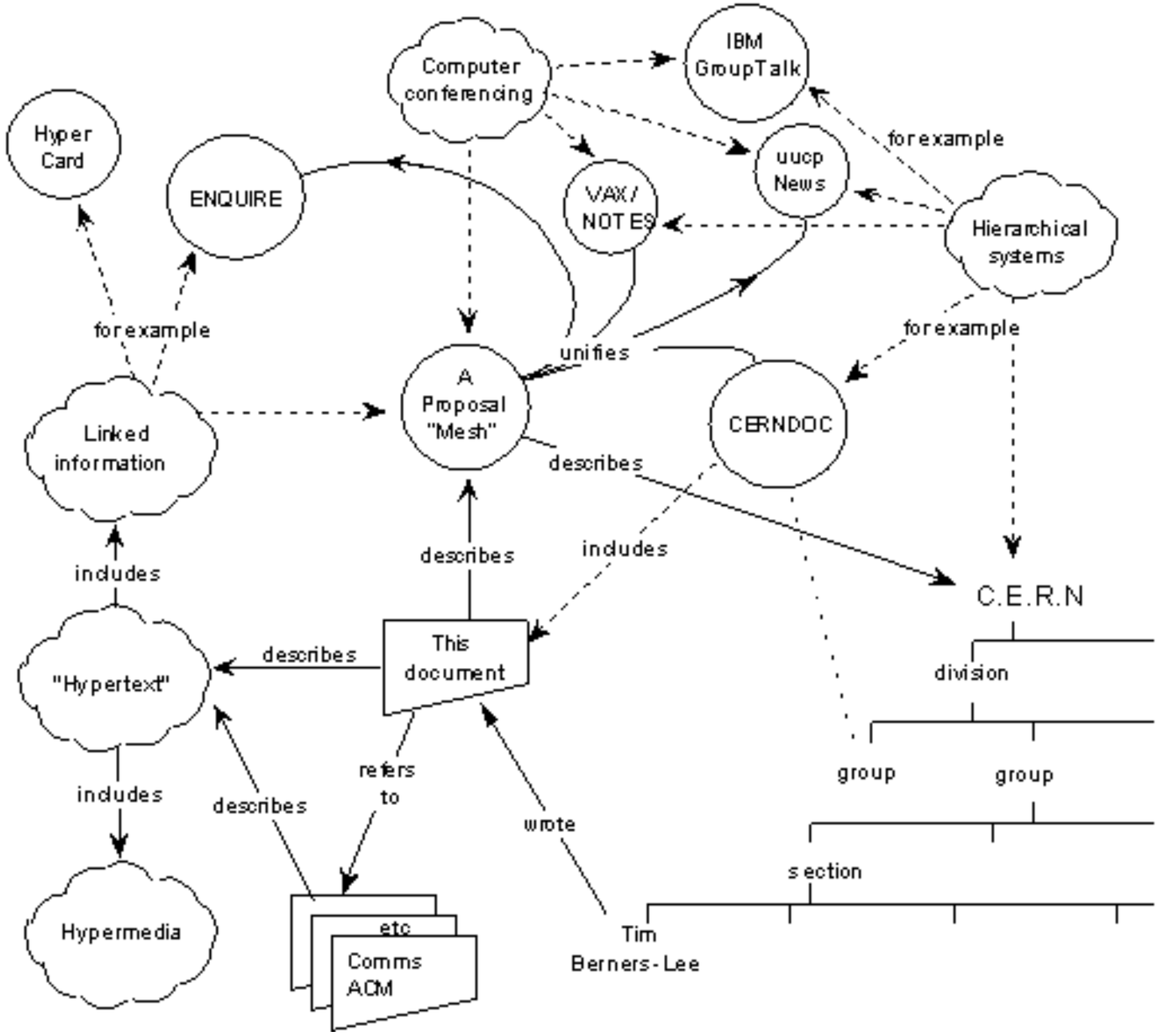
1963 – Ted Nelson invents the terms hypertext and hypermedia

- Concept of linked content
- Concerned with the complex nature of the creative impulse
- “Everything is interwingled”



1980's – Tim Berners Lee experiments with hypertext to build information systems

A reviewer commented:
"Vague, but exciting"



1990 – TimBL submits proposal at CERN for a WorldWideWeb for scientists and builds the 1st Web Server

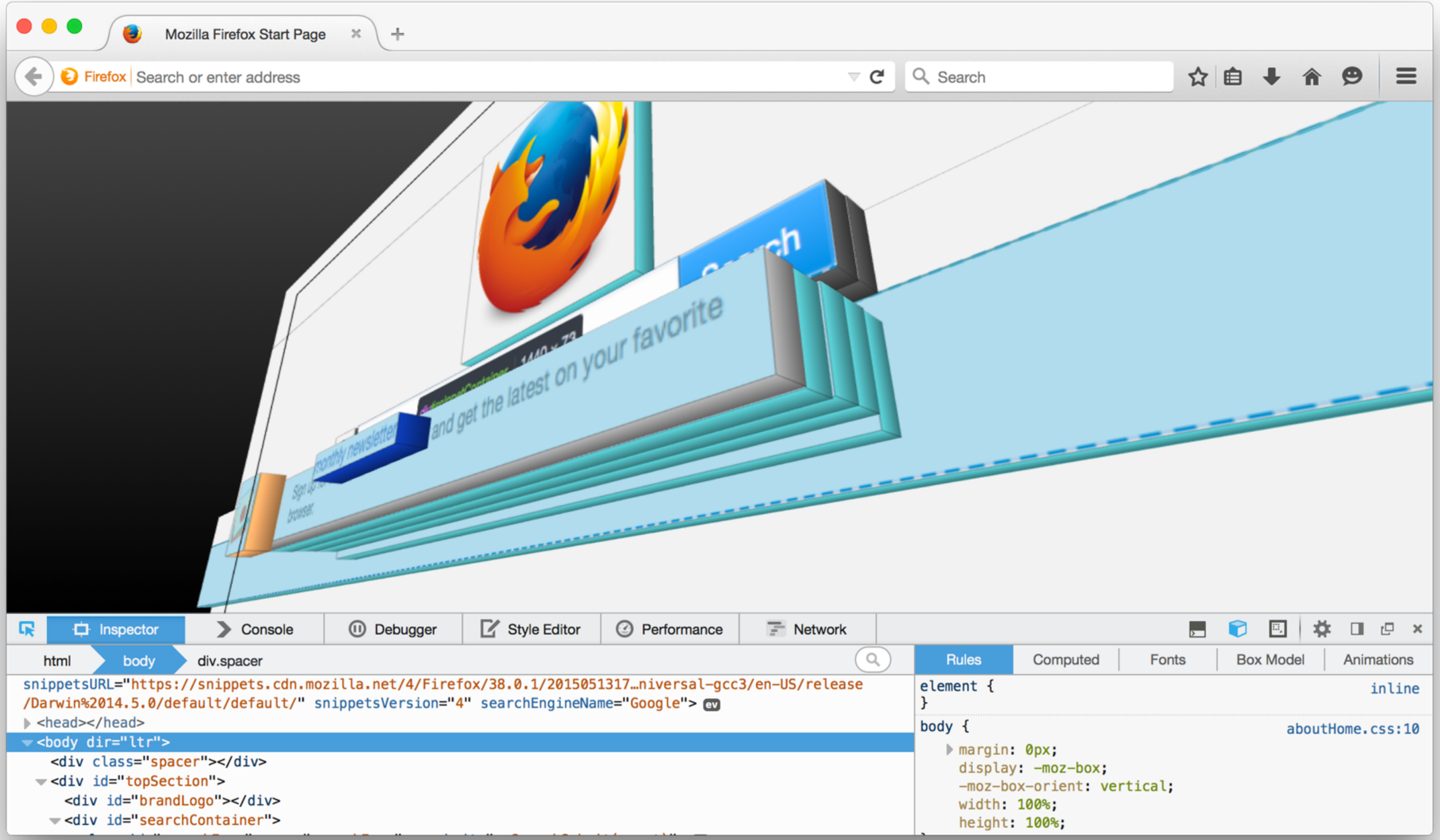
And we thanked him with this…

Behind the Web Page

Exploring HTML

Firefox Developer Tools Demo

- Launch Firefox (other browsers have similar tools)
  - Goto Tools / Web Developer
  - Select Inspector and observe mousing over page elements
  - Select the Network Tab in the Inspector Panel
  - In Inspector Panel Settings, switch on the 3D mode (may differ according to your version of FireFox)


Mozilla Firefox Start Page
Firefox
Search or enter address
Search
and get the latest on your favorite
Inspector
Console
Debugger
Style Editor
Performance
Network
html
body
div.spacer
Rules
Computed
Fonts
Box Model
Animations
snippetsURL="https://snippets.cdn.mozilla.net/4/Firefox/38.0.1/2015051317…niversal-gcc3/en-US/release
/Darwin%2014.5.0/default/default/" snippetsVersion="4" searchEngineName="Google">
<head></head>
<body dir="ltr">

element {
}
inline
body {
aboutHome.css:10
margin: 0px;
display: -moz-box;
-moz-box-orient: vertical;
width: 100%;
height: 100%;

Building Web Pages

HyperText Markup Language

HTML Demo

- Basic page structure
  - **doctype**, **html**, **head** and **body** tags
- Content
  - headers (**h1**, **h2**, …) and paragraphs (**p**)
  - bullet (**ul**), numbered lists (**ol**) and list items (**li**)
  - images (**img**) and links (**a**)
- Block and inline
  - dividers and sections (**div**, **section**) and content spanning (**span**)

```
<!DOCTYPE html>
<html>
<head>
     <title>My First Page</title>
</head>
<body>
<section>
     <h1>Big Title</h1>
     <p>Lorem ipsum dolor sit amet, consectetur <a href="www.google.com">adipisicing</a> elit, sed do eiusmod
     tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,
     quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo
     consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse
     cillum dolore eu fugiat nulla pariatur. Excepteur <span style="color:red;">sint occaecat cupidatat non
     proident, sunt in culpa qui officia</span> deserunt mollit anim id est laborum.</p>

     <ol>
          <li>first item</li>
          <li>second item</li>
          <li>third item</li>
     </ol>

     <ul>
          <li>First item</li>
          <li>Second item</li>
          <li>Third item</li>
     </ul>
</section>
<div>
     <h2>A Less Big Title</h2>

     <img src="cat.jpg">
</div>
</body>
```

Adding Style

Cascading Style Sheets

Adding Style with CSS

- HTML alone is quite basic
- CSS declares a series of style rules to be applied to HTML
- Another language :-)

- Short Demo
  - Creating a **style** section
  - Declaring rules (**selector**, **property** & **value**)
  - Specifying what to style (**element**, **id** and **class**)

```
<!DOCTYPE html>
<html>
<head>
   <title>My First Page</title>
   <style type="text/css">
   h1 {
      color: blue;
      background-color: red;
   }

   #special {
      color: green;
      background-color: black;
   }
   </style>
</head>
```

Practical Advice

Content Management and Client-side Frameworks

* That page was uninspiring…*

- Speed, Style and Quality
- **Option 1: Use a Content Management System**
  - Example: create a Wordpress blog
  - Fast, hosted, with few technical skills required
  - Less control
- **Option 2: Use a Client-side framework**
  - Example: create a Bootstrap page
  - More control but more developer experience required

* Review *

- HTML is the markup language of the Web
- Most browsers can explore what makes a web page
- HTML alone is quite basic; CSS adds style

- In practical terms, there are better ways to create web pages
  - Use a hosted Wordpress site
  - Use a client-side framework