COMP 307

Principles of Web Development

Course Introduction

Professor Joseph Vybihal

COMP 307
Principles of Web Development

Contents

Lecture Outline

Introduction to the course

Trending and History

About Web Development

About this course

Course Introduction

Coordinates

- Joseph Vybihal
  - ENGMC 323
  - www.cs.mcgill.ca/~jvybihal
- Communication
  - Email: joseph.vybihal@mcgill.ca
  - myCourses discussion board (experiment)
- Office hours:
  - By appointment (email me)
  - Hours:
    - M 3PM, W 1PM, F 9AM ENGMC 323 (starting next week)

Do you have previous experience

- I am a beginner
  - This course is for you !
  - We start from first principles and go pretty far
- I am an intermediate web developer
  - It will formalize some concepts that you took for granted when programming
  - It will introduce you to advanced techniques
- I am an advanced web developer
  - It will show you behind-the-scenes concepts that developers take for granted
  - You will learn software engineering techniques practical for web development
  - But, you will already be familiar with some topics

Prerequisites & Expectations

- Requirements
  - Prerequisite 206 (basic systems knowledge)
  - Co-requisite 303 (MVC-1, MVC-2, Singleton, Observer, etc)
    - Design patterns: https://sourcemaking.com/design_patterns
- Expectation
  - A lot of programming
  - 1st half of course assumes you have very little, knowledge of the Internet
  - The course attempts to survey a large portion of the Internet landscape. “Make you a real web developer”
  - Students are expected to learn some things on their own.

Complementary Courses

- Databases
  - Storing information online
- Operating Systems
  - The features your webserver will have depends on the OS
- Networks
  - Your internet connection is only as good as your network connection
- Cryptography
  - If your data is not secure, then your data is public

COMP 307
Principles of Web Development

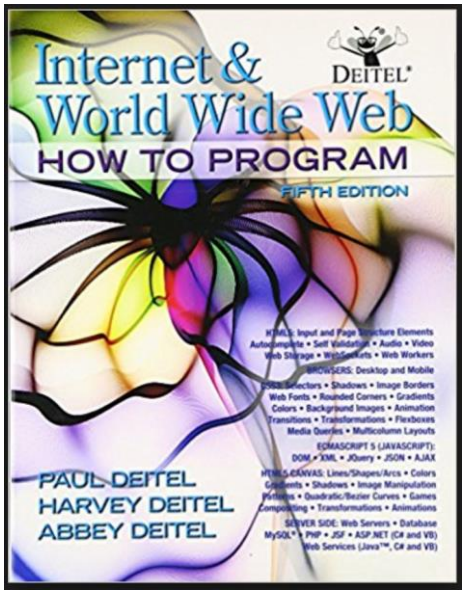
No Textbook

Readings: https://www.w3schools.com/ and https://www.tutorialspoint.com/index.htm

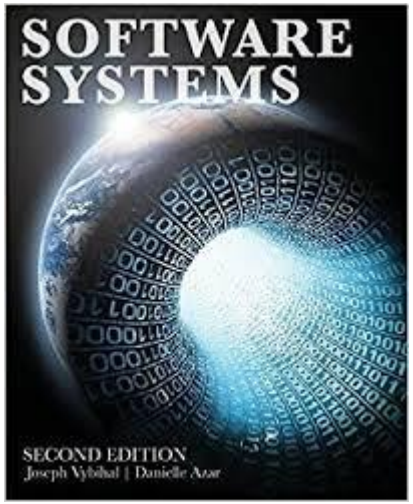
Other helpful texts:

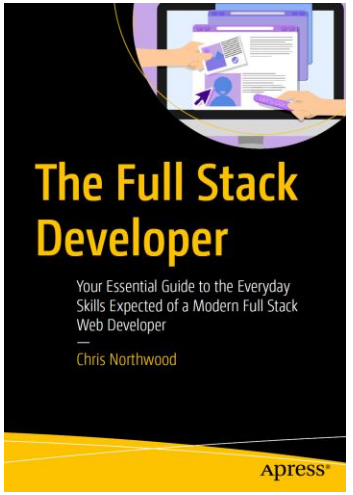
(Readings provided)

Internet and World Wide Web How To Program
Deitel – ISBN 978-0-13-215100-9
(online copy – myCourses)



Software systems
Kendall hunt
ISBN 978-0-7575-9514-1





Online copy - myCourses

Evaluation

- 6 Mini Assignments . . . 30%
  - 20%/day (max 2), README.txt waiver (1 time)
- Team Project 25%
  - Code quality
- Project Demo 25%
  - Execution quality
- Midterm Exam 20%

No final exam

Teaching

- Lectures
  - Theory: slides, whiteboard, and readings
  - Practice: in-class coding and demonstrations
  - Sample code to start with – Bring your laptop
- Expected but not graded
  - Self study portions (readings)
  - Guided **tutorials** (which can be done on your own)
- Multiple Mini Assignment
  - Small and to the point to build your skills
- Large project
  - Team of developers are required (do something real)

Previous Projects

- Winner built the SOCS website!
  - www.cs.mcgill.ca (SOCS accepted it)
- Winner built a TA Management system
  - SOCS did not accept it ☹
  - But we are upgrading it with a mobile interface ☺

Can't wait to see who will be this year's winner!!

Project

- Team of 3 or 4
- Comprehensive website built to a specification
- Project introduced midway through the course
- Best project will be crowned as winner
  - This year not SOCS, maybe open source…

Start making friends!

The Competition

- The best 3 projects will be submitted to three CS professors for selection
  - Qualifications
    - Instructor selects the best 3
    - Team MUST agree to let the school use/modify their project
    - CS Trio reserves the right to NOT use any of the projects
- The instructor will announce who were the 3 best projects on myCourses at the end of the term
- Top 3 sent to the CS Trio of Judges

COMP 307
Principles of Web Development

Contents

Tentative Course Schedule

2 Lectures per Week

Lecture	Description	Work handed out
	Unit 1 – The Internet Landscape - Networks	
Aug 31	Course introduction. Internet history. What's trending. Front-end vs Back-end design. Development stacks: XAMPP, MEAN, DJANGO, MERN. Internet as an unsupervised network: properties and example run-time.	
Sep 5	Parts of a network. Packets. Why ASCII. Importance of encryption. Network protocols: Client-server vs peer-to-peer comm, connection protocol, HTTP Protocol, the REST protocol (intro). Wireshark.	Mini 1 – Wireshark and Protocols Lab A – Setup Wireshark
	Unit 2 – Frontend Internet Languages	
Sep 7	**Local development and the browser run-time environment** Local - HTML 4 and 5 (formatting)	
Sep 12	Local – CSS (styling, div layouts)	Mini 2 – wel
Sep 14	Local - JS (basics, at load, at events)	
Sep 19	Local – JS (DOM manipulation)	Mini 3 – wel
Sep 21	Local – HTML forms and Graphics, browser inspect.	
	Unit 3 – Frontend Design	
Sep 26	**#0 - Planning a website build:** Design considerations def. (elements, shape, flow, and layouts), SE techniques (requirements, plan, review) and case studies. Methods of working in a team effectively (manage & Q/C).	Mini 4 – We effective tea
Sep 28	**#1 – Multi-paged websites**: Static vs Fixed page design. Interactive design: hide/show, hover, menu, accordion, frames. Webkit (peek).	
Oct 3	**#2 – Dynamic websites**: display media considerations. Responsive design vs Liquid design. DIY methods.	Mini 5 – wel tive + dynan
Oct 5	Library vs Framework vs Environment (Engine). Performance. Peek: Bootstrap. Flexbox. Vue.js. Popularity and problems of environments like WordPress.	Lab B - Boot
Oct 6 to Oct 11	**STUDY BREAK**	
	Unit 4 – Servers: focus XAMPP and MERN (assumes you already know C, Pyth	
Oct 12	About webservers: The web development stack, examples. Server development issues: performance, load, response time, privacy. Positives: shared information, community. SOCS server: Apache. Using SOCS to create your website. Copying your local website to SOCS. Using Forms with CGI for server-side C, Python, and Bash communication.	Assumes: COMP 250 a
Oct 17	XAMPP with Apache and PHP. About flexible applications. Installing XAMPP locally. Apache configuration. Why PHP? Introduction to PHP programming.	Lab C – XAM
Oct 19	MERN (part 1): about the MERN stack and single paged websites. What Facebook needed to solve. Introduction to NodeJS.	Mini 6 – wel & MERN

UNIT 1 – The Internet Landscape
UNIT 2 – Frontend Internet Languages
UNIT 3 – Frontend Design
UNIT 4 – Servers

UNIT 5 – Backend Design
UNIT 6 – Security
UNIT 7 – Working on the project
(optional extra classes)

		Lab D - MERN
Oct 24	MERN (part 2): Introduction to React programming.	
	Unit 5 – Backend Design	
Oct 26	**#1 – Dynamic multi-paged website.** Design pattern: Model View Controller vs Observer. Server comms: jQuery, Ajax (synchronous vs a-synchronous). (A) Dynamic content: server-side query. (B) Dynamic page generation.	Assumes: COMP 303
Oct 31	**#2 – Single paged website.** Websites issues: cell phone-based websites & real-estate. (A) DIY with DIV+JS + hide/show. (B) Using React programming & MERN	
Nov 2	**#3 – Database-based website** (3-tier web application). (A) CSV vs Matter vs Database files (B) SQL vs Mongo (structure & performance) (C) Databases with Dynamic content & pages (D) Databases with state information (user, session) (E) Introduction to SQL programming	Lab E – SQL & Mongo
Nov 7	**#4 – API-based Websites.** No GUI. REST API implementation. Example: Google API. (A) Show REST packet. Programs that show the packet. (B) Testing with programs like postman.	**Project Handed Out**
	Unit 6 – Security	
Nov 9	Security issues: Man-in-the-middle, Denial-of-service. Privacy - Absence, Adwords, MMap. Example: SOCS file security. Browser cache and cookie security issues.	
Nov 14	Obfuscation using: routing (DIY, Slim framework, NodeJS), secrets (encryption and hashing passwords. 1-key vs public-key).	Lab F –browser cache and cookies
Nov 16	Practical security: How do we know when someone has logged in? JSON Web-tokens. Login/logout database (user & session data). Tickets (page / object access rights).	
	Unit 7 – Working on the project	
Nov 21	Optional overflow class	
Nov 23	Optional overflow class	**Midterm Exam**
Nov 28	No lecture (work on project)	
Nov 30	**Monday Schedule – No Lectures**	
Dec 5	No lecture (work on project)	
	Project Submission	
TBD	Final project code due	Code zipped, repo link, website link
TBD	Final project video demonstration	Video, PDF of slides

Your Input is Important

To make this course better:

- What web trends would you like to see?
- What technology would you like to see?

Email your suggestions to me

- I'll file them and see if I can include them
- Take a look at the course outline first

As the course progresses email suggestions

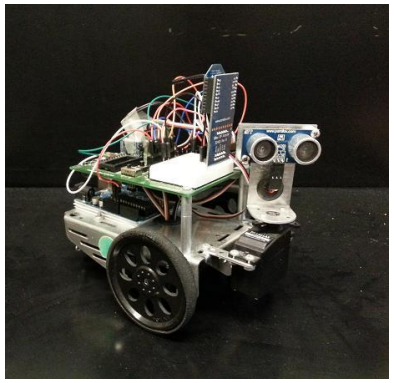
Let me introduce you to…

- MyCourses
  - Assignments – mostly programming
  - Course content
  - Course outline
  - Official announcements
  - TA information
  - Discussion board

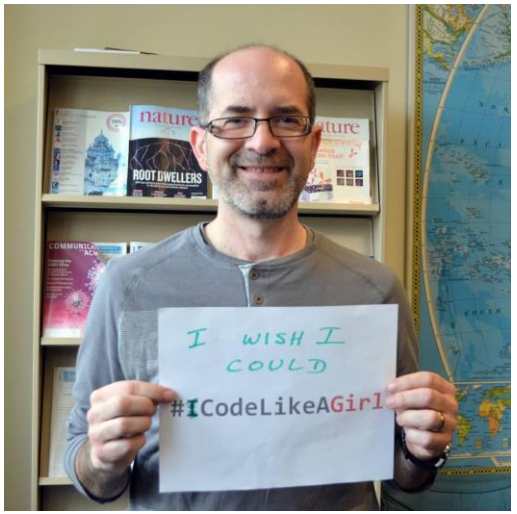
Email Rules

- Is this about an assignment grade?
  - Email the TA, or go to TA's office hours
- Is this about a test grade?
  - Email the prof, or go to the prof's office hours
- Is this a question about the assignment?
  - Post on Ed, or go to someone's office hour
- Is this a question about the lecture material?
  - Post on Ed, or go to someone's office hour
- Is this about a personal matter?
  - Email the prof, or go to the prof's office hours

COMP 307
Principles of Web Development



Joseph Vybihal



Teaching since I graduated in 1990
Worked in industry for 10 years
I'm Canadian, eh.
www.cs.mcgill.ca/~jvybihal

General Artificial Intelligence

Robotics

Impact of Social Media on Society

Contents

COMP 307
Principles of Web Development

Any questions about the course?

Short Lecture Follows

Contents

Trending and History

Short Lecture

COMP 307
Principles of Web Development

Contents

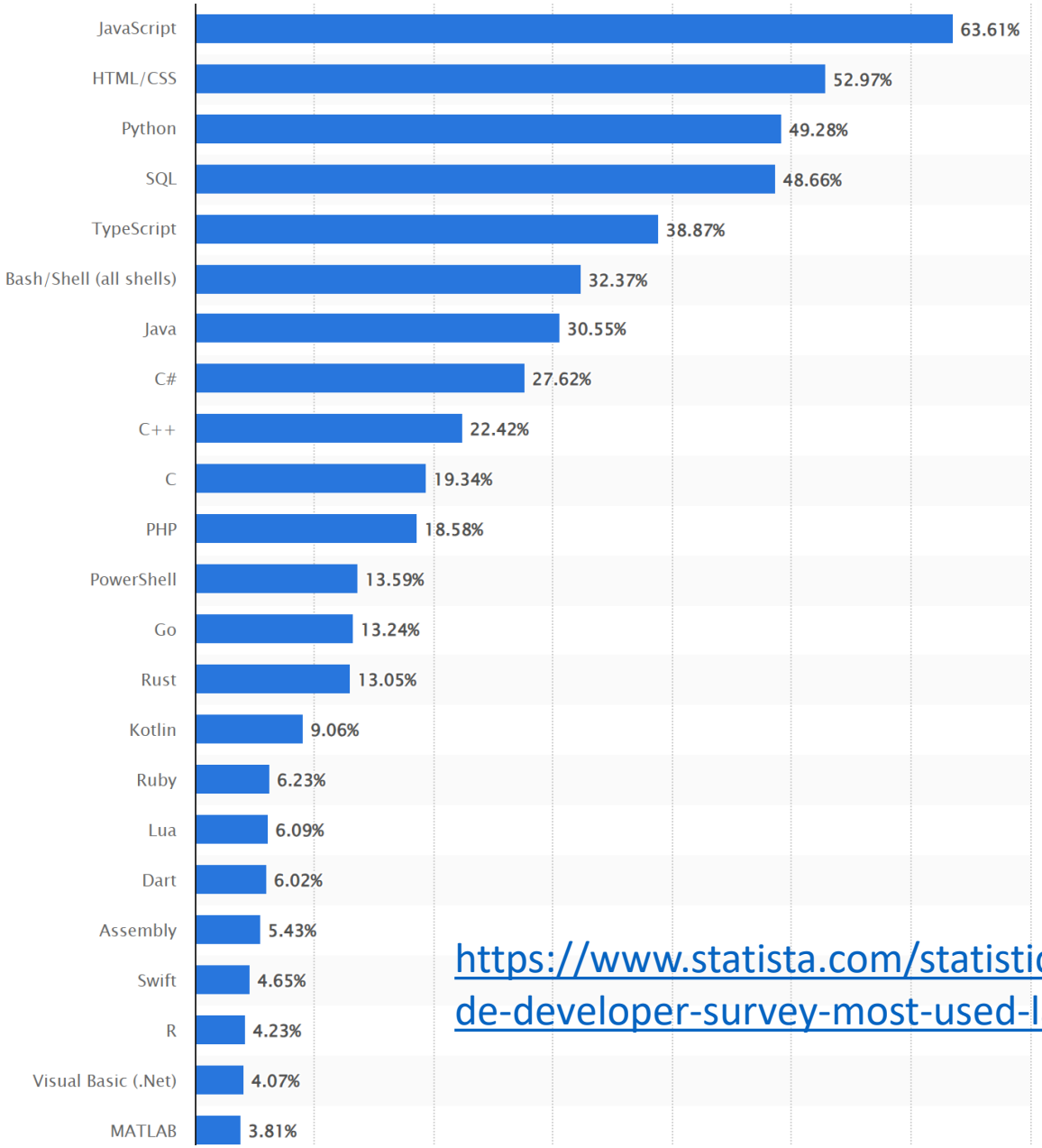
THE INTERNET IN 2023 EVERY MINUTE

Created by: eDiscovery Today & LTMG

COMP 307
Principles of Web Development

Contents

Popular Languages 2023



https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/

Popular Websites 2023

Websites	Popularity (unique visitors per month)[1]	Front-end (Client-side)	Back-end (Server-side)	Database	Notes
Google[2]	2,800,000,000	JavaScript, TypeScript	C, C++, Go,[3] Java, Python, Node	Bigtable,[4] MariaDB[5]	The most used search engine in the world.
Facebook	1,120,000,000	JavaScript, Typescript, Flow	Hack/HHVM, Python, C++, Java, Erlang, D,[6] Haskell[7]	MariaDB, MySQL,[8] HBase, Cassandra[9]	The most visited social networking site.
YouTube	1,100,000,000	JavaScript, TypeScript	Python, C, C++, Java,[10] Go[11]	Vitess, BigTable, MariaDB[5][12]	The most popular video sharing site.
Yahoo	750,000,000	JavaScript	PHP	PostgreSQL, HBase, Cassandra, MongoDB,[13]	
Etsy	516,000,000 (Total, not unique)[14]	JavaScript	PHP[15][16]	MySQL, Redis[17]	E-commerce website.
Amazon	2,400,000,000[18]	JavaScript	Java, C++, Perl[19]	DynamoDB, RDS/Aurora, Redshift[20]	The most used e-commerce site in the world.
Wikipedia	475,000,000	JavaScript	PHP	MariaDB[21]	A free online encyclopedia based on MediaWiki, which is programmed in PHP.
Fandom	315,000,000[22]	JavaScript	PHP	MySQL	Wiki hosting service.
X	290,000,000	JavaScript	C++, Java,[23] Scala,[24] Ruby (Ruby On Rails)	MySQL[25]	Popular social network.
Bing	285,000,000	JavaScript	C++, C#	Microsoft SQL Server, Cosmos DB	Search engine from Microsoft.
eBay	285,000,000	JavaScript	Java,[26] JavaScript,[27] Scala[28]	Oracle Database	Online auction house.
MSN	280,000,000	JavaScript	C# (ASP.NET)	Microsoft SQL Server	An email client, for simple use. Previously known as "messenger", not to be confused with Facebook's messaging platform.
LinkedIn	260,000,000	JavaScript	Java, JavaScript,[29] Scala	Venice[30][31]	World's largest professional network.
Pinterest	250,000,000	JavaScript	Python (Django),[32] Erlang, Elixir[33]	MySQL, Redis[34]	Search engine for ideas.
WordPress.com	240,000,000 [35]	JavaScript	PHP[36]	MariaDB[37]	Website manager software.
Netflix	223.090.000 (Subscribers, not visitors)	JavaScript	Python, Java[38]	NMDB,[39] PostgreSQL	The biggest video streaming service in the world.

https://en.wikipedia.org/wiki/Programming_languages_used_in_most_popular_websites

Backend Server-side 2023

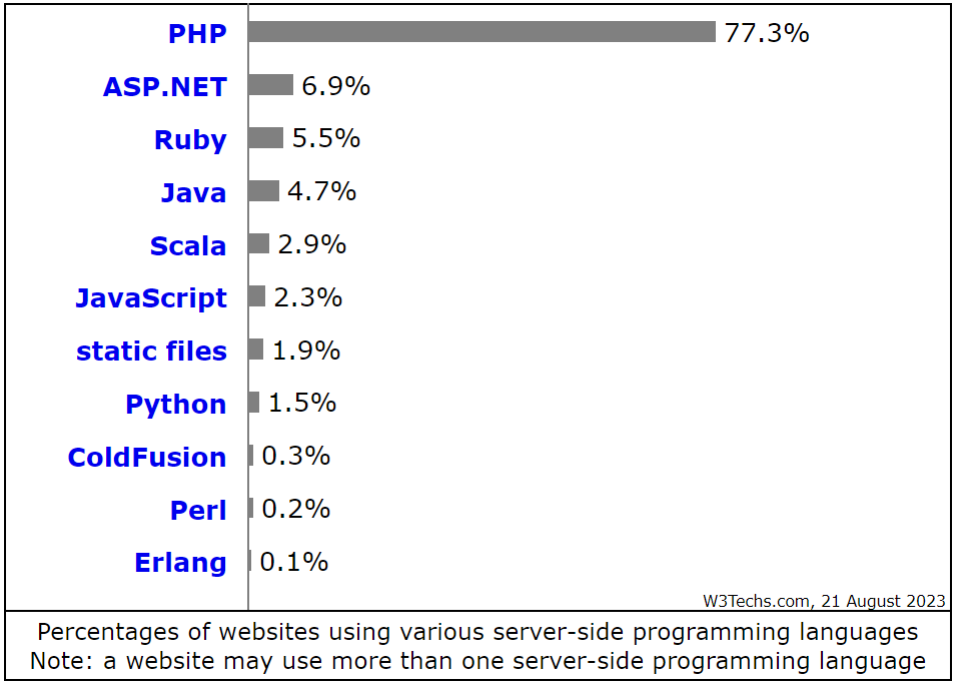
Websites	C#	C	C++	D	Elixir	Erlang	Go	Hack	Haskell	Java	JavaScript	Perl	PHP	Python	Ruby	Scala
Google	No	Yes	Yes	No	No	No	Yes	No	No	Yes	Yes	No	No	Yes	No	No
YouTube	No	Yes	Yes	No	No	No	Yes	No	No	Yes	No	No	No	Yes	No	No
Facebook	No	No	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes	No	No
Yahoo	No	Yes	Yes	No	No	No	Yes	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Amazon	No	No	Yes	No	No	No	No	No	No	Yes	No	Yes	No	No	No	No
Wikipedia	No	No	No	No	No	No	No	No	No	No	No	No	Yes	No	No	No
X	No	No	Yes	No	No	No	No	No	No	Yes	No	No	No	No	Yes	Yes
Bing	Yes	No	Yes	No	No	No	No	No	No	No	No	No	No	No	No	No
eBay	No	No	No	No	No	No	No	No	No	Yes	Yes	No	No	No	No	Yes
MSN	Yes	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
LinkedIn	No	No	No	No	No	No	No	No	No	Yes	Yes	No	No	No	No	Yes
Pinterest	No	No	No	No	Yes	Yes	No	No	No	No	No	No	No	Yes	No	No
WordPress.com	No	No	No	No	No	No	No	No	No	No	No	No	Yes	No	No	No
Netflix	No	Yes	No	No	No	No	Yes	No	No	Yes	No	No	No	Yes	No	No

Contents

What are the languages 2023?

https://w3techs.com/technologies/overview/programming_language

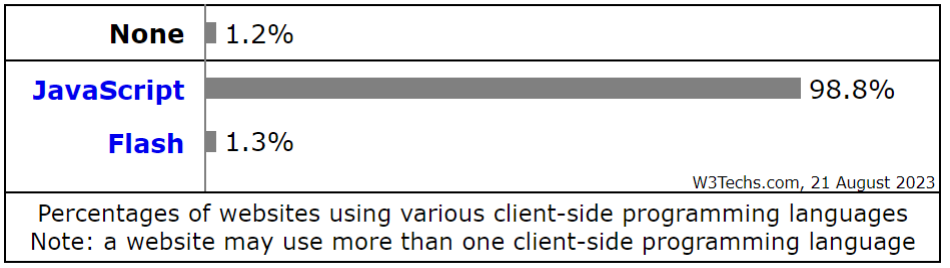
Backend



Percentages of websites using various server-side programming languages
Note: a website may use more than one server-side programming language

Frontend

Of course HTML and CSS
(but we don't consider this programming.)



Percentages of websites using various client-side programming languages
Note: a website may use more than one client-side programming language

https://w3techs.com/technologies/overview/client_side_language

Some History

Beginnings of the Internet

Research Papers:

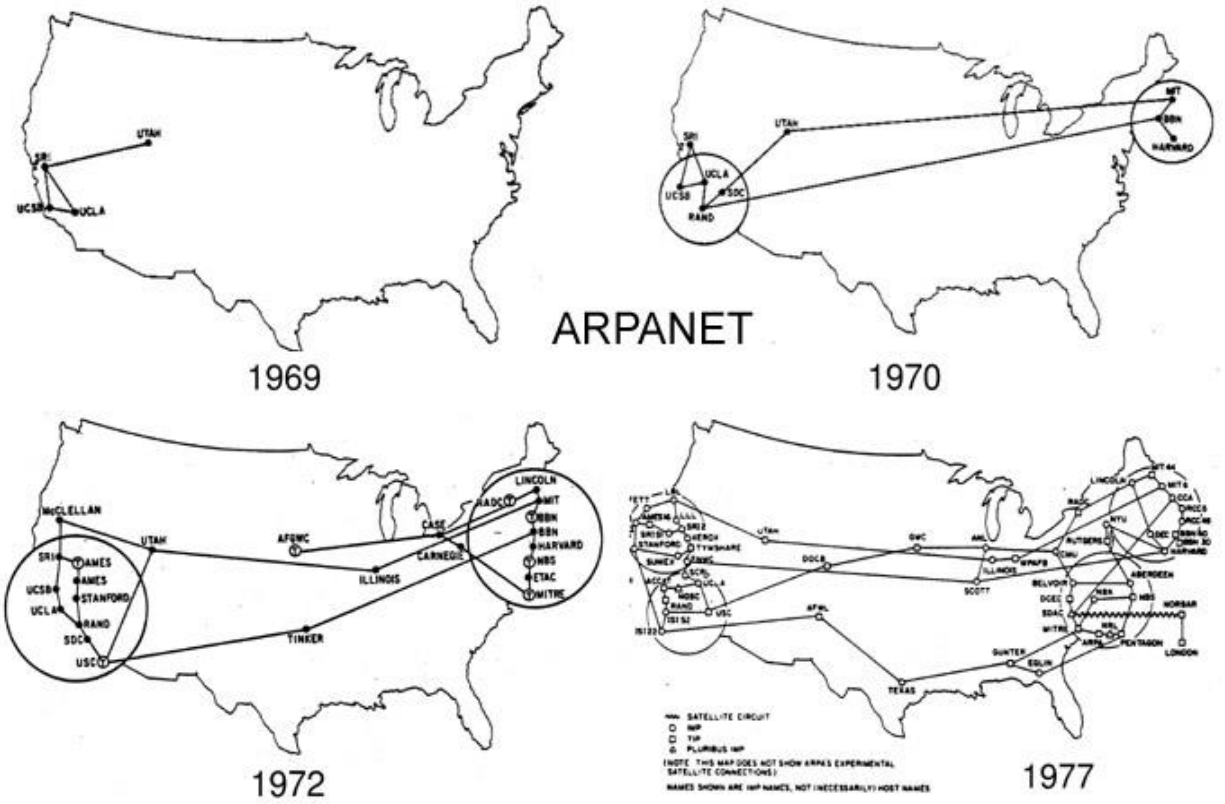
- July 1961 – Leonard Kleinrock – MIT
  - Theory of packet-based networks
- August 1962 – J.C.R. Licklider – MIT
  - Theorized a “Galactic Network”
- During 1965 – Kleinrock, Roberts, Merrill
  - Low speed dial-up connection between MIT and California
  - It was the first wide-area network
- In 1966 – Roberts went to DARPA
  (USA: Defence Advanced Research Projects Agency)

  The military got interested…

COMP 307
Principles of Web Development

ARPANET

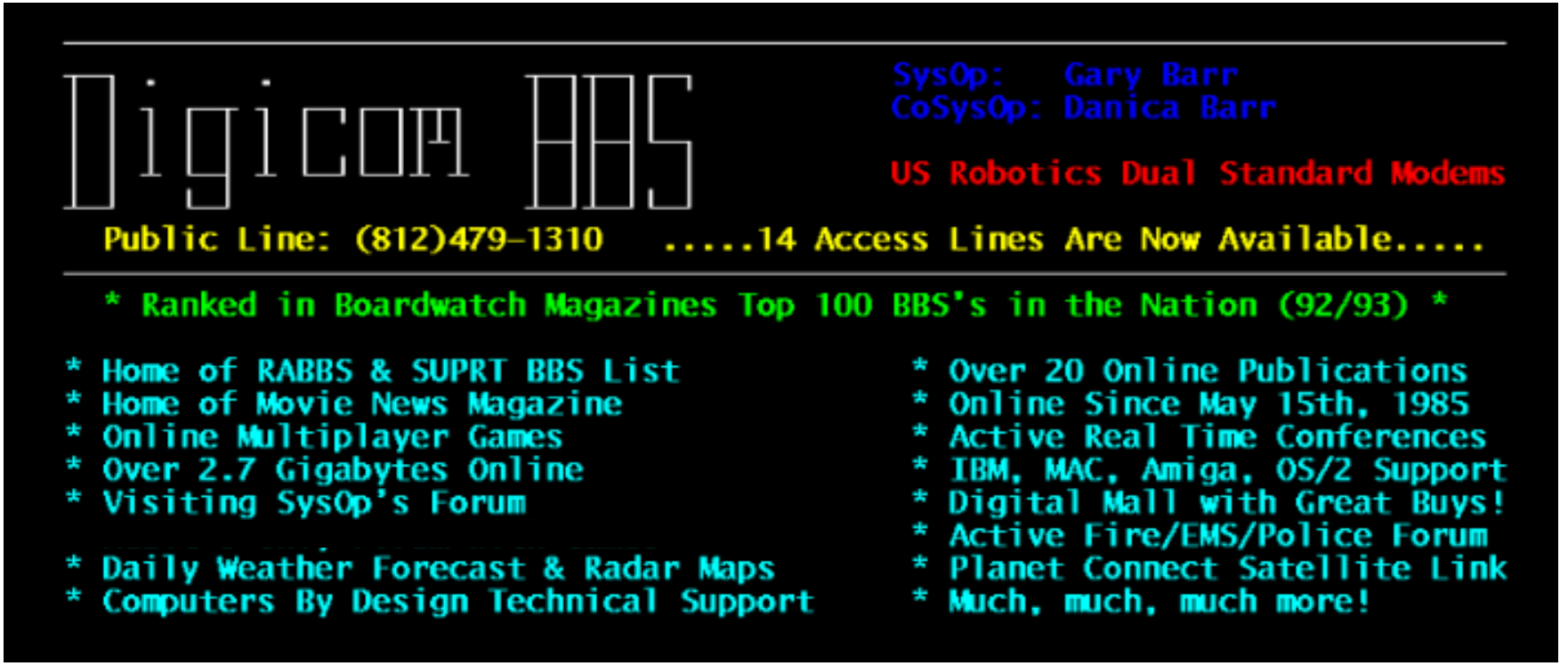
Based on a concept first published in 1967 (Roberts, Kleinrock) at DARPA (US Defence Advanced Research Project Agency), there ARPANET was developed under the direction of the U.S. Advanced Research Projects Agency (ARPA).



The Builders of ARPANET

IMP Team (Left to Right): Truett Thatch, Bill Bartell (Honeywell), Dave Walden,
Jim Geisman, Robert Kahn, Frank Heart, Ben Barker, Marty Thrope, Will Crowther, Severo Ornstein.

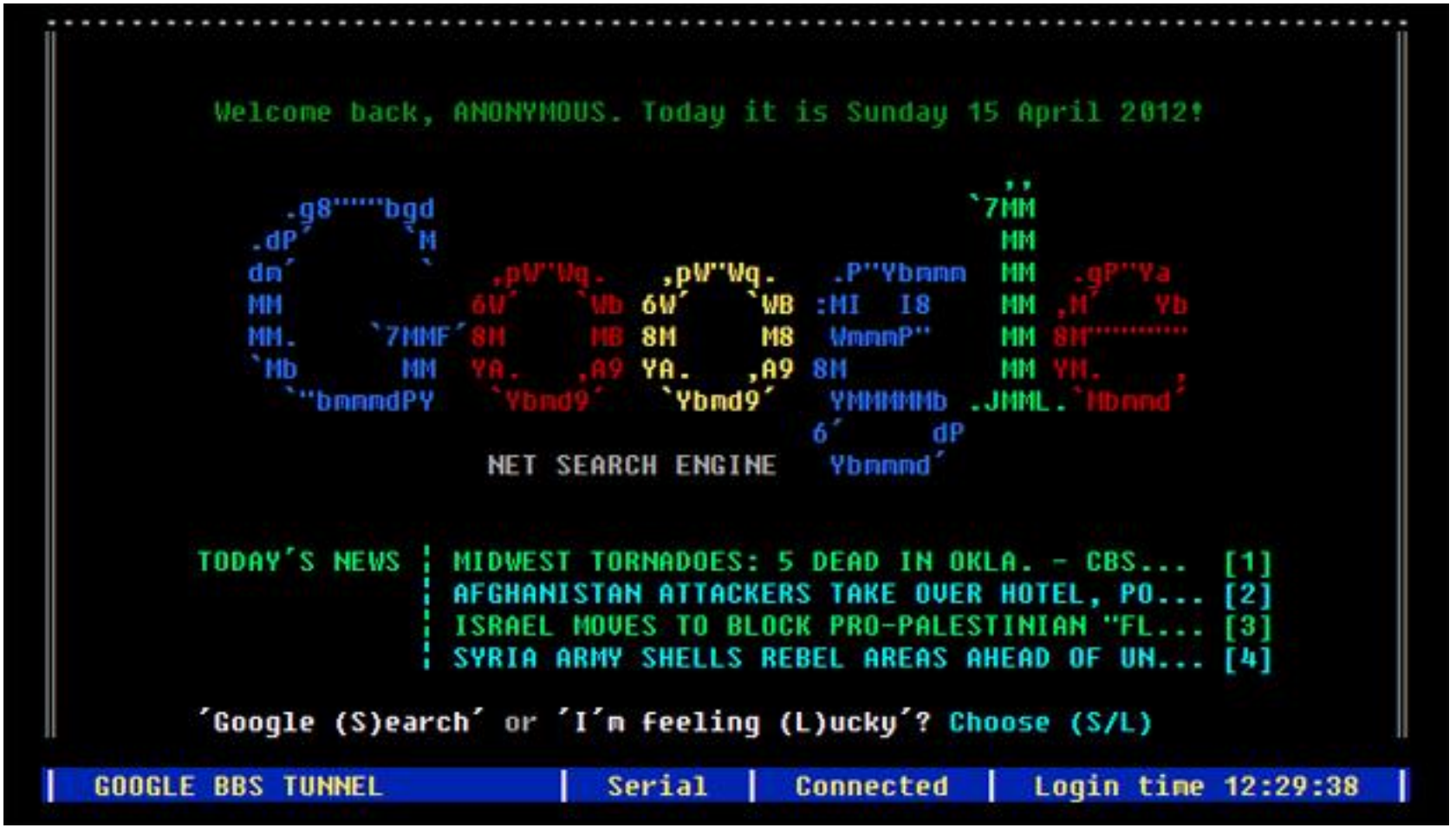
The BBS (1980s)



Bulletin Board Services (first popular online social environments)

Remote connect with your modem to their server.
Just a window into their server.
Nothing executes on your end.

Search Engines (1990s)



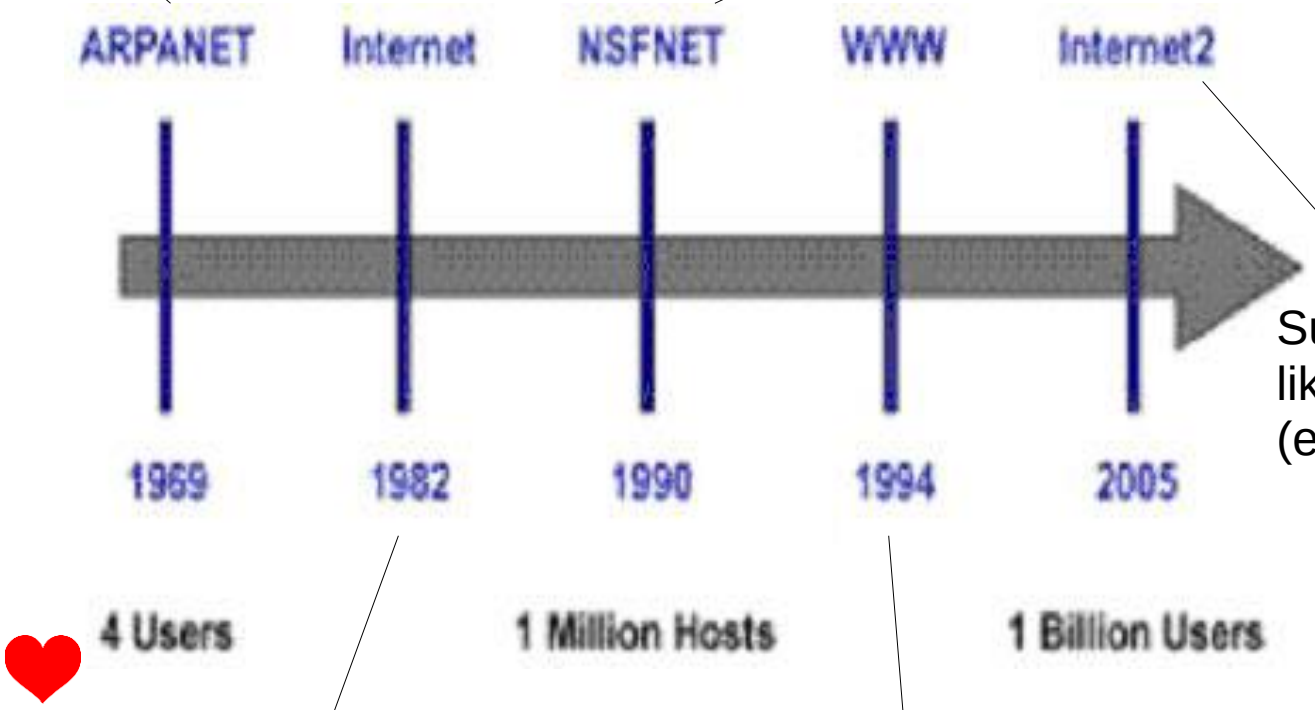
No mouse interface.
Character data input at a prompt.

- Google in text mode

Evolution

USA Advanced Research Project Agency Network

Nation Science Foundation Network (global schools)

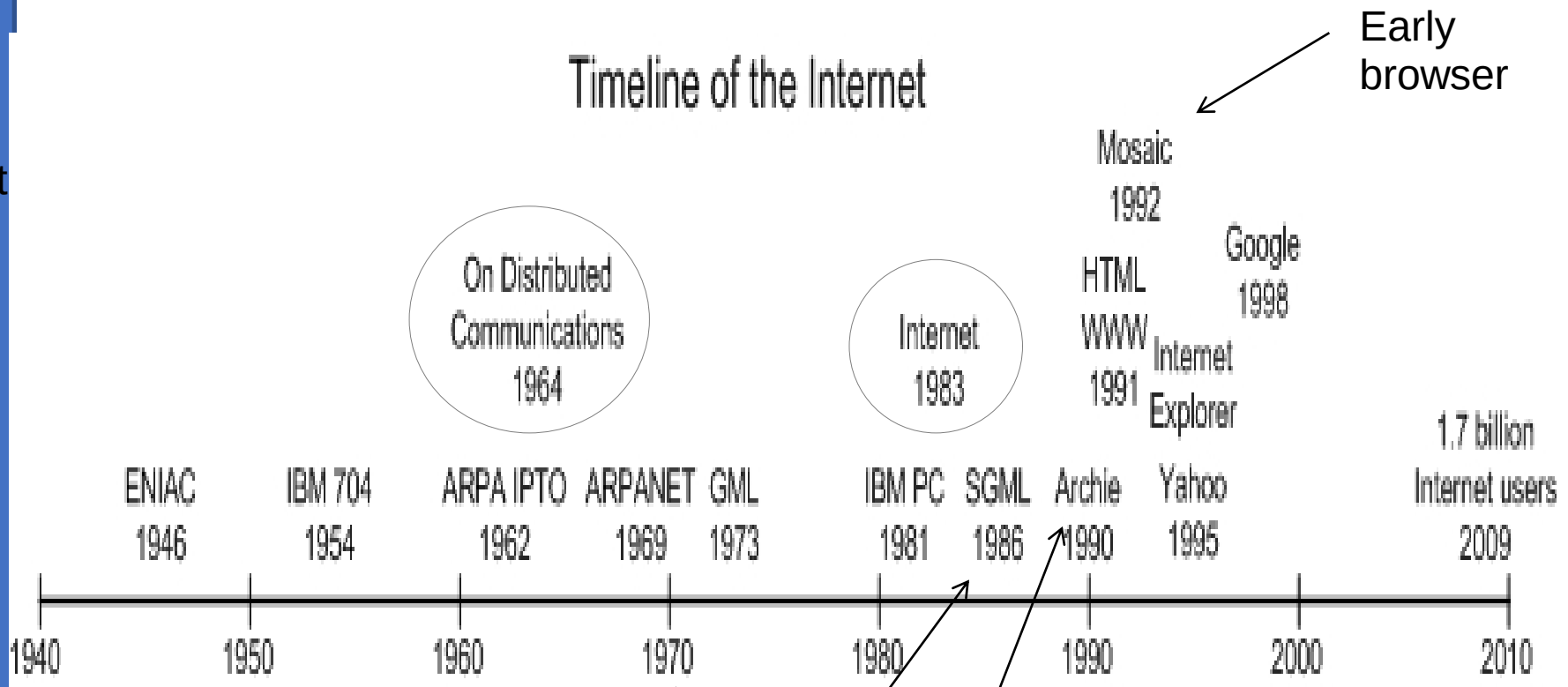


Sub-networks like Facebook (ecosystems)

"Globalized" (USA & Canada)

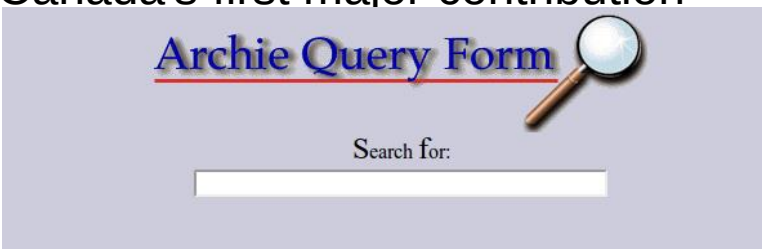
Browser based

Another View



Cerf & Kahn
Paper on TCP

First true Internet Search Engine
(McGill University)
https://en.wikipedia.org/wiki/Archie_search_engine
Canada's first major contribution



General Markup
Language defined

Evolution of the Internet

Web1 – Web 1.0

1991-2004 Consumer static webpages

Web2 – Web 2.0

2004-today Web as a platform to publish your own content using social media platforms

Web3 – Web 3.0

Future – Decentralization & blockchains (maybe Semantic Web)

coined by Ethereum co-founder Gavin Wood in 2014

About Web Development

Short Lecture

The Development Stack

What is the stack?

- A website is not built from one programming language, but a family of languages
- Like dressing ourselves in the morning, we can mix and match
- Web development languages are divided into these groupings:
  - Front-end languages
  - Data transportation languages
  - Back-end languages

Are you a full stack developer?

This means, can you program the front-end, the back-end, and the data transport areas of a website?

If the answer is no, then you are not a full stack developer, you are only a:

- Front-end developer, or
- Back-end developer, or
- Data transport specialist (not really a dev)

Common Dev Stacks

Stack Level	XAMPP	MEAN	DJANGO	MERN
Font-end (HTML + CSS)	JavaScript	Angular	Java Script	React
Data Transport	HTTP CGI XML/JSON AJAX/jQuery	HTTP JSON Ang.lib	HTTP CGI XML/JSON AJAX/jQuery	HTTP JSON React.lib
Backend	Server Apache Languages PHP C/C++ Python Perl Bash Databases SQL DB Any really	Server Node.js w/ Express Languages Java Databases Mongo DB Any really	Server Apache Languages Python w/ Django lib Databases SQL DB Any really	Server Node.js w/ Express Languages Java Databases Mongo DB Any really

Internet Technology

Is a set of applications that work together as a software system to connect the front-end and back-end architectures.

- Front-end browser: IE, Chrome, Safari, etc.
- Back-end machine: Windows, Unix, Mac, etc.
- Back-end sever: Apache, Node.js, etc.

Even though each of these things work differently, we need **standards** to unify them.

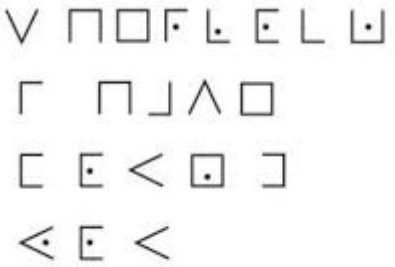
Why is the Internet an unsupervised network?

War

World War 2 (1930/40)

Motivated Computers

"We need to crack cyphers!"

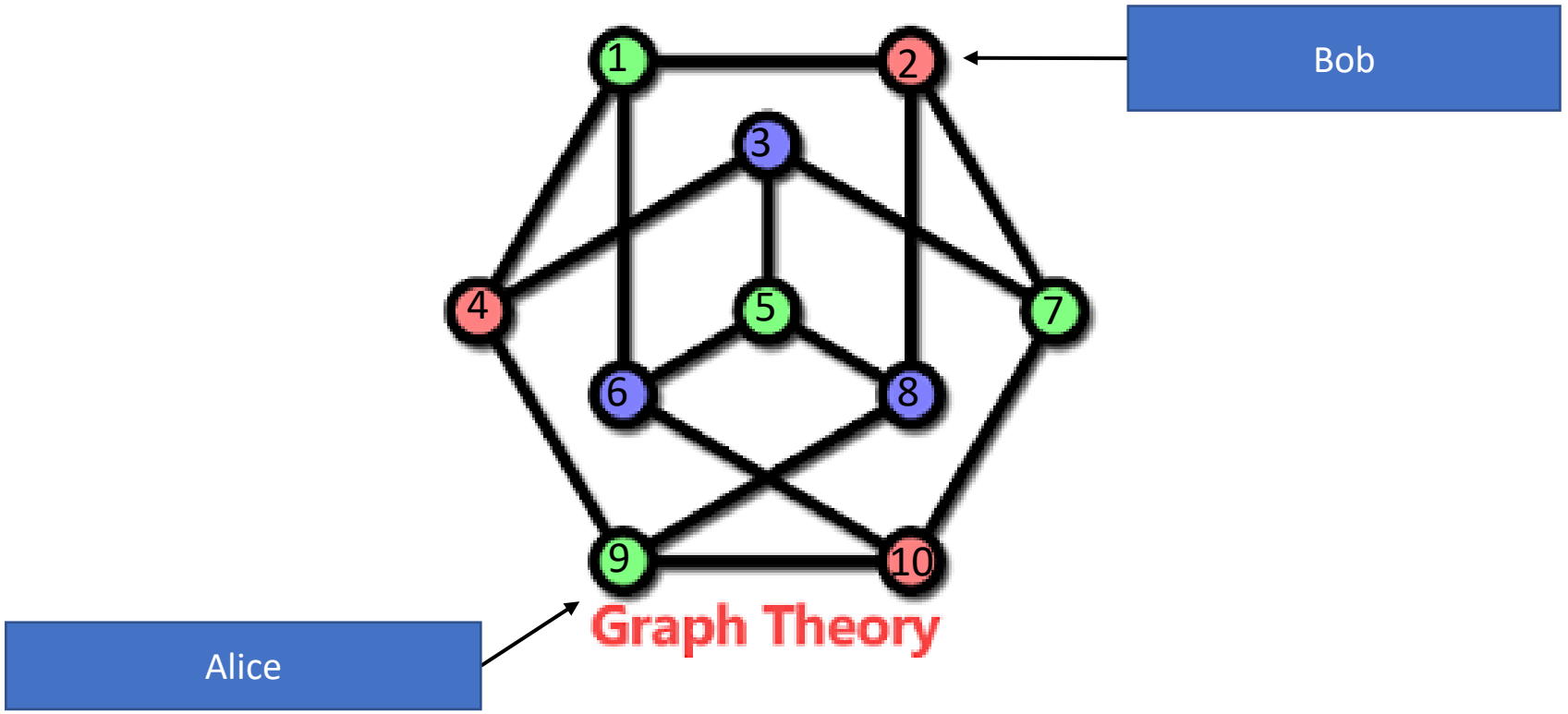


The Cold War (1960/70)

Motivated Internet

"What if a bomb blows up half the country!"

Peer to Peer Network

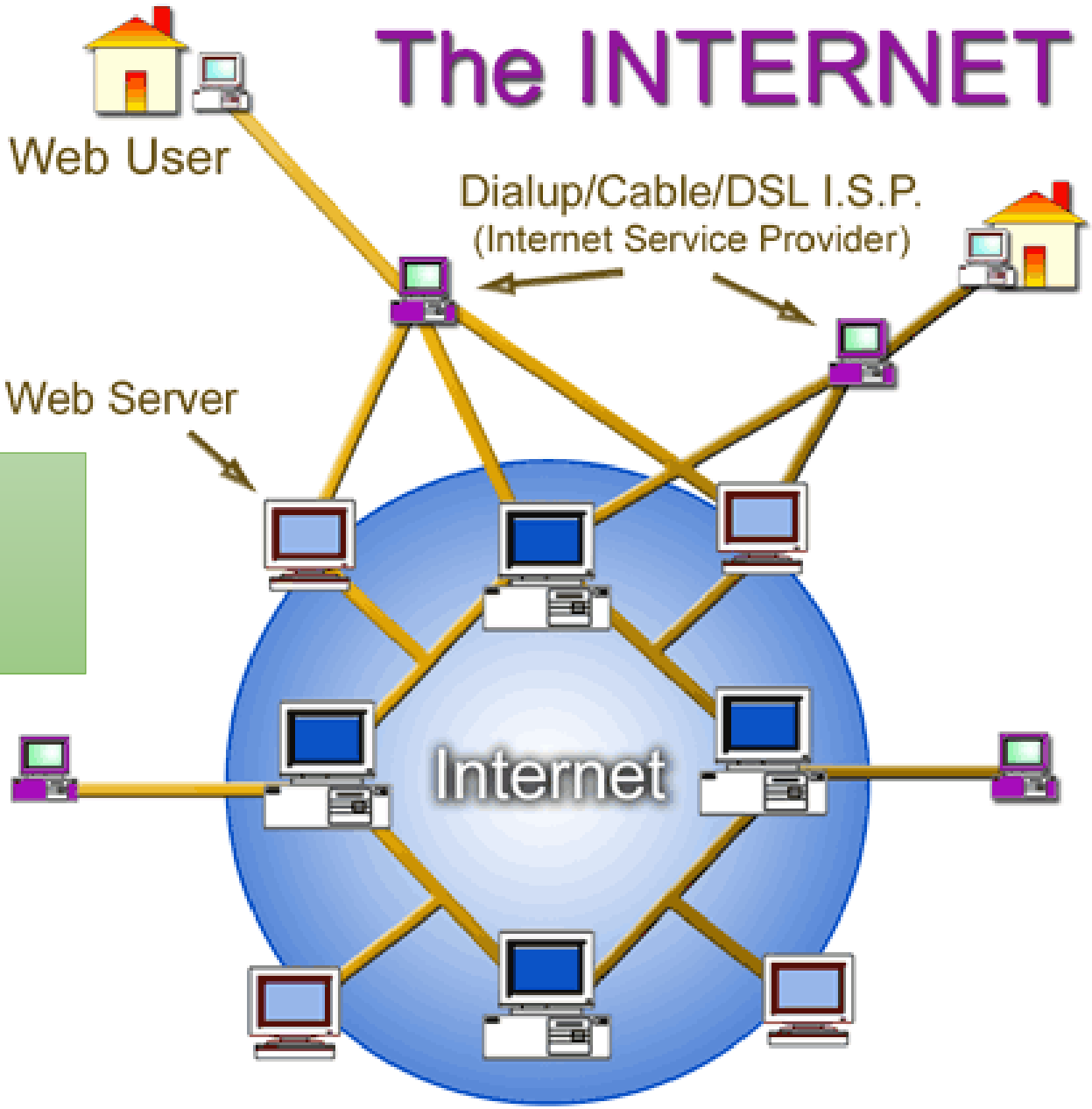


How many paths are there from Alice to Bob?

If nodes 3,5,6,8 stop working, can Alice still talk to Bob?

Peer-to-Peer automatically finds a path to the destination using a "**broadcasting** race" or a "mapping technique" called **routing** (optimality not guaranteed).

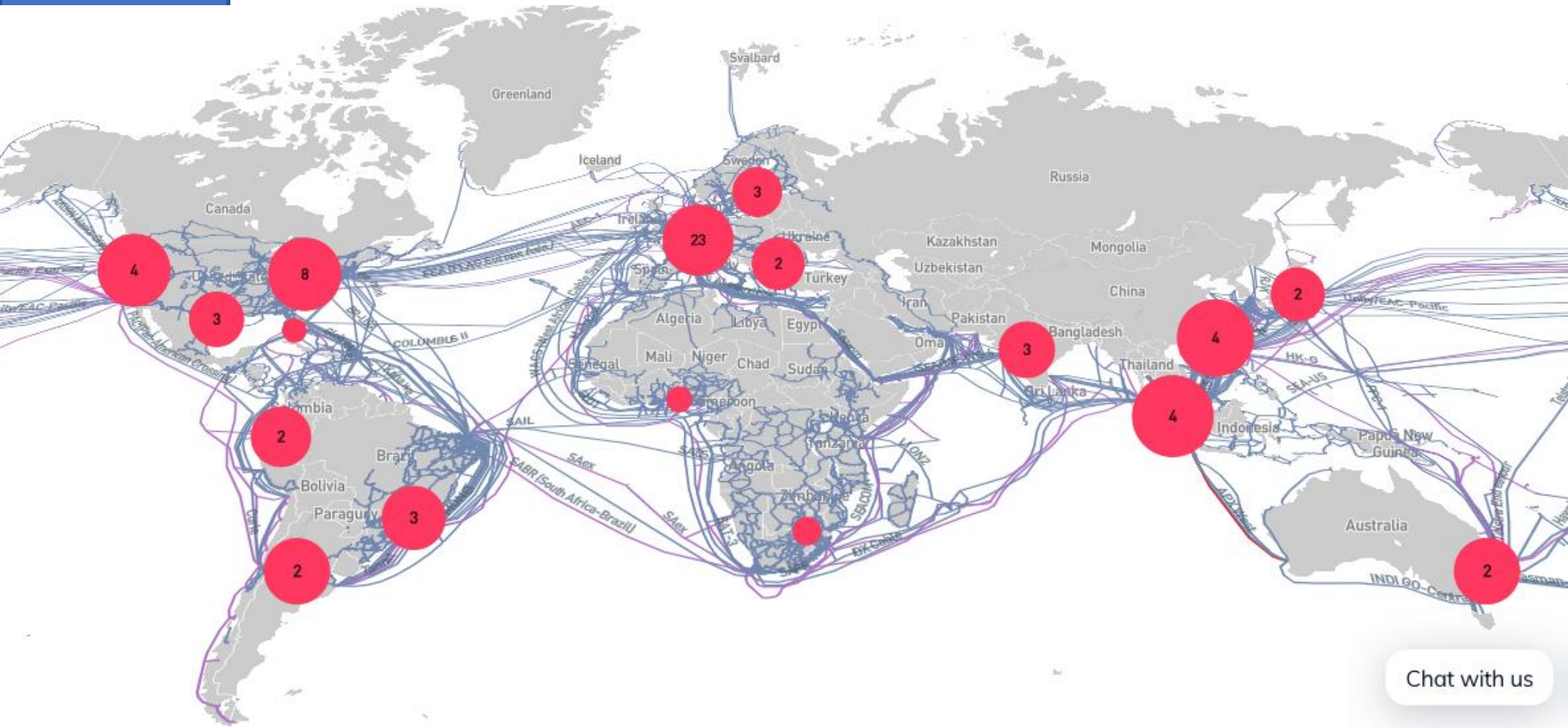
A Peer-to-Peer Network



- Users
- Providers
- Edge Nodes
- Backbone

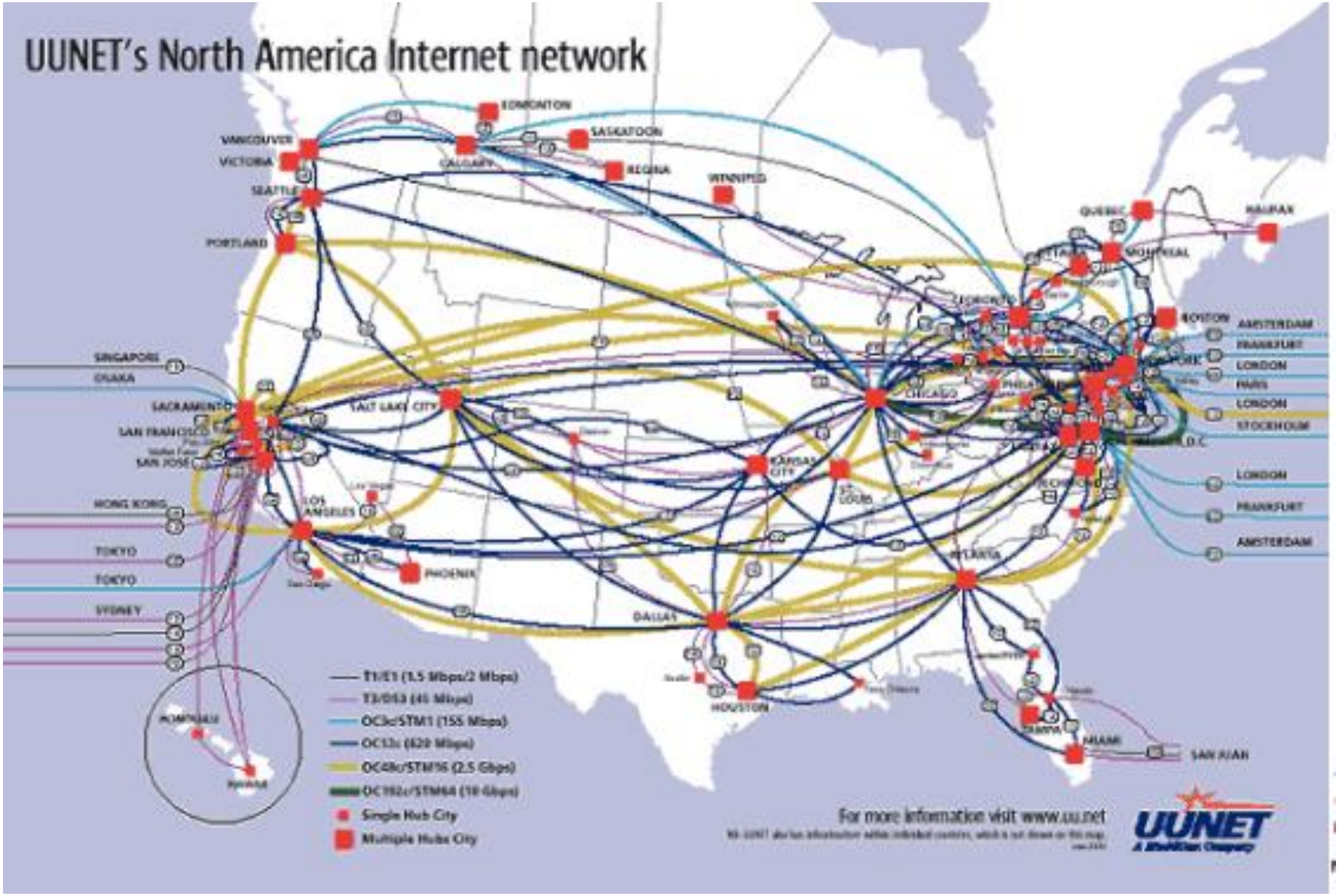
https://www.thinglink.com/scene/679834952350564352

Global Infrastructure



COMP 307
Principles of Web Development

Contents



Four Internet Ground Rules

- Each distinct node would have to stand on its own
- Communication would be on a best effort basis
  - If a message did not get through it would be re-transmitted eventually
- Nodes would be “black boxes”
  - Each node could be implemented as you want, but it must supply specific functionalities
- No global control at the operational level
  - “semi-auto” manages itself

About Design

Website Designs

- Static Multi-paged website https://earthsky.org/
  - Responsive Design
  - Interactive Design
- Dynamic multi-paged website http://www.cs.mcgill.ca/~jvybihal
  - Dynamic content
  - Dynamic page generation
- Single page website https://www.upstatelaundromat.com/
  - Cell phones
  - React
- Database-based website https://www.google.com/
  - 3-tired applications

The IT Crowd

Jen: [Moss has a small plastic box with a flashing light] What is it?
Moss: This, Jen, is the Internet.
Jen: What?
Moss: That's right.
Jen: This is the Internet? The whole Internet?
Moss: Yep. I asked for a loan of it so that you could use it in your speech.
Jen: It's so small.
Moss: That's one of the surprising things about it.
Jen: Hang on, it doesn't have any wires or anything.
Moss: It's wireless.
Jen: Oh, yes, everything's wireless nowadays, isn't it... yeah. So, I can really use it in my speech? What if someone needs it?
Moss: Oh, no, no, people will still be able to go online and everything. It will still work.
Jen: Oh, good, good...
Moss: I tell you, you present this to the shareholders and you will get quite the response.
Jen: Can I touch it? It's so light!
Moss: Of course it is, Jen. The Internet doesn't weigh anything.
Jen: No, of course it doesn't.
[laughs nervously]
Roy: Hey! What is Jen doing with the Internet?
Jen: Moss said I could use it for my speech.
Roy: Are you insane? What if she drops it?
Jen: I won't drop it, I'll look after it.
Roy: No. No, no, no, no, Jen. No, this needs to go straight back to Big Ben.
Jen: Big Ben?
Moss: Yep. It goes on top of Big Ben. That's where you get the best reception.

Introduce yourself!

- Make friends because of the project
- Introduce yourself to your neighbour:
  - Name one of your core values
  - How does CS relate to that core value?