IN4MATX 133: User Interface Software

Lecture 1:
Introduction & History, Continued
Basics of Web Communication

Announcements

- Undergraduate Research Lab

https://forms.gle/ohs7yA2EvRsCxsyj8

Today’s goals

By the end of today, you should be able to…

- CONTINUE….Describe how society got to today’s ubiquitous computing
- Hypothesize why web technology has become the de-facto tool for interface development
- Describe the fundamentals of web communication
- Identify the syntax of HTML tags and attributes and describe their roles
- Create a HTML template which follows W3C specifications

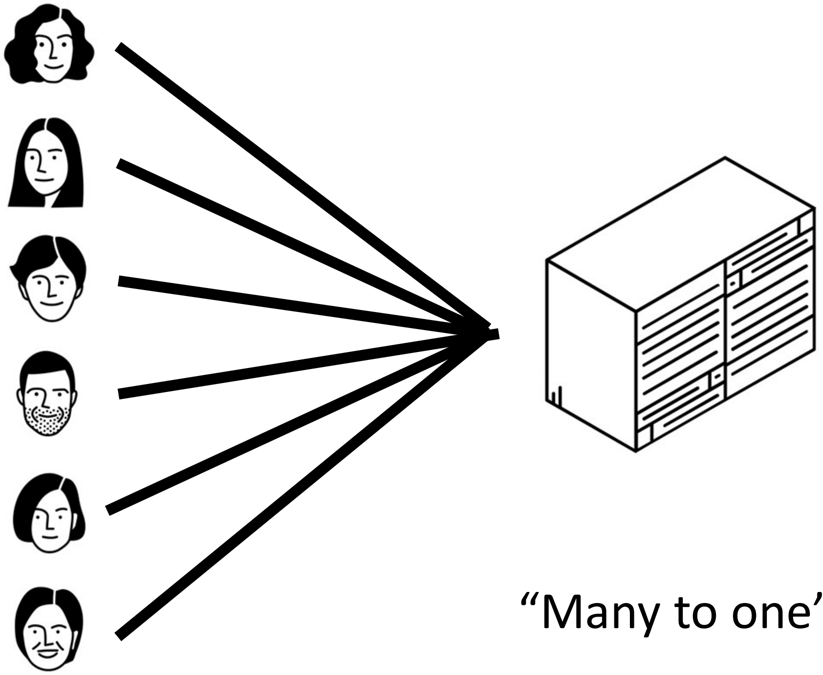
Three waves of computing

1 Mainframe computing

2 Personal computing

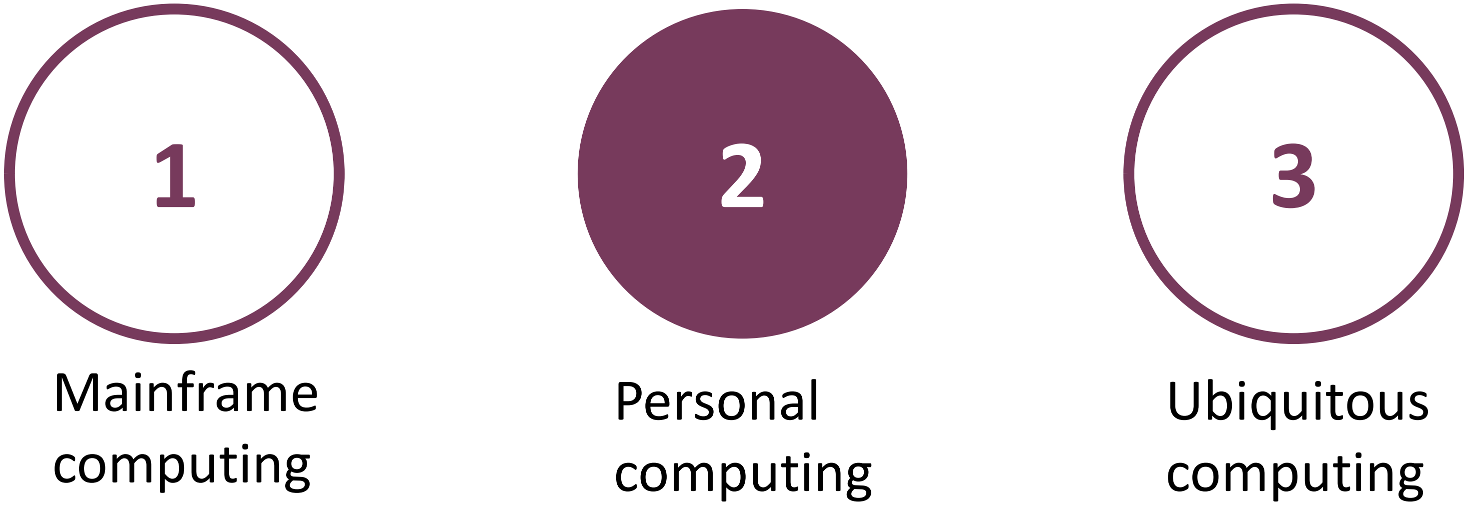
3 Ubiquitous computing

First wave: mainframe computing



“Many to one”

Three waves of computing



1

Mainframe computing

2

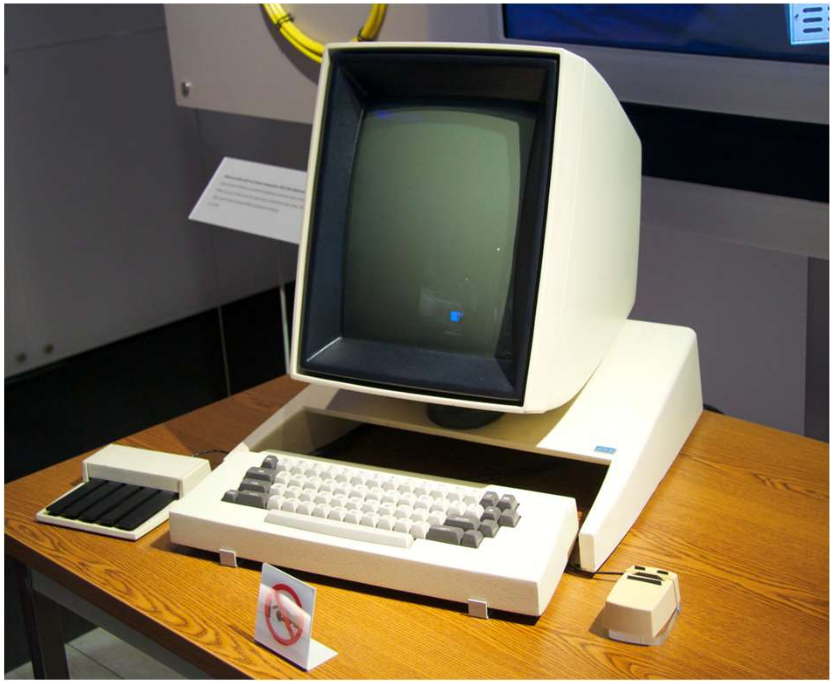
Personal computing

3

Ubiquitous computing

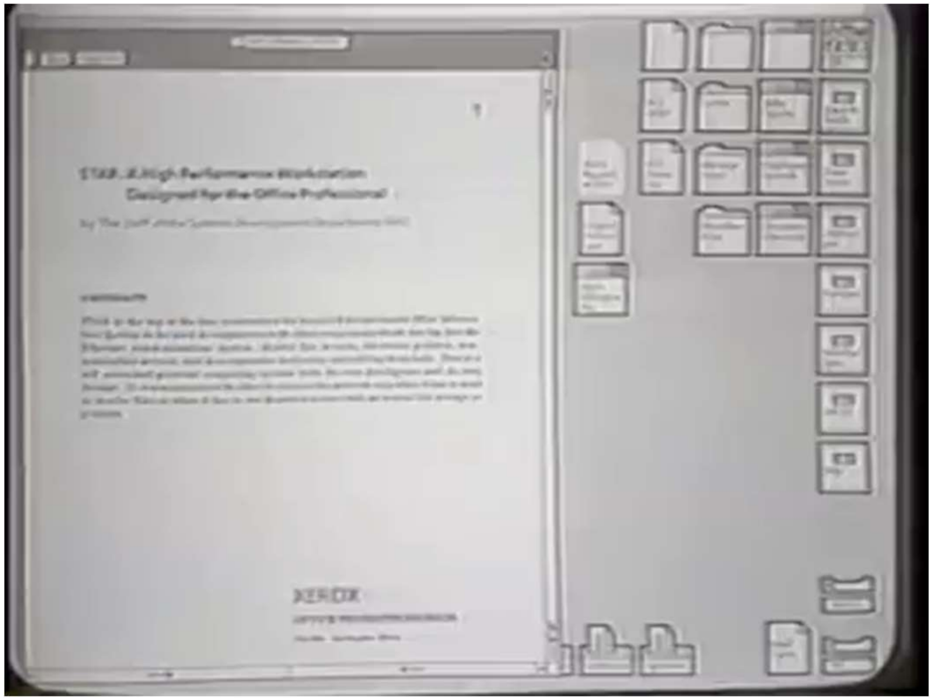
Second wave: personal computing

- First introduced by Xerox
- Xerox Alto, 1973
  - Mouse
  - Chording keyboard
- Xerox Star, 1981
- Xerox models
  were commercially unsuccessful
  - Still expensive, too few applications



Second wave: personal computing

Xerox Star (1981)



STAR: A High Performance Workstation
Designed for the Office Professional

[illegible]

XEROX

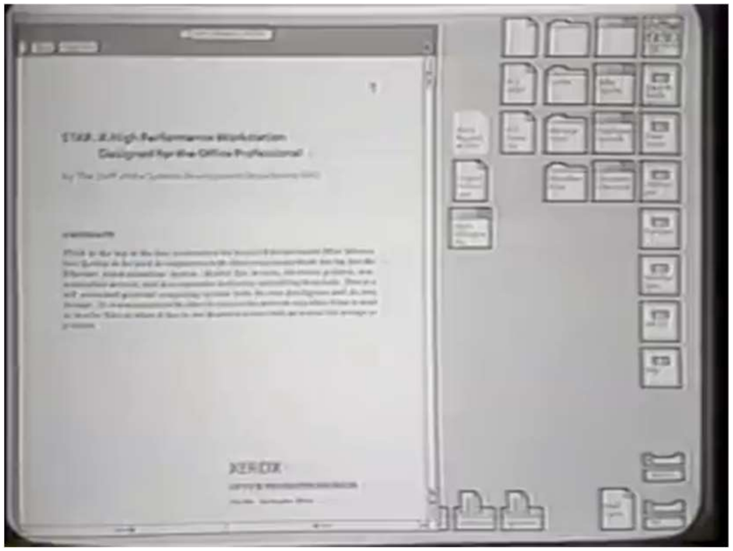
https://www.youtube.com/watch?v=ODZBL80JPqw

Did you recognize any interactions that are commonly used today?

Second wave: personal computing

Xerox Star (1981)

- Software running in windows
- Desktop with icons for navigating between files and programs
- Super slow!



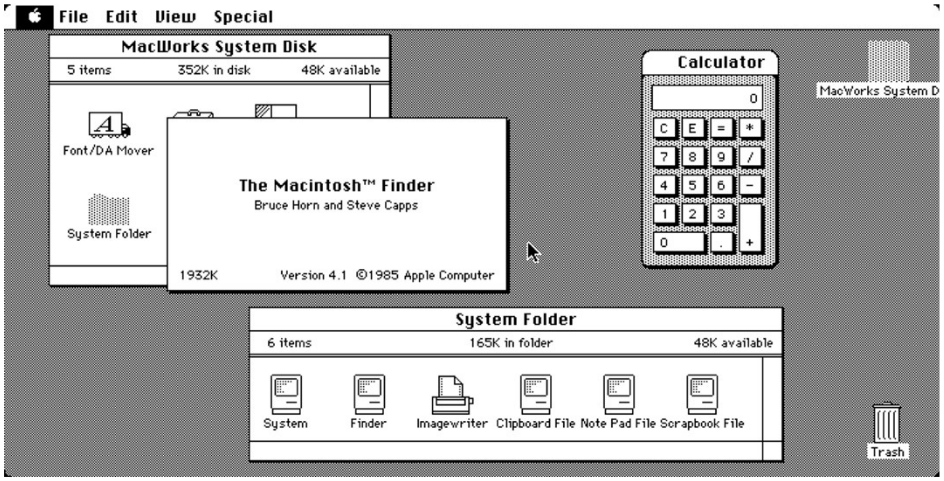
STAR, A High Performance Workstation
Designed for the Office Professional

[illegible]

XEROX

Second wave: personal computing

Macintosh (1984)



5 items 352K in disk 48K available

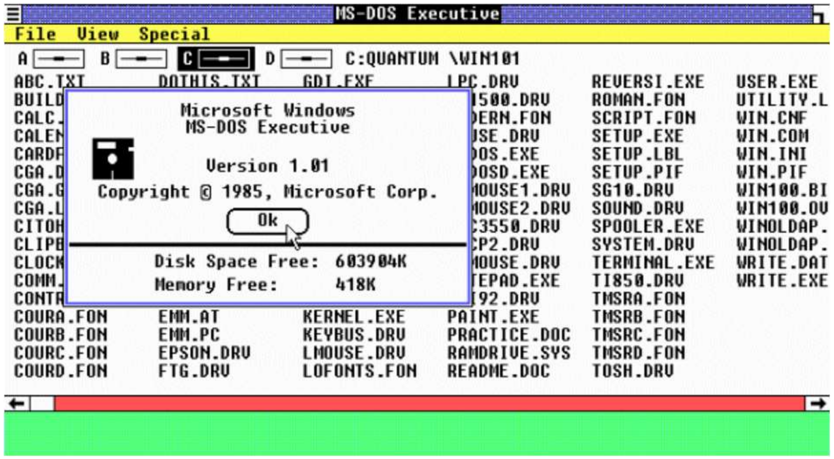
The Macintosh™ Finder
Bruce Horn and Steve Capps

1932K Version 4.1 ©1985 Apple Computer

6 items 165K in folder 48K available

Second wave: personal computing

Windows 1.0 (1985)

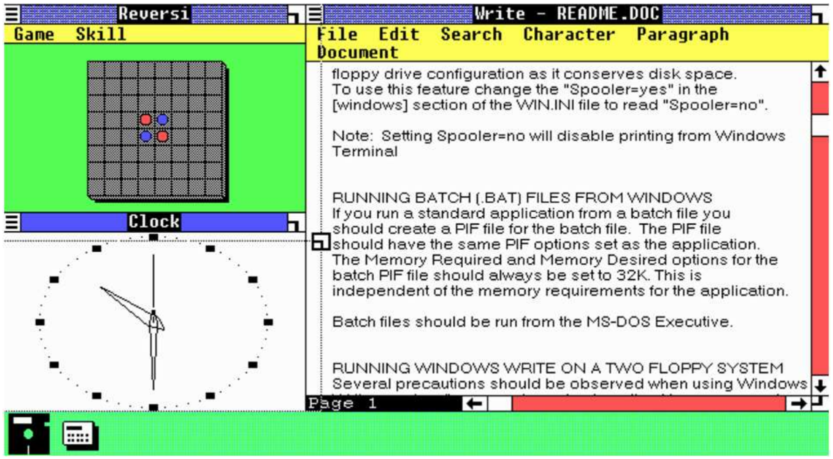


```
C:QUANTUM \WIN101
ABC.TXT          DOTHIS.TXT       GDI.EXE          LPC.DRV          REVERSI.EXE      USER.EXE
BUILD                                              1500.DRV         ROMAN.FON        UTILITY.L
CALC.                                              DERN.FON         SCRIPT.FON       WIN.CNF
CALEN                                              USE.DRV          SETUP.EXE        WIN.COM
CARDF                                              DOS.EXE          SETUP.LBL        WIN.INI
CGA.D                                              DOSD.EXE         SETUP.PIF        WIN.PIF
CGA.G                                              MOUSE1.DRV       SG10.DRV         WIN100.BI
CGA.L                                              MOUSE2.DRV       SOUND.DRV        WIN100.OV
CITOH                                              C3550.DRV        SPOOLER.EXE      WINOLDAP.
CLIPB                                              CP2.DRV          SYSTEM.DRV       WINOLDAP.
CLOCK                                              MOUSE.DRV        TERMINAL.EXE     WRITE.DAT
COMM.                                              TEPAD.EXE        TI850.DRV        WRITE.EXE
CONTR                                              I92.DRV          TMSRA.FON
COURA.FON        EMM.AT           KERNEL.EXE       PAINT.EXE        TMSRB.FON
COURB.FON        EMM.PC           KEYBUS.DRV       PRACTICE.DOC     TMSRC.FON
COURC.FON        EPSON.DRV        LMOUSE.DRV       RAMDRIVE.SYS     TMSRD.FON
COURD.FON        FTG.DRV          LOFONTS.FON      README.DOC       TOSH.DRV
```

```
Microsoft Windows
MS-DOS Executive

Version 1.01
Copyright © 1985, Microsoft Corp.

Disk Space Free:   603904K
Memory Free:       418K
```



floppy drive configuration as it conserves disk space.
To use this feature change the "Spooler=yes" in the
[windows] section of the WIN.INI file to read "Spooler=no".

Note: Setting Spooler=no will disable printing from Windows
Terminal

RUNNING BATCH (.BAT) FILES FROM WINDOWS
If you run a standard application from a batch file you
should create a PIF file for the batch file. The PIF file
should have the same PIF options set as the application.
The Memory Required and Memory Desired options for the
batch PIF file should always be set to 32K. This is
independent of the memory requirements for the application.

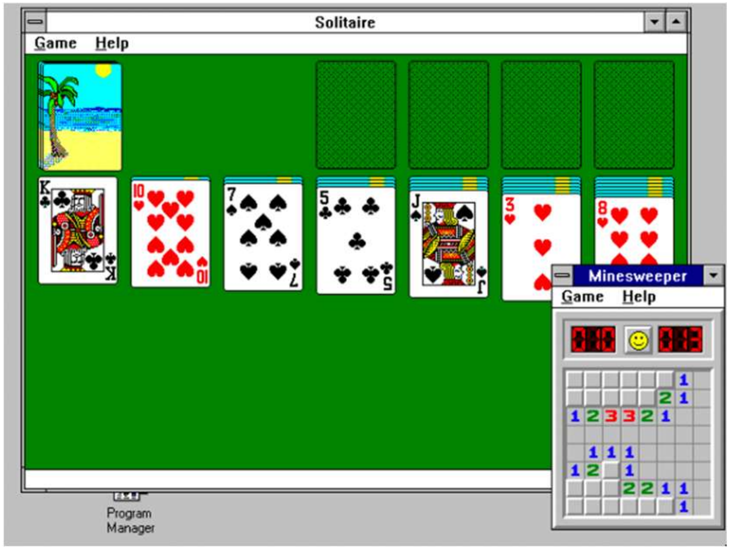
Batch files should be run from the MS-DOS Executive.

RUNNING WINDOWS WRITE ON A TWO FLOPPY SYSTEM
Several precautions should be observed when using Windows

Second wave: personal computing

Windows 3.0 & 3.1 (1990 & 1992)

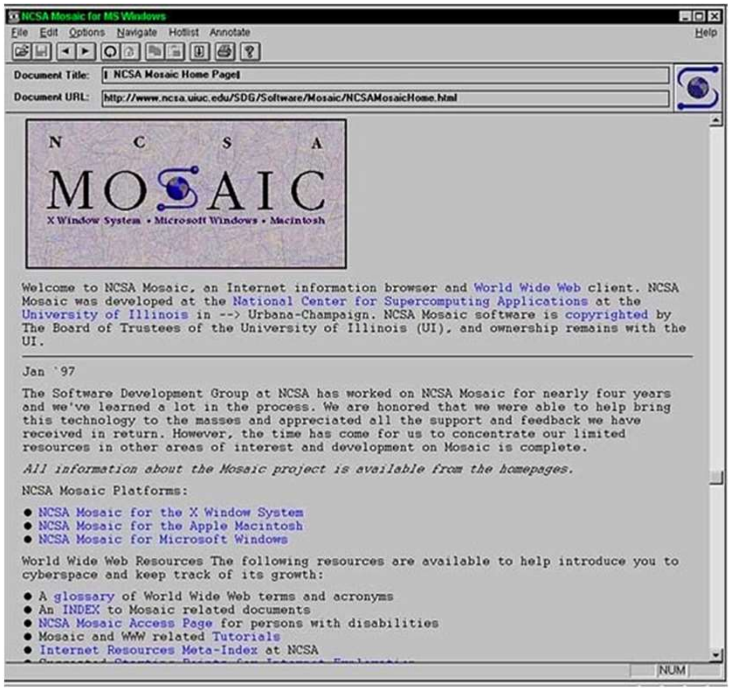
- Windowing became primary
- Added games: Solitaire, Minesweeper, and FreeCell!
  - These were a trick to teach mouse skills



Second wave: personal computing

Mosaic Web Browser (1993)

- Originally for Unix systems, later ported to Mac and Windows
- “First” graphical web browser
- Microsoft IE came in 1995
- Apple didn’t make a browser until Safari in 2003


NCSA Mosaic for MS Windows
File
Edit
Options
Navigate
Hotlist
Annotate
Help
N C S A
MOSAIC
X Window System • Microsoft Windows • Macintosh
NUM

Document Title: NCSA Mosaic Home Page

Document URL: http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/NCSAMosaicHome.html

Welcome to NCSA Mosaic, an Internet information browser and World Wide Web client. NCSA Mosaic was developed at the National Center for Supercomputing Applications at the University of Illinois in --> Urbana-Champaign. NCSA Mosaic software is copyrighted by The Board of Trustees of the University of Illinois (UI), and ownership remains with the UI.

Jan `97

The Software Development Group at NCSA has worked on NCSA Mosaic for nearly four years and we've learned a lot in the process. We are honored that we were able to help bring this technology to the masses and appreciated all the support and feedback we have received in return. However, the time has come for us to concentrate our limited resources in other areas of interest and development on Mosaic is complete.

All information about the Mosaic project is available from the homepages.

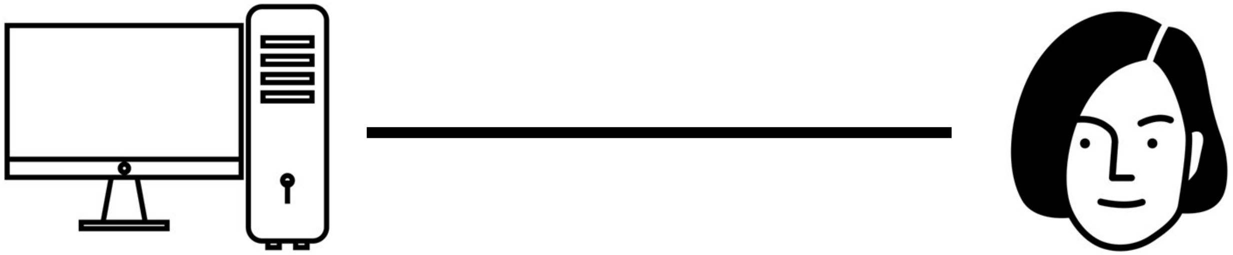
NCSA Mosaic Platforms:

- NCSA Mosaic for the X Window System
- NCSA Mosaic for the Apple Macintosh
- NCSA Mosaic for Microsoft Windows

World Wide Web Resources The following resources are available to help introduce you to cyberspace and keep track of its growth:

- A glossary of World Wide Web terms and acronyms
- An INDEX to Mosaic related documents
- NCSA Mosaic Access Page for persons with disabilities
- Mosaic and WWW related Tutorials
- Internet Resources Meta-Index at NCSA

Second wave: personal computing



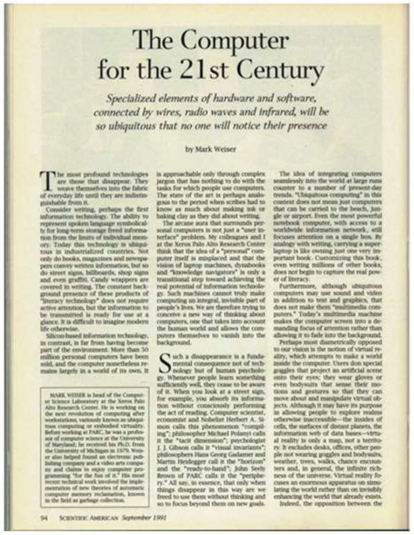
“One to one”

Three waves of computing

1. Mainframe computing
2. Personal computing
3. Ubiquitous computing

Third wave: ubiquitous computing

- Weiser speculated people would interact with three types of computers
    - Tabs: inch-scale devices, like post-its
    - Pads: foot-scale devices, like paper
    - Boards: yard-scale devices, like whiteboards
- Speculated devices would have shared ownership



The Computer for the 21st Century

Specialized elements of hardware and software, connected by wires, radio waves and infrared, will be so ubiquitous that no one will notice their presence

by Mark Weiser

The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.

Consider writing, perhaps the first information technology. The ability to represent spoken language symbolically for long-term storage freed information from the limits of individual memory. Today this technology is ubiquitous in industrialized countries. Not only do books, magazines and newspapers convey written information, but so do street signs, billboards, shop signs and even graffiti. Candy wrappers are covered in writing. The constant background presence of these products of "literacy technology" does not require active attention, but the information to be transmitted is ready for use at a glance. It is difficult to imagine modern life otherwise.

Silicon-based information technology, in contrast, is far from having become part of the environment. More than 50 million personal computers have been sold, and the computer nonetheless remains largely in a world of its own. It is approachable only through complex jargon that has nothing to do with the tasks for which people use computers. The state of the art is perhaps analogous to the period when scribes had to know as much about making ink or baking clay as they did about writing.

The arcane aura that surrounds personal computers is not just a "user interface" problem. My colleagues and I at the Xerox Palo Alto Research Center think that the idea of a "personal" computer itself is misplaced and that the vision of laptop machines, dynabooks and "knowledge navigators" is only a transitional step toward achieving the real potential of information technology. Such machines cannot truly make computing an integral, invisible part of people's lives. We are therefore trying to conceive a new way of thinking about computers, one that takes into account the human world and allows the computers themselves to vanish into the background.

Such a disappearance is a fundamental consequence not of technology but of human psychology. Whenever people learn something sufficiently well, they cease to be aware of it. When you look at a street sign, for example, you absorb its information without consciously performing the act of reading. Computer scientist, economist and Nobelist Herbert A. Simon calls this phenomenon "compiling"; philosopher Michael Polanyi calls it the "tacit dimension"; psychologist J. J. Gibson calls it "visual invariants"; philosophers Hans Georg Gadamer and Martin Heidegger call it the "horizon" and the "ready-to-hand"; John Seely Brown of PARC calls it the "periphery." All say, in essence, that only when things disappear in this way are we freed to use them without thinking and so to focus beyond them on new goals.

The idea of integrating computers seamlessly into the world at large runs counter to a number of present-day trends. "Ubiquitous computing" in this context does not mean just computers that can be carried to the beach, jungle or airport. Even the most powerful notebook computer, with access to a worldwide information network, still focuses attention on a single box. By analogy with writing, carrying a super-laptop is like owning just one very important book. Customizing this book, even writing millions of other books, does not begin to capture the real power of literacy.

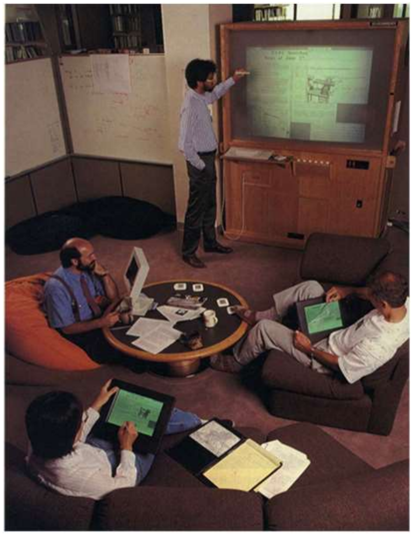
Furthermore, although ubiquitous computers may use sound and video in addition to text and graphics, that does not make them "multimedia computers." Today's multimedia machine makes the computer screen into a demanding focus of attention rather than allowing it to fade into the background.

Perhaps most diametrically opposed to our vision is the notion of virtual reality, which attempts to make a world inside the computer. Users don special goggles that project an artificial scene onto their eyes; they wear gloves or even bodysuits that sense their motions and gestures so that they can move about and manipulate virtual objects. Although it may have its purpose in allowing people to explore realms otherwise inaccessible—the insides of cells, the surfaces of distant planets, the information web of data bases—virtual reality is only a map, not a territory. It excludes desks, offices, other people not wearing goggles and bodysuits, weather, trees, walks, chance encounters and, in general, the infinite richness of the universe. Virtual reality focuses an enormous apparatus on simulating the world rather than on invisibly enhancing the world that already exists.

Indeed, the opposition between the

MARK WEISER is head of the Computer Science Laboratory at the Xerox Palo Alto Research Center. He is working on the next revolution of computing after workstations, variously known as ubiquitous computing or embodied virtuality. Before working at PARC, he was a professor of computer science at the University of Maryland; he received his Ph.D. from the University of Michigan in 1979. Weiser also helped found an electronic publishing company and a video arts company and claims to enjoy computer programming "for the fun of it." His most recent technical work involved the implementation of new theories of automatic computer memory reclamation, known in the field as garbage collection.

94 SCIENTIFIC AMERICAN September 1991

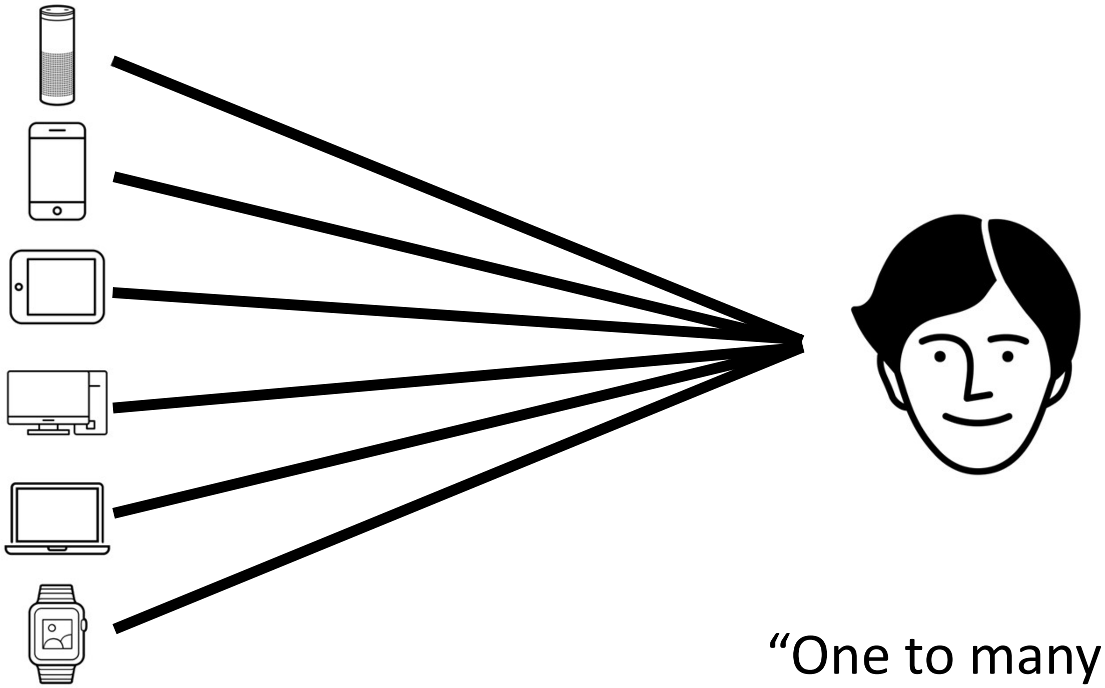


Third wave: ubiquitous computing

Third wave: ubiquitous computing

- Lines up with what we use today, for the most part
  - Tabs = phones and watches
  - Pads = tablets and laptops
  - Boards = interactive projectors? smart TVs? augmented reality?
- Still a strong sense of device ownership

Third wave: ubiquitous computing

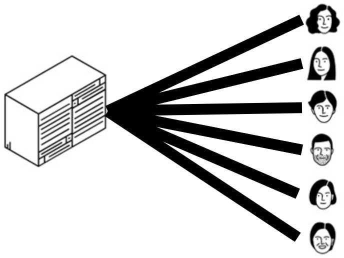


“One to many”

Three waves of computing

1

Mainframe computing



"Many to one"

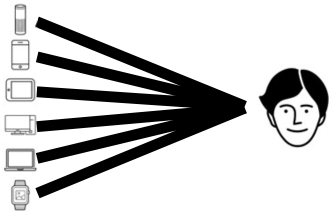
2

Personal computing

"One to one"

3

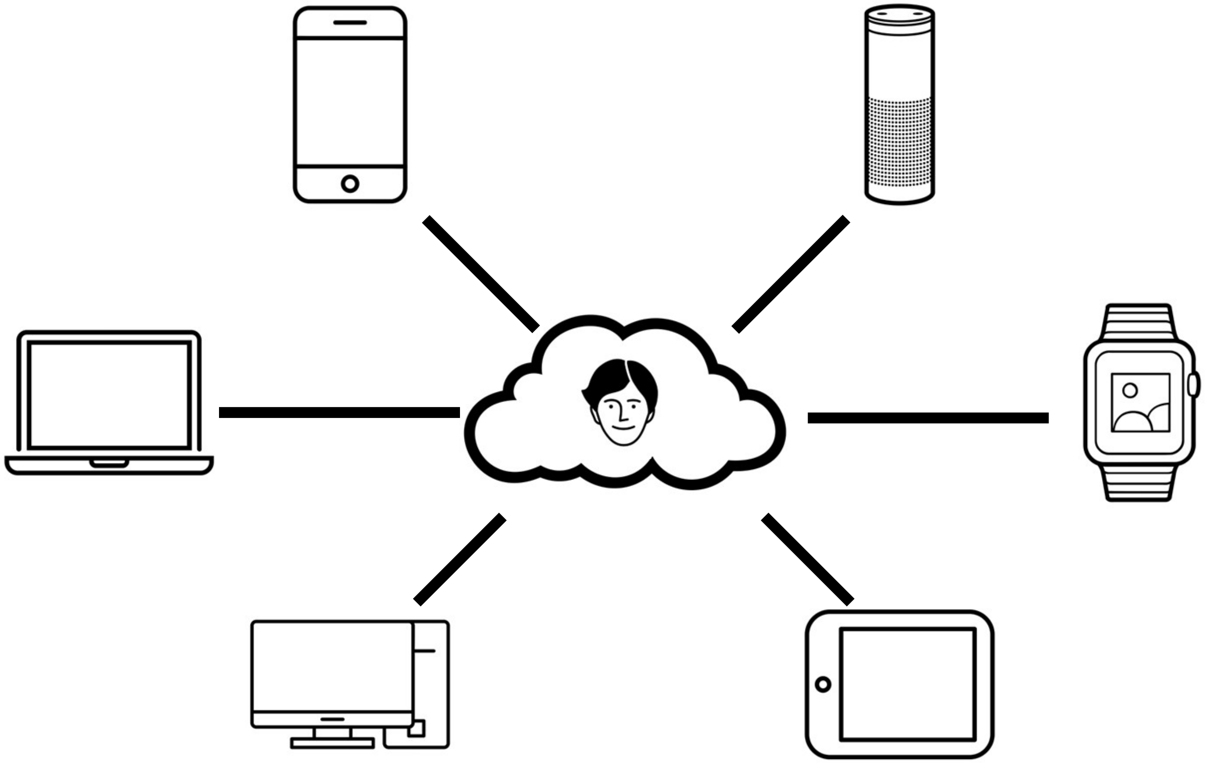
Ubiquitous computing



"One to many"

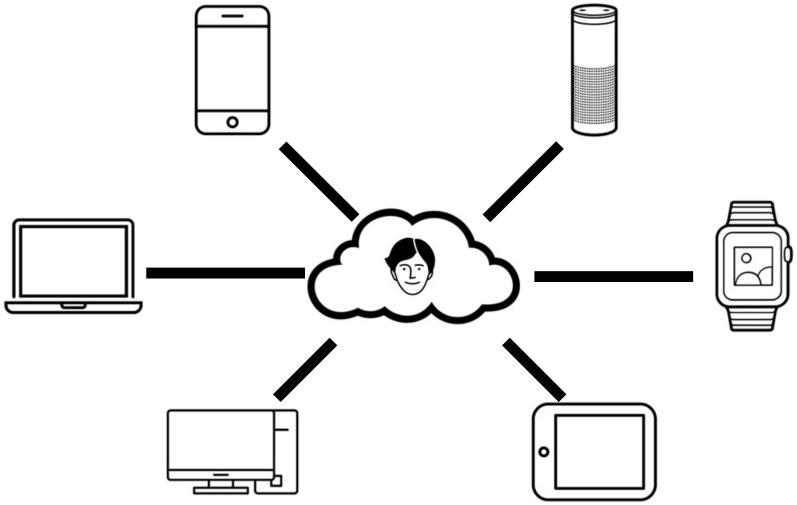
Why are web tools now the standard for interface development?

One to many, synced over the cloud



One to many, synced over the cloud

- Use HTTP requests to send data to the cloud and receive data from it
    - JavaScript provided early tools to do this
- Render that data with HTML
- Style it with CSS

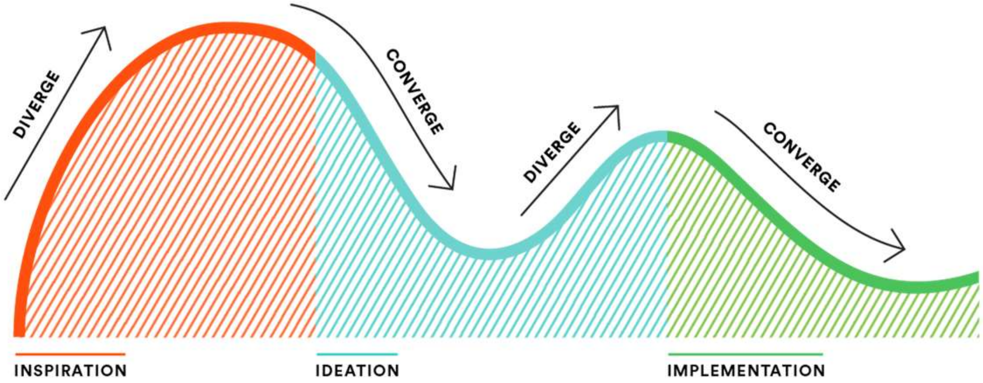


Ubiquitous computing is, in large part, why
web tools are the current standard
for interface development

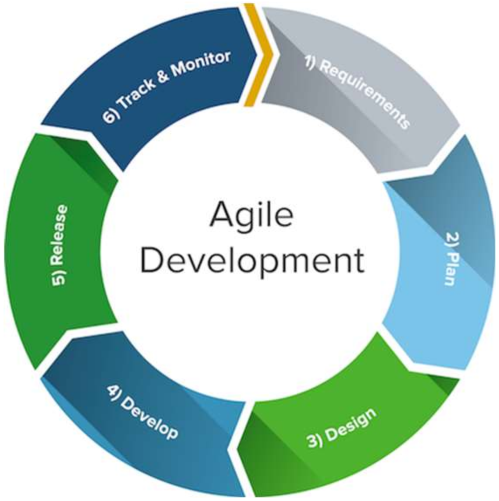
Web tools as the standard

- Nearly every platform needs to communicate with a cloud system
- Most need a web browser so people can access sites
- Shared programming language and development environment enables efficient work
- Developers can write once, deploy to many platforms
    - Hopefully customize style and functionality to the device
- Other reasons?

Product design process

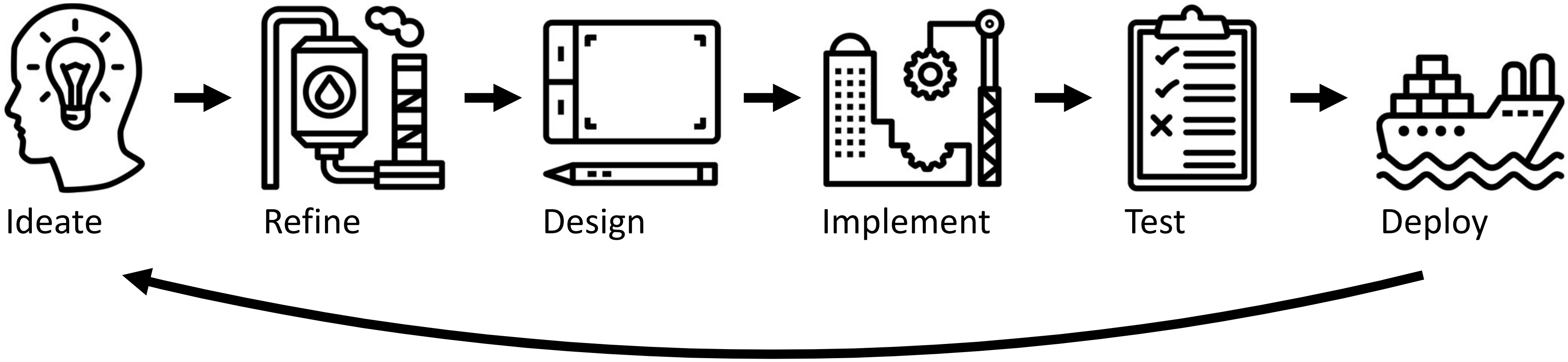


Human-Centered Design, IDEO



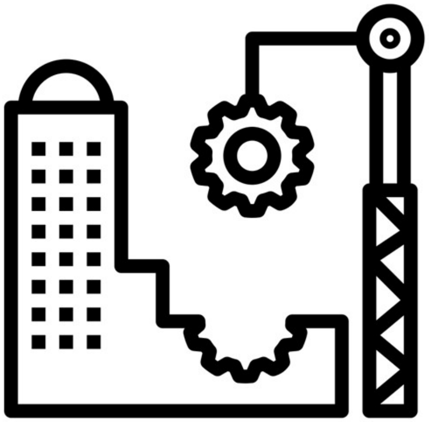
Agile Development, Agile Manifesto

Product design process, simplified



User interface implementation

- Has the power to turn ideas into reality
- Often dictates design decisions and timelines, for better or for worse
- Either you will be implementing, or you will need to communicate with your colleagues who are



What is interface implementation today?

Often HTML, CSS, and JavaScript

ANGULARJS
by Google
Bootstrap
React JS
BACKBONE.JS
METEOR
ionic
node
jQuery
ember

There are lot of languages
and development frameworks.
Why do most people use web tools?

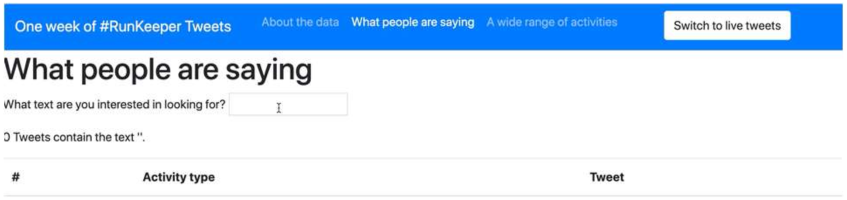
Assignments

- A1: Static web with HTML and CSS
- A2: Programming on the web
- A3: Web frameworks
- A4: Mobile development
- Final Project

A2

Runkeeper Tweet Report in JavaScript and TypeScript

- Learning goal: become comfortable with JavaScript, a widely-used development language on the web
- Will learn to use JavaScript libraries for visualization and interaction
- Optional partner



What people are saying

What text are you interested in looking for?

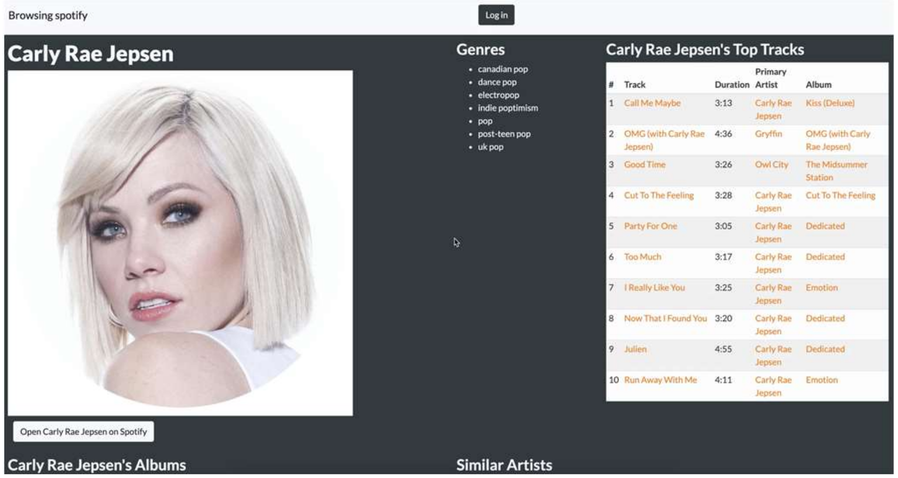
0 Tweets contain the text ''.

#	Activity type	Tweet

A3

Spotify Browser in Angular

- Learning goal: develop skills in web frameworks which separate interface from data and interaction (Model-View-Controller)
- Will make an interactive browser of Spotify's library
- Optional partner



Carly Rae Jepsen

Genres

- canadian pop
- dance pop
- electropop
- indie poptimism
- pop
- post-teen pop
- uk pop

Carly Rae Jepsen's Top Tracks

#	Track	Duration	Primary Artist	Album
1	Call Me Maybe	3:13	Carly Rae Jepsen	Kiss (Deluxe)
2	OMG (with Carly Rae Jepsen)	4:36	Gryffin	OMG (with Carly Rae Jepsen)
3	Good Time	3:26	Owl City	The Midsummer Station
4	Cut To The Feeling	3:28	Carly Rae Jepsen	Cut To The Feeling
5	Party For One	3:05	Carly Rae Jepsen	Dedicated
6	Too Much	3:17	Carly Rae Jepsen	Dedicated
7	I Really Like You	3:25	Carly Rae Jepsen	Emotion
8	Now That I Found You	3:20	Carly Rae Jepsen	Dedicated
9	Julien	4:55	Carly Rae Jepsen	Dedicated
10	Run Away With Me	4:11	Carly Rae Jepsen	Emotion

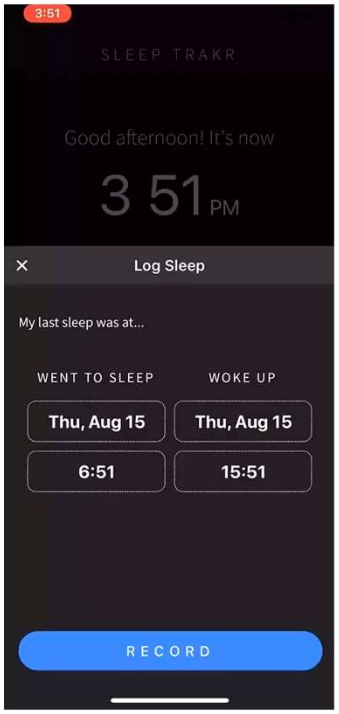
Carly Rae Jepsen's Albums

Similar Artists

A4

Sleep Tracker Mobile App

- Learning goal: learn to leverage UI components in a mobile framework and align with principles of good mobile design
- Will implement an app to log daily sleep
- Optional partner



3:51

SLEEP TRAKR

Good afternoon! It's now

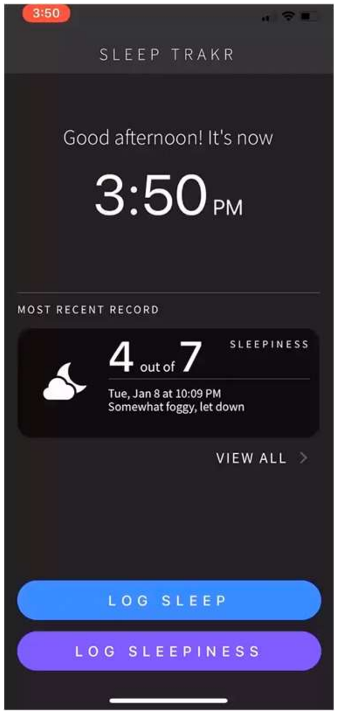
3 51 PM

✕ Log Sleep

My last sleep was at...

WENT TO SLEEP	WOKE UP
Thu, Aug 15	Thu, Aug 15
6:51	15:51

RECORD



3:50

SLEEP TRAKR

Good afternoon! It's now

3:50 PM

MOST RECENT RECORD

4 out of 7 SLEEPINESS

Tue, Jan 8 at 10:09 PM
Somewhat foggy, let down

VIEW ALL >

LOG SLEEP

LOG SLEEPINESS

A5

Final Project

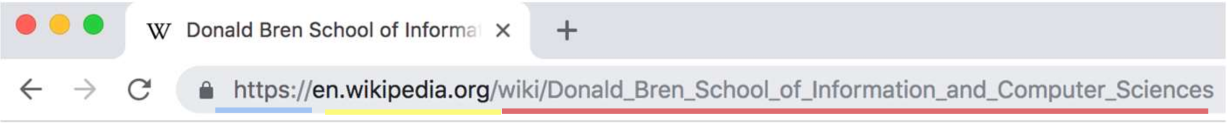
- Learning goal: Apply principles of user interface design to build an alternative mode of interaction
- Implement with a web, mobile, or wearable framework of your choice
- Optional partner

Client-side web development

Your browser

Web server

Using the internet

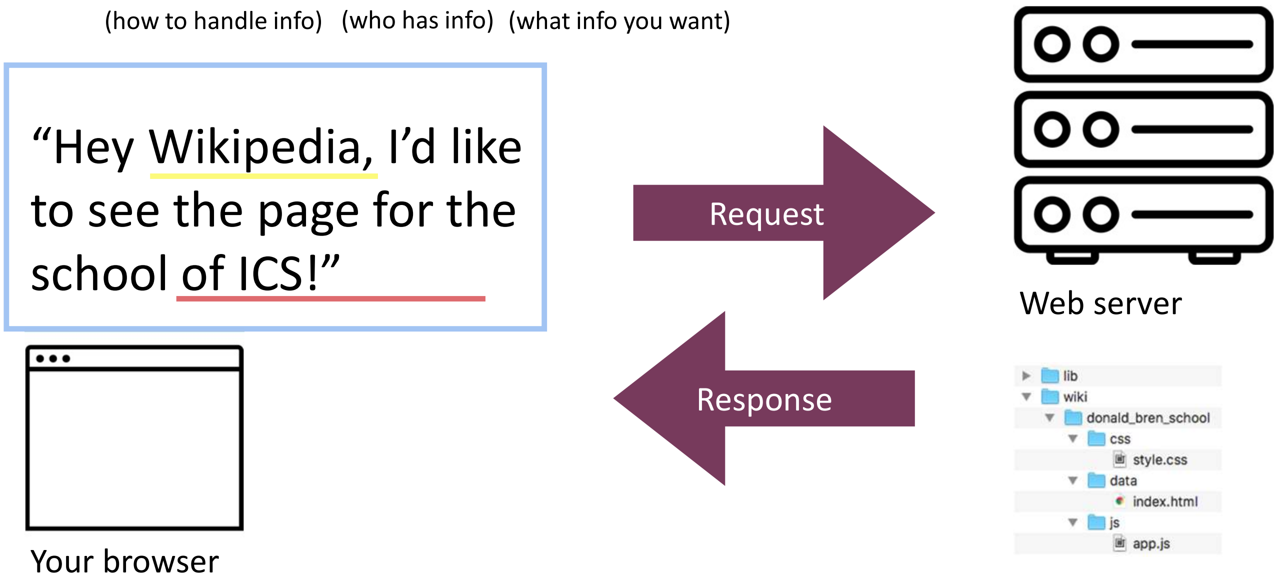


https://en.wikipedia.org/wiki/Donald_Bren_School_of_Information_and_Computer_Sciences

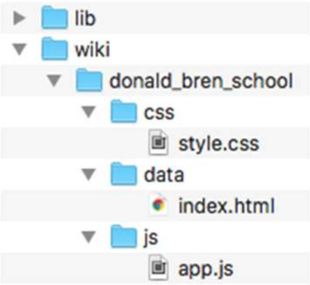
Protocol (how to handle info)

Host (who has info)

Resource (what info you want)



"Hey Wikipedia, I'd like to see the page for the school of ICS!"

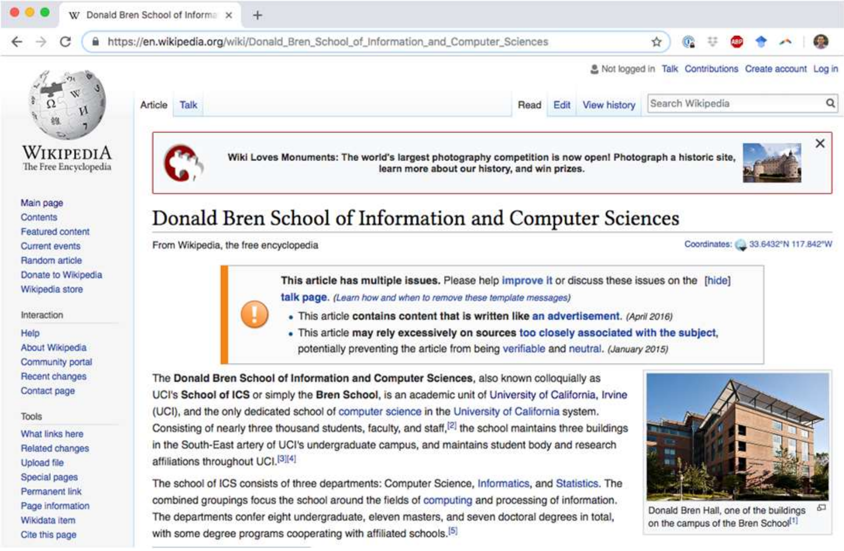


```
▶ lib
▼ wiki
  ▼ donald_bren_school
    ▼ css
        style.css
    ▼ data
        index.html
    ▼ js
        app.js
```

+

Opera
Google Chrome
Safari
Mozilla Firefox
Internet Explorer
Microsoft Edge

=



Wiki Loves Monuments: The world's largest photography competition is now open! Photograph a historic site, learn more about our history, and win prizes.

Donald Bren School of Information and Computer Sciences

From Wikipedia, the free encyclopedia

Coordinates: 33.6432°N 117.842°W

This article has multiple issues. Please help **improve it** or discuss these issues on the **talk page**. *(Learn how and when to remove these template messages)* [hide]

- This article **contains content that is written like an advertisement.** *(April 2016)*
- This article **may rely excessively on sources too closely associated with the subject**, potentially preventing the article from being verifiable and neutral. *(January 2015)*

The **Donald Bren School of Information and Computer Sciences**, also known colloquially as UCI's **School of ICS** or simply the **Bren School**, is an academic unit of University of California, Irvine (UCI), and the only dedicated school of computer science in the University of California system. Consisting of nearly three thousand students, faculty, and staff,[2] the school maintains three buildings in the South-East artery of UCI's undergraduate campus, and maintains student body and research affiliations throughout UCI.[3][4]

The school of ICS consists of three departments: Computer Science, Informatics, and Statistics. The combined groupings focus the school around the fields of computing and processing of information. The departments confer eight undergraduate, eleven masters, and seven doctoral degrees in total, with some degree programs cooperating with affiliated schools.[5]

Donald Bren Hall, one of the buildings on the campus of the Bren School[1]

Fundamentally, the web is designed to send files around

So what does a file on the web look like?

Take a minute to create a file, name it with the extension ‘html’

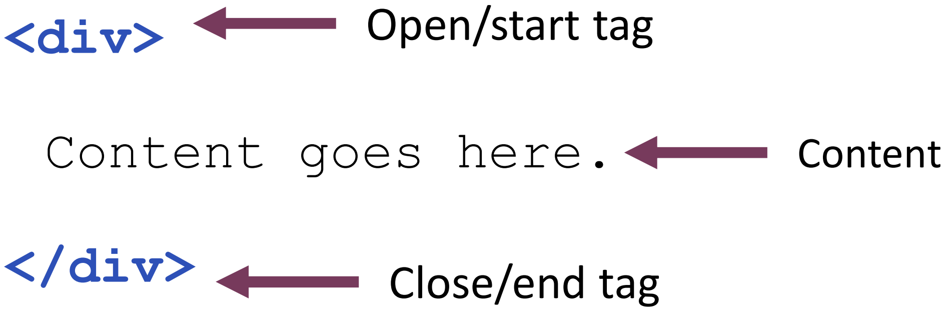
Ex: mypage.html

What if we wanted to specify how the content is rendered?

HTML (HyperText Markup Language)

- Adds meaning to text
- Links documents to one another
  - Vanneaver Bush, hypertext vision

Tags



```
<div>
 Content goes here.
</div>
```

Open/start tag

Content

Close/end tag

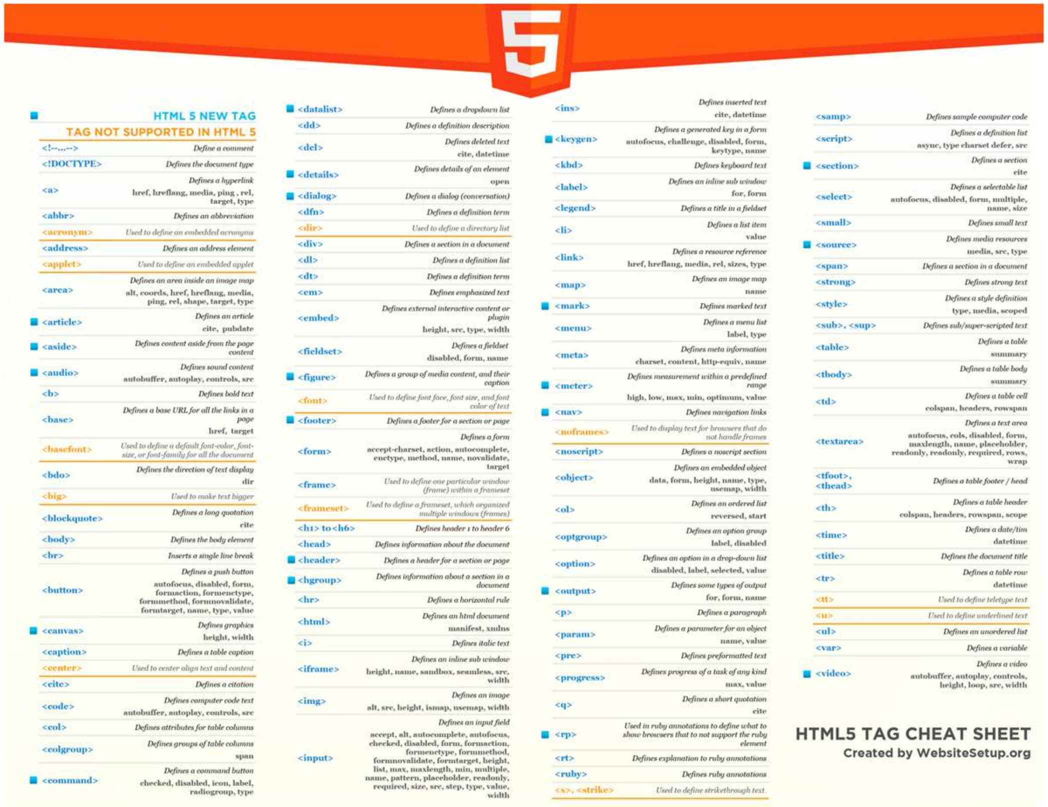
Whitespace and tag case are ignored

Some common tags

```
<h1>Heading level 1</h1>
<h2>Heading level 2</h2>
...
<p>A paragraph</p>
<!--A comment-->
<img> An image
<ul> An unordered list (bullets)
<li> A list item
<table> A data table
<strong> Important content (bolded)
<em> Emphasized content (italicized)
<div> A division (section) of content
```

Tags

- There are hundreds of tags!
- You may not use them all, but it’s good to explore them
- Search on Google or W3C to understand each tag’s purpose
- https://www.w3schools.com/tags/



■ HTML 5 NEW TAG

TAG NOT SUPPORTED IN HTML 5

	<!--...-->	*Define a comment*	
	<!DOCTYPE>	*Defines the document type*	
	<a>	*Defines a hyperlink*	href, hreflang, media, ping , rel, target, type
	<abbr>	*Defines an abbreviation*	
	<acronym>	*Used to define an embedded acronyms*	
	<address>	*Defines an address element*	
	<applet>	*Used to define an embedded applet*	
	<area>	*Defines an area inside an image map*	alt, coords, href, hreflang, media, ping, rel, shape, target, type
■	<article>	*Defines an article*	cite, pubdate
■	<aside>	*Defines content aside from the page content*	
■	<audio>	*Defines sound content*	autobuffer, autoplay, controls, src
	<b>	*Defines bold text*	
	<base>	*Defines a base URL for all the links in a page*	href, target
	<basefont>	*Used to define a default font-color, font-size, or font-family for all the document*	
	<bdo>	*Defines the direction of text display*	dir
	<big>	*Used to make text bigger*	
	<blockquote>	*Defines a long quotation*	cite
	<body>	*Defines the body element*	
	<br>	*Inserts a single line break*	
	<button>	*Defines a push button*	autofocus, disabled, form, formaction, formenctype, formmethod, formnovalidate, formtarget, name, type, value
■	<canvas>	*Defines graphics*	height, width
	<caption>	*Defines a table caption*	
	<center>	*Used to center align text and content*	
	<cite>	*Defines a citation*	
	<code>	*Defines computer code text*	autobuffer, autoplay, controls, src
	<col>	*Defines attributes for table columns*	
	<colgroup>	*Defines groups of table columns*	span
■	<command>	*Defines a command button*	checked, disabled, icon, label, radiogroup, type
■	<datalist>	*Defines a dropdown list*	
	<dd>	*Defines a definition description*	
	<del>	*Defines deleted text*	cite, datetime
■	<details>	*Defines details of an element*	open
■	<dialog>	*Defines a dialog (conversation)*	
	<dfn>	*Defines a definition term*	
	<dir>	*Used to define a directory list*	
	<div>	*Defines a section in a document*	
	<dl>	*Defines a definition list*	
	<dt>	*Defines a definition term*	
	<em>	*Defines emphasized text*	
	<embed>	*Defines external interactive content or plugin*	height, src, type, width
	<fieldset>	*Defines a fieldset*	disabled, form, name
■	<figure>	*Defines a group of media content, and their caption*	
	<font>	*Used to define font face, font size, and font color of text*	
■	<footer>	*Defines a footer for a section or page*	
	<form>	*Defines a form*	accept-charset, action, autocomplete, enctype, method, name, novalidate, target
	<frame>	*Used to define one particular window (frame) within a frameset*	
	<frameset>	*Used to define a frameset, which organized multiple windows (frames)*	
	<h1> to <h6>	*Defines header 1 to header 6*	
	<head>	*Defines information about the document*	
■	<header>	*Defines a header for a section or page*	
■	<hgroup>	*Defines information about a section in a document*	
	<hr>	*Defines a horizontal rule*	
	<html>	*Defines an html document*	manifest, xmlns
	<i>	*Defines italic text*	
	<iframe>	*Defines an inline sub window*	height, name, sandbox, seamless, src, width
	<img>	*Defines an image*	alt, src, height, ismap, usemap, width
	<input>	*Defines an input field*	accept, alt, autocomplete, autofocus, checked, disabled, form, formaction, formenctype, formmethod, formnovalidate, formtarget, height, list, max, maxlength, min, multiple, name, pattern, placeholder, readonly, required, size, src, step, type, value, width
	<ins>	*Defines inserted text*	cite, datetime
■	<keygen>	*Defines a generated key in a form*	autofocus, challenge, disabled, form, keytype, name
	<kbd>	*Defines keyboard text*	
	<label>	*Defines an inline sub window*	for, form
	<legend>	*Defines a title in a fieldset*	
	<li>	*Defines a list item*	value
	<link>	*Defines a resource reference*	href, hreflang, media, rel, sizes, type
	<map>	*Defines an image map*	name
■	<mark>	*Defines marked text*	
	<menu>	*Defines a menu list*	label, type
	<meta>	*Defines meta information*	charset, content, http-equiv, name
■	<meter>	*Defines measurement within a predefined range*	high, low, max, min, optimum, value
■	<nav>	*Defines navigation links*	
	<noframes>	*Used to display text for browsers that do not handle frames*	
	<noscript>	*Defines a noscript section*	
	<object>	*Defines an embedded object*	data, form, height, name, type, usemap, width
	<ol>	*Defines an ordered list*	reversed, start
	<optgroup>	*Defines an option group*	label, disabled
	<option>	*Defines an option in a drop-down list*	disabled, label, selected, value
■	<output>	*Defines some types of output*	for, form, name
	<p>	*Defines a paragraph*	
	<param>	*Defines a parameter for an object*	name, value
	<pre>	*Defines preformatted text*	
	<progress>	*Defines progress of a task of any kind*	max, value
	<q>	*Defines a short quotation*	cite
■	<rp>	*Used in ruby annotations to define what to show browsers that to not support the ruby element*	
	<rt>	*Defines explanation to ruby annotations*	
	<ruby>	*Defines ruby annotations*	
	<s>, <strike>	*Used to define strikethrough text*	
	<samp>	*Defines sample computer code*	
	<script>	*Defines a definition list*	async, type charset defer, src
■	<section>	*Defines a section*	cite
	<select>	*Defines a selectable list*	autofocus, disabled, form, multiple, name, size
	<small>	*Defines small text*	
■	<source>	*Defines media resources*	media, src, type
	<span>	*Defines a section in a document*	
	<strong>	*Defines strong text*	
	<style>	*Defines a style definition*	type, media, scoped
	<sub>, <sup>	*Defines sub/super-scripted text*	
	<table>	*Defines a table*	summary
	<tbody>	*Defines a table body*	summary
	<td>	*Defines a table cell*	colspan, headers, rowspan
	<textarea>	*Defines a text area*	autofocus, cols, disabled, form, maxlength, name, placeholder, readonly, readonly, required, rows, wrap
	<tfoot>, <thead>	*Defines a table footer / head*	
	<th>	*Defines a table header*	colspan, headers, rowspan, scope
	<time>	*Defines a date/tim*	datetime
	<title>	*Defines the document title*	
	<tr>	*Defines a table row*	datetime
	<tt>	*Used to define teletype text*	
	<u>	*Used to define underlined text*	
	<ul>	*Defines an unordered list*	
	<var>	*Defines a variable*	
■	<video>	*Defines a video*	autobuffer, autoplay, controls, height, loop, src, width

HTML5 TAG CHEAT SHEET

Created by WebsiteSetup.org

How would you specify a `<div>` with the `<p>` (paragraph) `I` **`love`** `HTML!` ?

```
<div><p>I <strong>love HTML!

<div><p>I <strong>love</strong> HTML!</p>

<div><p>I <strong>love<strong> HTML!<p><div>

<div><p>I <strong>love</strong> HTML!</p></div>

<div><p>I </p><strong>love</strong><p> HTML!</p></div>
```

How would you specify a `<div>` with the `<p>` (paragraph) `I` **`love`** `HTML!` ?

`<div><p>I <strong>love HTML!`

`<div><p>I <strong>love</strong> HTML!</p>`

`<div><p>I <strong>love<strong> HTML!<p><div>`

➡ `<div><p>I <strong>love</strong> HTML!</p></div>`

`<div><p>I </p><strong>love</strong><p> HTML!</p></div>`

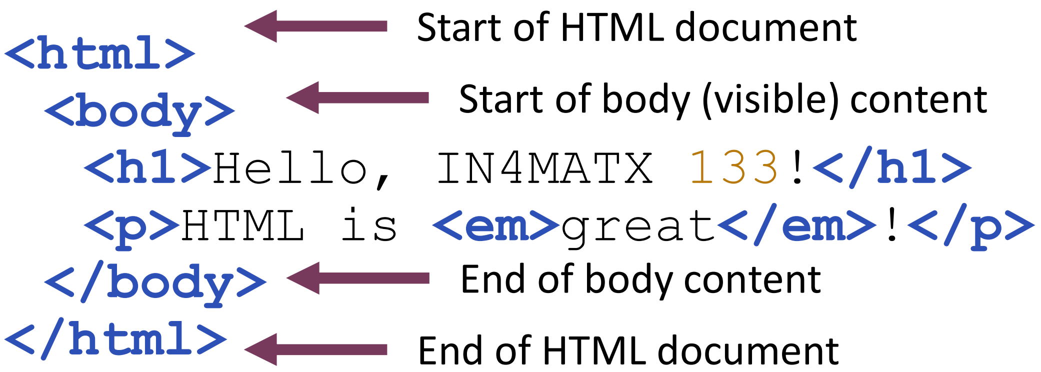
Nesting

- The Content of a tag can contain other HTML tags

```
<div><p>I <strong>love</strong> HTML!</p></div>
```

Nesting: HTML

- By convention, HTML is specified via the Content of an `<html>` element.



```
<html>
  <body>
    <h1>Hello, IN4MATX 133!</h1>
    <p>HTML is <em>great</em>!</p>
  </body>
</html>
```

Start of HTML document

Start of body (visible) content

End of body content

End of HTML document

Attributes

- Attributes specify options and add meaning
- Attributes are space-separated lists of names and values.
  - Kind of like variables
  - Almost always Strings

```
<div attributeA="valueA" attributeB="valueB">
 Content goes here
</div>
```

Attributes

```
<a href="http://inf133-fa20.baldwin.in/">IN4MATX 133</a>
```

anchor (hyperlink)

hypertext reference

```
<img src="logo.jpg" alt="My Dog Dylan Playing Pawball">
```

source

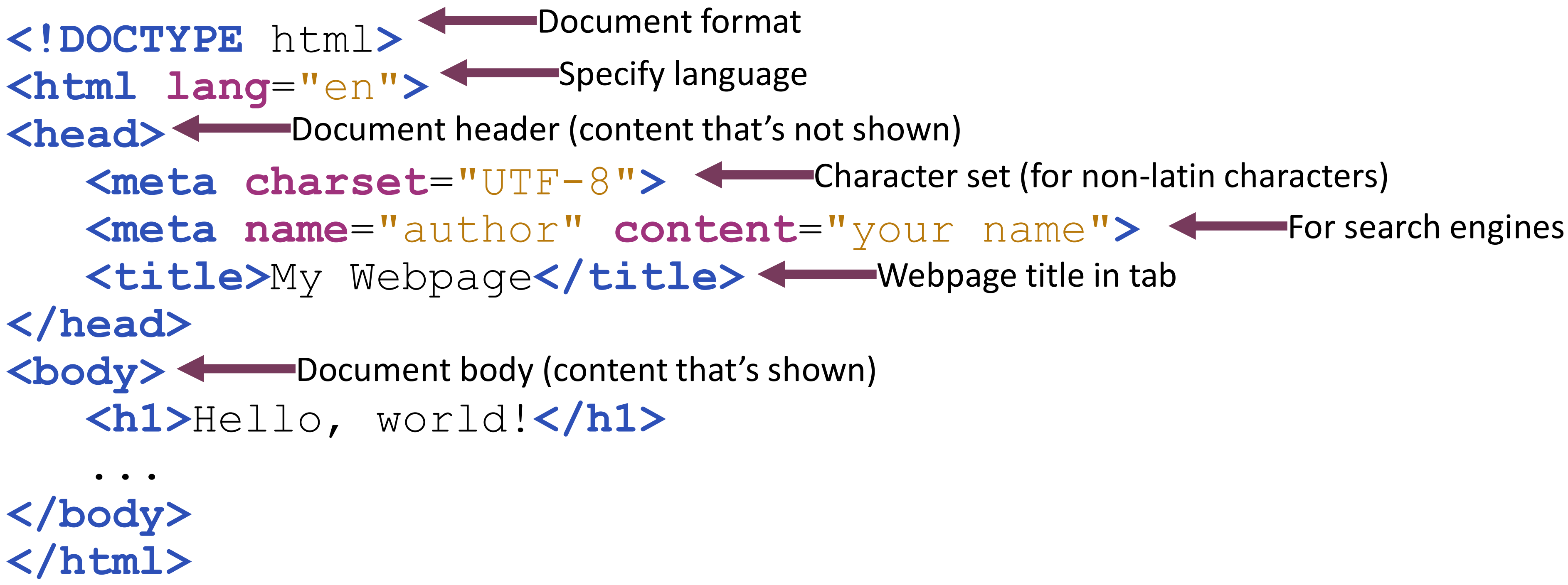
alternative text for screen readers

`img` tags have no (text) content, so no closing tag

```
<html lang="en">
...
</html>
```

Language of document is English

HTML structure



```
<!DOCTYPE html>  ← Document format
<html lang="en">  ← Specify language
<head> ← Document header (content that’s not shown)
    <meta charset="UTF-8">  ← Character set (for non-latin characters)
    <meta name="author" content="your name">  ← For search engines
    <title>My Webpage</title>  ← Webpage title in tab
</head>
<body> ← Document body (content that’s shown)
    <h1>Hello, world!</h1>
    ...
</body>
</html>
```

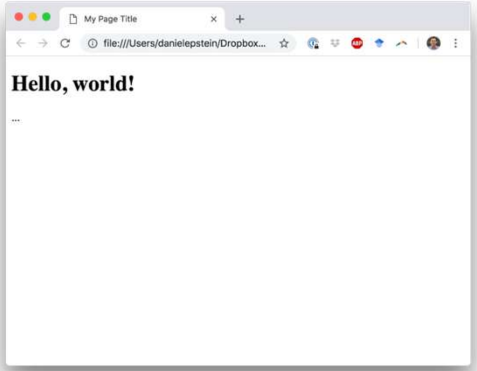
HTML structure

- Surprisingly, browsers are accommodating about HTML structure
- No “compiler errors”
- However, validation can help ensure browser compatibility and site usability

HTML structure

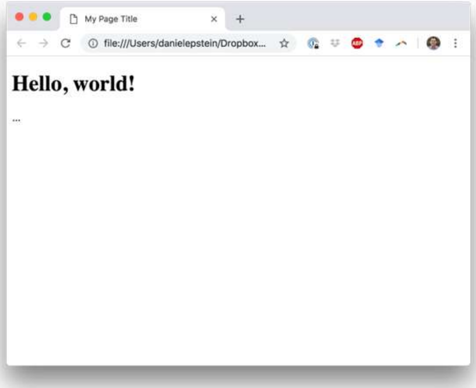
```html
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="author" content="your name">
   <title>My Page Title</title>
</head>
<body>
   <h1>Hello, world!</h1>
   ...
</body>
</html>
```

```html
<html>
<head>
   <title>My Page Title</title>
</head>
<body>
   <h1>Hello, world!</h1>
   <p>...
```



Hello, world!

...



Hello, world!

...

Let’s make a shopping list

Mark’s shopping list

- Milk
- Eggs
- Sandwich ingredients:
  - Bread
  - Tomato
  - Lettuce

W3C validator

https://validator.w3.org/

Today’s goals

By the end of today, you should be able to…

- CONTINUE….Describe how society got to today’s ubiquitous computing
- Hypothesize why web technology has become the de-facto tool for interface development
- Describe the fundamentals of web communication
- Identify the syntax of HTML tags and attributes and describe their roles
- Create a HTML template which follows W3C specifications