DAT076/DIT126 Web Applications 2024

Who We Are

Course responsible and Examiner

Robin Adams <robinad@chalmers.se>

Teaching assistant

Wincent Stålbert Holm <wincenth@chalmers.se>

The Aim of This Course

- A project-based course in which you build a full-stack Web application
  - Full-stack = browser + server + database (+ anything else necessary)
- Teaching uses the following stack:
  - NodeJS – a JavaScript runtime designed for Web servers
  - ExpressJS – a JavaScript library for writing Web servers
  - ReactJS – a JavaScript library for building user interfaces
  - MySQL – relational database

Goal of the course

Design, implement, deploy, test and debug a basic web application

- Project-based course
- Lab assignments (first half of course) guide you through the first steps of building a full-stack Web application
- Lectures and readings explain the concepts and technologies encountered
- Weekly meetings with supervisor (from week 3)

Course philosophy

You will learn a large number of technologies

- HTTP
- HTML
- CSS
- Bootstrap
- NodeJS
- ExpressJS
- JavaScript
- TypeScript
- Persistence
- MySQL
- ReactJS
- …….

Course philosophy

You will learn a large number of technologies

- Eight weeks is not long! Lectures cannot cover more than the basics
- We encourage you to research and experiment
- You are ***learning how to learn a framework***
    - Learn to read docs and specs
    - Learn to debug
    - Debugging and Googling by supervisors (and lecturer!) is not a waste of your time!

Changes from last year

- Move to *contract-based grading* – explicit requirements for grades 3, 4, 5 in each section
- Emphasis on security and accessibility from early in the course

Our Slack Workspace

- Slack workshop for group communication (see Canvas for link)
  - Use #general for general Q&A
  - Each lab assignment has a channel for questions about that lab
  - If you PM me a question, I will share it (anonymously) on #general unless you ask me not to.
  - #random for off-topic chat (keep it professional!)
- During lab sessions:
  - When you want to speak to a supervisor, post an invitation to #supervision-help-requests with either room number or Zoom link
  - We will answer the requests in order
  - Please do not use this channel for anything else

Course website

https://chalmers.instructure.com/courses/22446

Structure of the course

- Group project to be completed in groups of 3-4 by end of the course
- Lectures plus assigned readings over the course
- 4 mandatory laboratory assignments, starting from the first week of the course. Assignments are not graded, but must be completed and passed. (Lab 0 is an exception – not assessed.)
- Lab 0 can be done individually. Labs 1-4 are done in project groups
- Weekly supervised group meetings with your supervisor starting from week 3
- Presentation during exam week to demonstrate your application to us and the class

Group project

- Please start forming groups today - work in groups of 3-4 people (ask me about exceptions)

- Use channel #seeking-project-partners

- During week 3 of the course you will meet your supervisor for the first time. Discuss your project ideas with your supervisor and make sure it’s a good fit for the course - anything that has a lot of user interaction and requires persistent storage is generally a perfect fit

- Start lab 0 now

Group project grading criteria

When we grade the project we look at a number of things

- Design, code quality and style (25%)
  Good software engineering principles
- Functionality (20%)
  Number and complexity of use cases
- Testing (20%)
  Coverage by automated tests
- Security (20%)
  Conformance to OWASP Application Security Verification Standard
- Accessibility (10%)
  Conformance to WCAG 2.3
- Documentation (5%)

Individual grading

In addition to the grading of the project we also grade the individual contribution to the project from each student

- Make sure you are active in the project from the beginning and continuously part of the programming and design decisions. Let us know early if a group is not working well together.

- We will judge your contribution to the project by:
  - Git commits
    - **If you use pair programming**: be sure to add the other person as a co-author on the git commit
  - Responsibilities of each person in project report
  - Anonymous evaluation of group members
- Final grade for the course is usually project grade, sometimes plus or minus 1

Group project documentation

When handing in the project you should provide some documentation in PDF format containing the following

- User manual with installation instructions
- Description of the project, including the technologies used
- A list of use cases
- A description of the design, including API specification
- Anything else you think might help us to better grade your project

GitHub Copilot Etc.

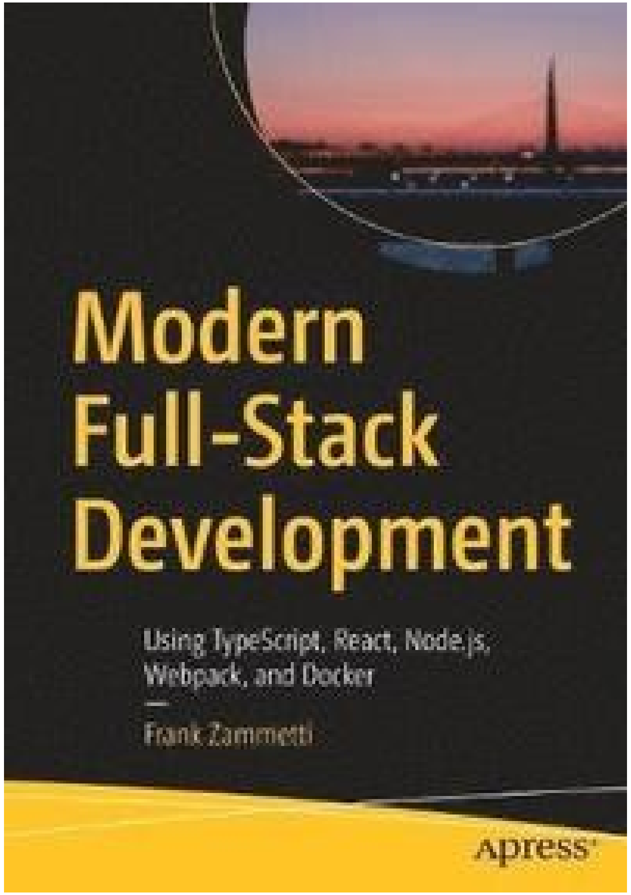
Chalmers and GU guidelines on using AI tools (Copilot, GPT, etc.)

- None

Guidance for this Course

- Allowed, but recommended against.
- No guidance given on how to use it.
- **Warning:** Github Copilot produces correct code only 43% of the time. It introduces security vulnerabilities 40% of the time.

Resources

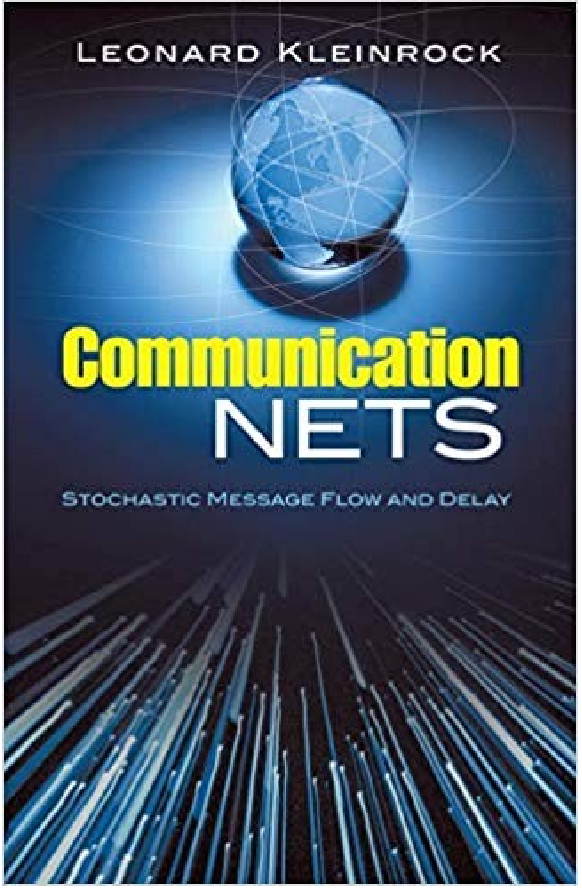


- The course book is *"Modern Full-Stack Development"* by Frank Zametti (ISBN: 978-1-4842-5738-8)

- Use the web and Google for anything not covered by the book (official docs > blogs and tutorials)

- As the course progresses I will put additional links to articles, tutorials and documentation

History of the Internet



1964

- Leonard Kleinrock of MIT publishes *“Communication Nets: Stochastic Message Flow and Delay”.* Available in a 2007 edition (ISBN: 9780486458809)
- Paul Baran applies Kleinrock’s theory for secure voice communication in the military
- The purpose? To create a communications network that can survive partial destruction (nuclear attack?)

History of the Internet

Paul Barans implementation is later used in ARPANET

- *Advanced Research Projects Agency (ARPA)*
- Originally created in February 1958 by Eisenhower in response to the Soviet Union launching *Sputnik 1* in 1957
- Late 1960s: ARPANET *(Advanced Research Projects Agency Network) conceived*
- Packets sent across wires with data shared by multiple simultaneous sessions. Increases network efficiency and robustness

What was the ARPANET?

For each of these three terminals, I had three different sets of user commands. So if I was talking online with someone at S.D.C. and I wanted to talk to someone I knew at Berkeley or M.I.T. about this, I had to get up from the S.D.C. terminal, go over and log into the other terminal and get in touch with them.…

I said, oh man, it's obvious what to do: If you have these three terminals, there ought to be **one terminal that goes anywhere you want to go** where you have interactive computing. That idea is the ARPAnet.

– Bob Taylor (1967)

What was the ARPANET?

We set up a telephone connection between us and the guys at SRI (Stanford Research Institute)

We typed the L and we asked on the phone, *"Do you see the L?"*

"Yes, we see the L," came the response.

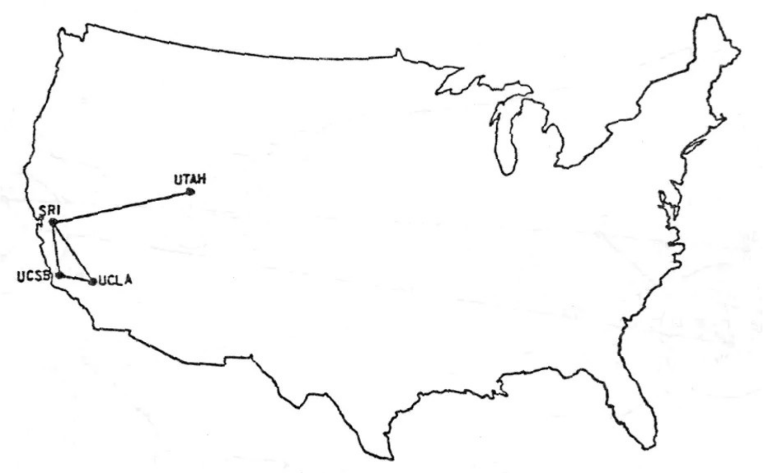
We typed the O, and we asked, *"Do you see the O."*

"Yes, we see the O."

Then we typed the G, and the system crashed…

Yet a revolution had begun…
(Gregory Gromov, 1969)

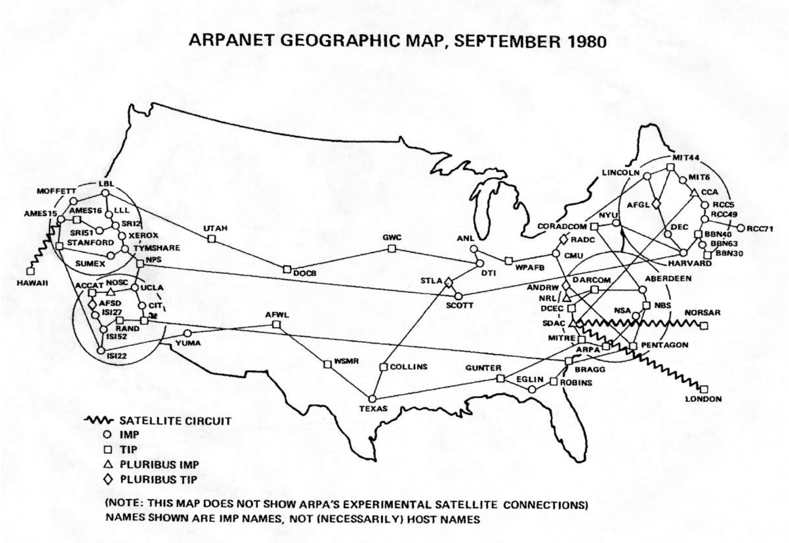
What was the ARPANET?



The ARPANET in December 1969

- 1969: Network consists of 4 computers
- 1974: First implementation of TCP/IP and packet switching to allow multiple computers to communicate on a single network. First use of *Internet* to describe the internetworked system
- Data consists of a header, used by networking hardware to direct the packet to its destination, and a payload, which is extracted and used at the destination

What was the ARPANET?



ARPANET GEOGRAPHIC MAP, SEPTEMBER 1980

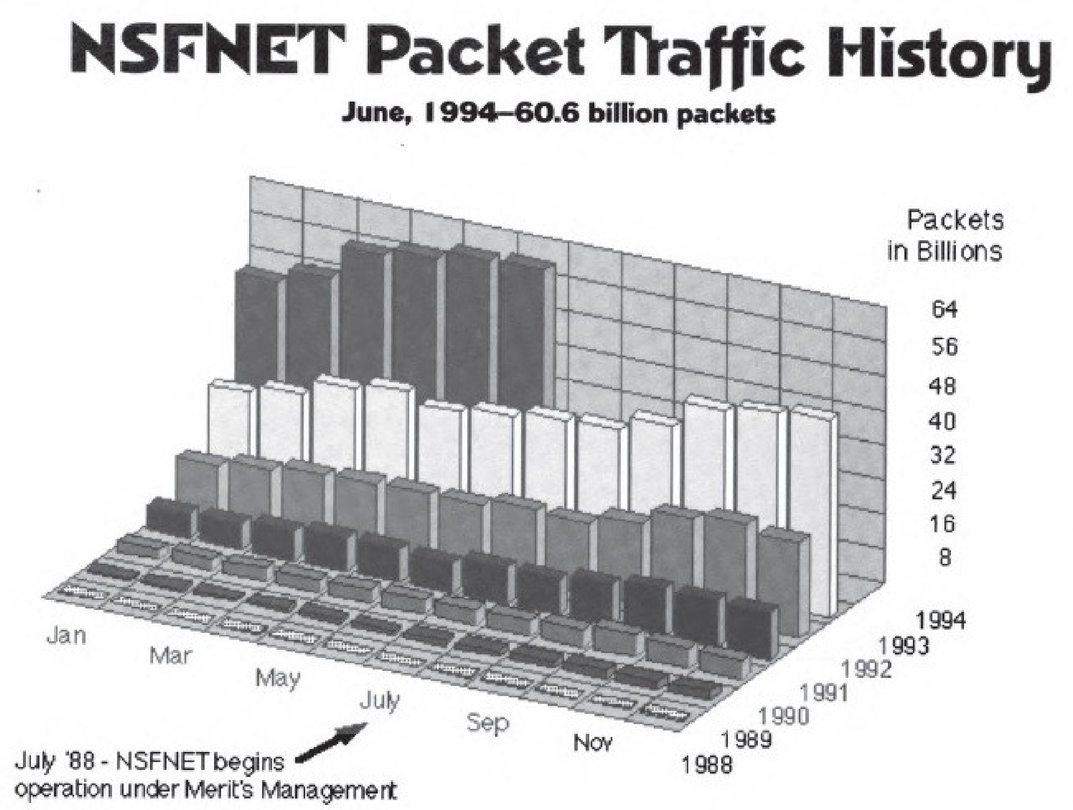
(NOTE: THIS MAP DOES NOT SHOW ARPA'S EXPERIMENTAL SATELLITE CONNECTIONS)
NAMES SHOWN ARE IMP NAMES, NOT (NECESSARILY) HOST NAMES

- Grew very quickly during the 1970’s with many institutions joining. The network gained a direct link to London via a satellite Link

- 1985: NSFNET project initiated - sponsored by the *National Science Foundation (NSF)* from 1985 to 1995

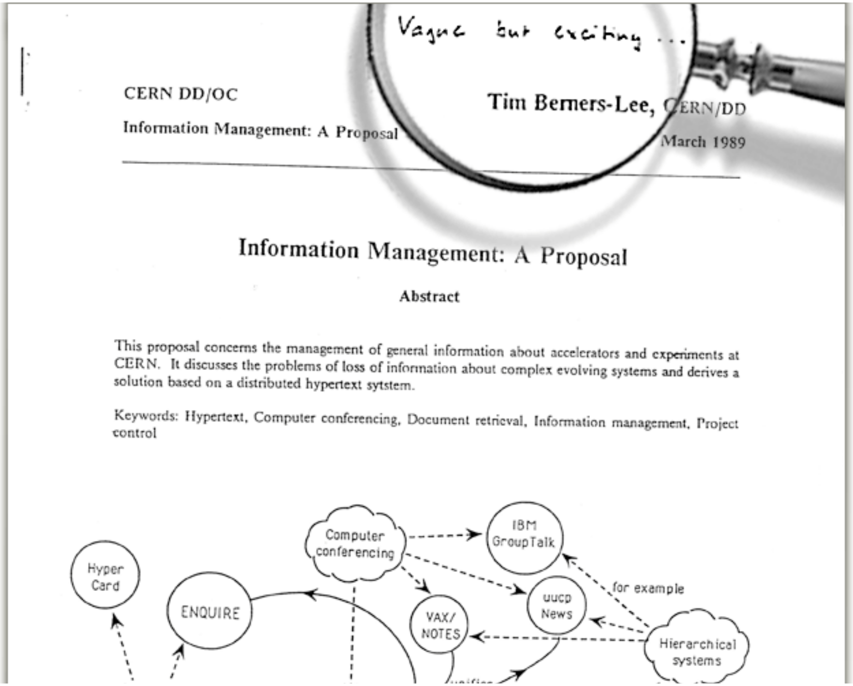
- ARPANET decommissioned in 1990 in favor of the NSFNET (becoming the Internet)

NSFNET starts growing



- July 1988: NSFNET is officially launched
- 1991: NSF removes access restrictions, resulting in rapid growth of the commercial ISP business

The World-Wide-Web



Vague but exciting . . .

CERN DD/OC

Information Management: A Proposal

Tim Berners-Lee, CERN/DD

March 1989

Information Management: A Proposal

Abstract

This proposal concerns the management of general information about accelerators and experiments at CERN. It discusses the problems of loss of information about complex evolving systems and derives a solution based on a distributed hypertext sytstem.

Keywords: Hypertext, Computer conferencing, Document retrieval, Information management, Project control

- 1989:
- “In those days, there was different information on different computers, but you had to log on to different computers to get at it. Also, sometimes you had to learn a different program on each computer. Often it was just easier to go and ask people when they were having coffee…” - Tim Berners-Lee

The World-Wide-Web

- 1990: First running web server developed by Tim Berners-Lee
- A NeXT machine at CERN running the NeXTSTEP operating system
- **CERN httpd** was released to the public in June 1991
- Still available here, www.w3.org/Daemon

www.w3.org/People/Berners-Lee/1991/08/art-6484.txt

The World-Wide-Web

- 1990: First running web server developed by Tim Berners-Lee
- A NeXT machine at CERN running the NeXTSTEP operating system
- **CERN httpd** was released to the public in June 1991
- Still available here, www.w3.org/Daemon

www.w3.org/People/Berners-Lee/1991/08/art-6484.txt

Public adoption of the Internet

- 1993: AOL spreads commercial Internet access to millions of people (dubbed the eternal september)

- Usenet was previously mostly restricted to universities and research institutions. Every September, many incoming students would acquire access to Usenet for the first time, taking time to become accustomed to standards of conduct and "netiquette". After a month or so, these new users would either learn to comply with the social norms or tire of using the service

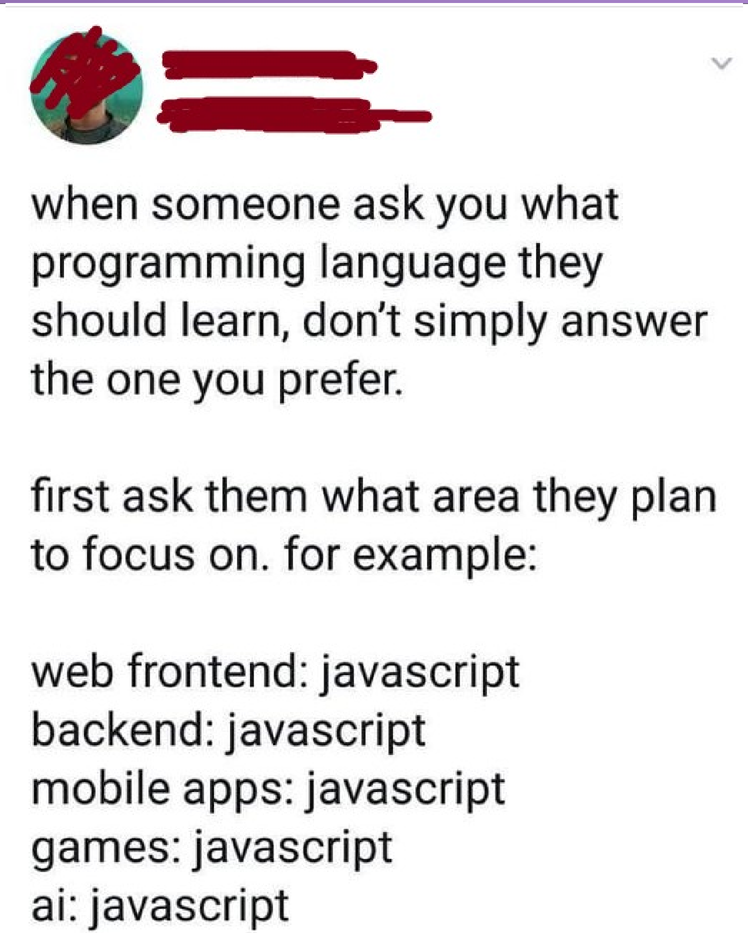
JavaScript

- 1995: Brendan Eich hired by Netscape to create a programming language for dynamic Web pages
- Prototype created in 10 days
- Not related to Java (except syntax designed to look like Java syntax)
  - Pure marketing – Java was new, hot and trendy in 1995
- Multi-paradigm language: functional, imperative and object-oriented
- Dynamically typed
- Lots of weird features we are now stuck with (null vs undefined vs NaN)

JavaScript

Activate Your Web Pages
6th Edition
Covers ECMAScript 5 & HTML5
JavaScript
The Definitive Guide
O'REILLY®
David Flanagan
Unearthing the Excellence in JavaScript
JavaScript:
The Good Parts
O'REILLY®
YAHOO!. PRESS
Douglas Crockford

JavaScript



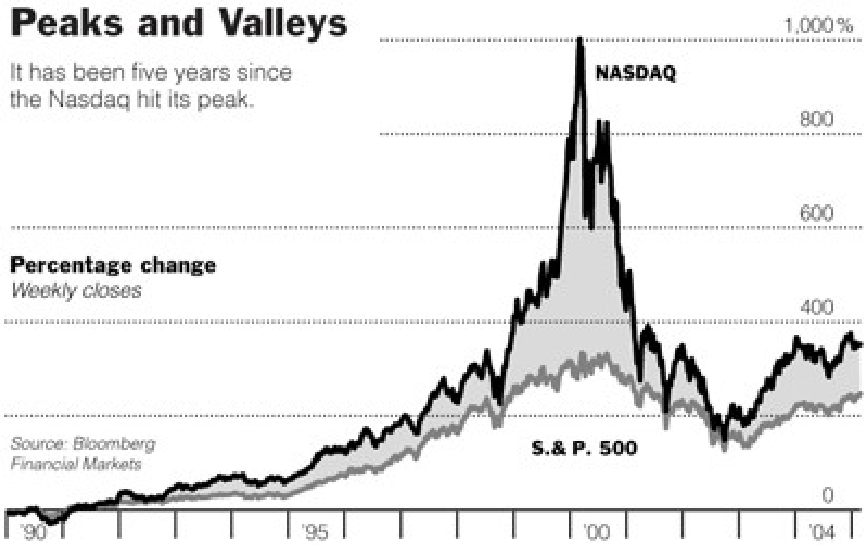
[redacted]

when someone ask you what programming language they should learn, don't simply answer the one you prefer.

first ask them what area they plan to focus on. for example:

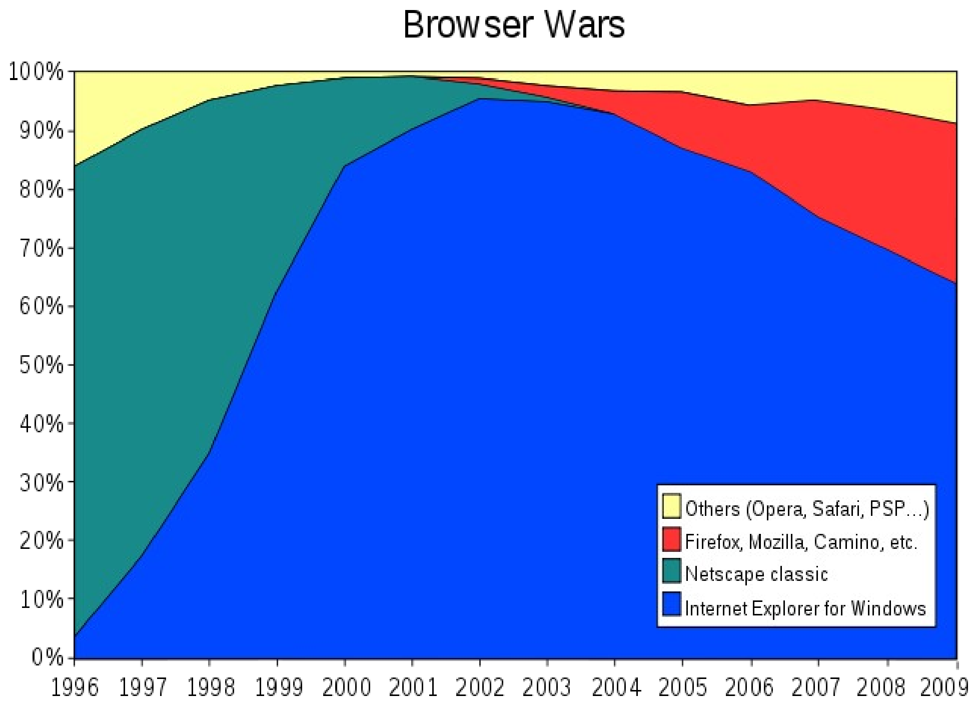
web frontend: javascript
backend: javascript
mobile apps: javascript
games: javascript
ai: javascript

The dot-com boom



- 1995-2000: Investors throw money at anything related to the Internet. Some shares can rise 2000% in one year
- 2000-2003: NASDAQ market crashes Loses $5 trillion (roughly 78% of its value)

The Browser Wars



- The 1990s: different browsers had different features

- Web sites would work on some browser but not on others

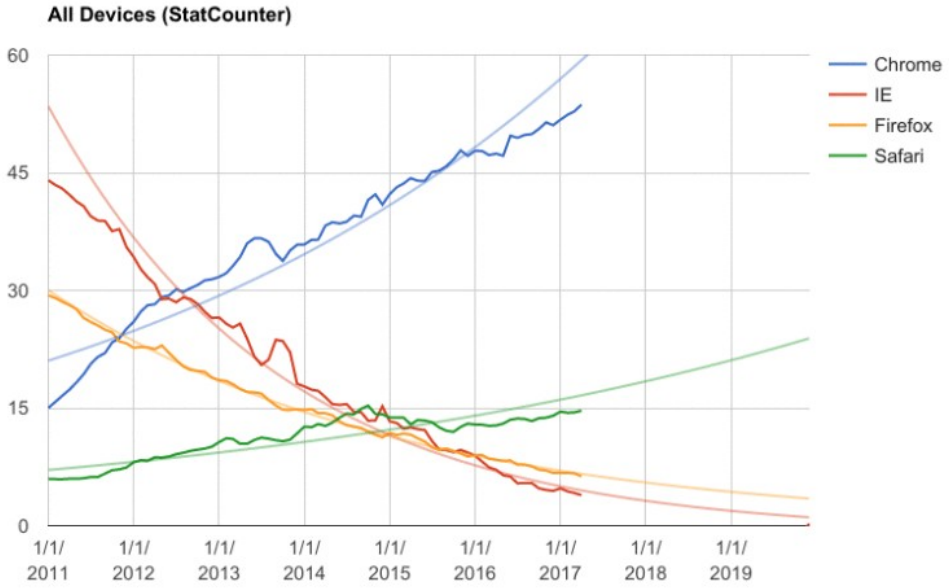
- In the very early beginning, Mosaic was popular. Eventually Netscape Navigator and Internet Explorer became the most popular alternatives

The Browser Wars get Ugly

- Some people claim that Microsoft intentionally sabotage competing browsers by relying and promoting features specific to Internet Explorer

- People get really passionate, leading to *“browser activists”* doing campaigns to ridicule the competitor.

Browser wars continue



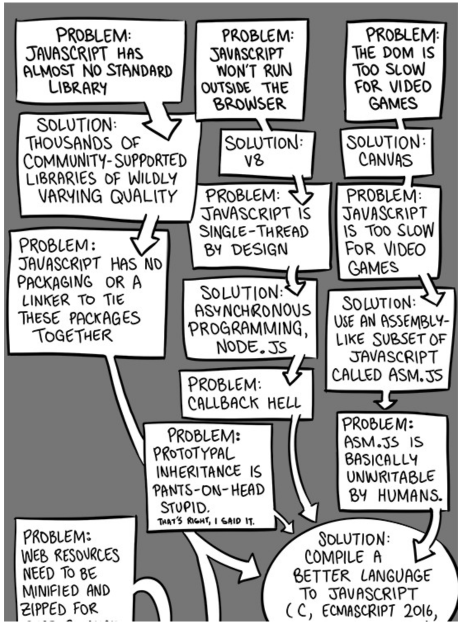
- Apple, Firefox, Opera fight back
- 2008: Google Chrome is released. With more and more interactivity being introduced, the web rapidly transitions from a repository of documents to a repository of applications
- 2020: Google Chrome has about 70% market share

NodeJS and V8

- Google Chrome (2008) – based on a JavaScript run-time environment V8
- We could use that for other things...
- NodeJS (2009) – written by Ryan Dahl
  - presentation at European JSConf received a standing ovation
  - back-end JavaScript run-time environment that runs on V8 engine
  - compiles JavaScript to machine code as it runs (just-in-time compilation)
  - asynchronous, event-driven, single-threaded

Web development is messy

The Web is now about apps. The problem is that none of the core technologies were designed with apps in mind


PROBLEM: JAVASCRIPT HAS ALMOST NO STANDARD LIBRARY
SOLUTION: THOUSANDS OF COMMUNITY-SUPPORTED LIBRARIES OF WILDLY VARYING QUALITY
PROBLEM: JAVASCRIPT HAS NO PACKAGING OR A LINKER TO TIE THESE PACKAGES TOGETHER
PROBLEM: JAVASCRIPT WON'T RUN OUTSIDE THE BROWSER
SOLUTION: V8
PROBLEM: JAVASCRIPT IS SINGLE-THREAD BY DESIGN
SOLUTION: ASYNCHRONOUS PROGRAMMING, NODE.JS
PROBLEM: CALLBACK HELL
PROBLEM: PROTOTYPAL INHERITANCE IS PANTS-ON-HEAD STUPID. THAT'S RIGHT, I SAID IT.
PROBLEM: THE DOM IS TOO SLOW FOR VIDEO GAMES
SOLUTION: CANVAS
PROBLEM: JAVASCRIPT IS TOO SLOW FOR VIDEO GAMES
SOLUTION: USE AN ASSEMBLY-LIKE SUBSET OF JAVASCRIPT CALLED ASM.JS
PROBLEM: ASM.JS IS BASICALLY UNWRITABLE BY HUMANS.
PROBLEM: WEB RESOURCES NEED TO BE MINIFIED AND ZIPPED FOR
SOLUTION: COMPILE A BETTER LANGUAGE TO JAVASCRIPT (C, ECMASCRIPT 2016,

- NPM and Yarn
- Sass
- The TypeScript language and transpiler

Web standards

- W3C – World Wide Web Consortium

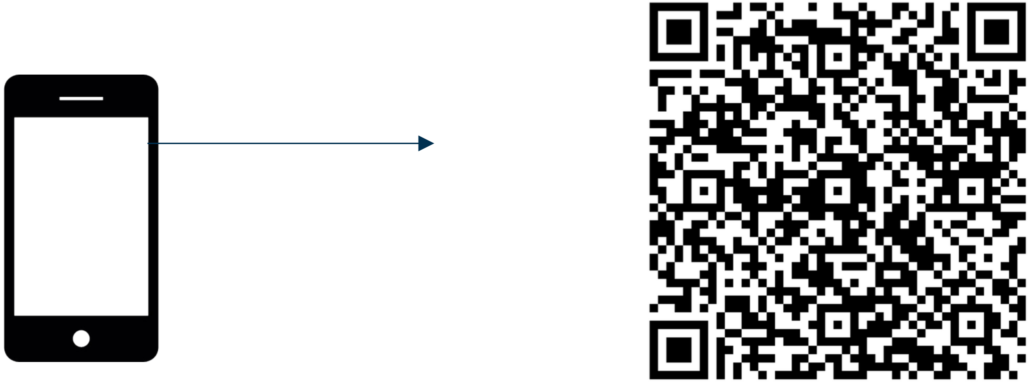
- WHATWG - Web Hypertext Application Technology Working Group. Founded in 2004 by individuals from Apple, Mozilla and Opera Software to push W3C to develop HTML with support for interactivity

- HTML5 and CSS3 were designed with better document structuring and interactivity in mind

- The process will most likely always be a work in progress. Different browsers support different features and proposals of the standard. A good resource to use when checking if you should use a feature is caniuse.com

EDUCATIONAL SUPPORT IS LOOKING FOR A NOTE-TAKER

- You help other students who would otherwise find it difficult to take up their studies in a good way.
- You also help yourself as your own notes make you perform better and remember longer.
- Through the work, you do you will learn to plan and structure your notes, which you will benefit from even after studying.
- You also get paid for your work as a note-taker.



https://www.chalmers.se/en/education/student-support/studying-with-disabilities/note-taking-support/#for-those-who-want-to-become-a-note-taker