

# **IN4MATX 133: User Interface Software**

Lecture 2:  
HTML & Accessibility

# Announcements

- Discussion tomorrow: Git and GitHub
- Office hours schedule posted, now with locations

# Announcements

- PollEV for participation credit will begin on Tuesday
  - Assuming the “practice round” today goes smoothly
- A2 is posted, though not due for a while (February 5th)
  - Trying to work ahead as the APIs in A3-A5 have changed significantly

# **Live demos**

# Live demos

- When possible, I'll try to live demo coding concepts during lecture
- I'll aim to post the starter and solution code on the class website before class
  - You can follow along, and I'm including the solution if I go too fast
- Live coding is hard! I will inevitably mess up
  - Unmute and correct me when you see mistakes



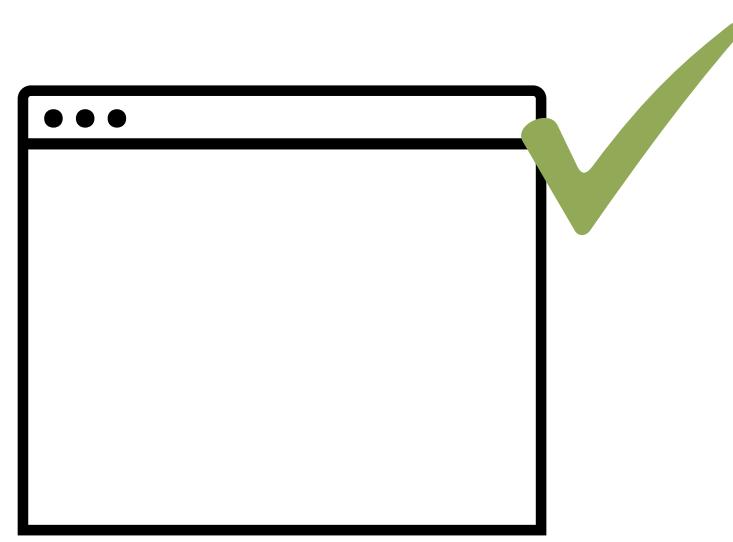
# Today's goals

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

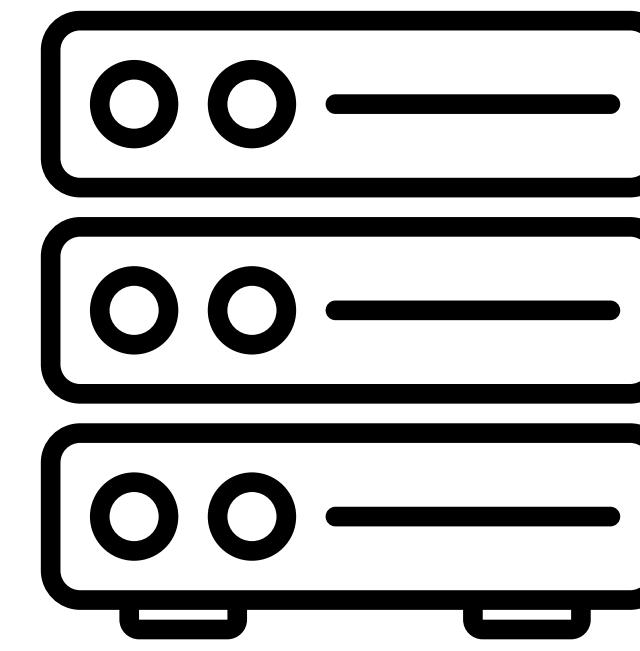
- Describe the fundamentals of web communication
- Identify the syntax of HTML tags and attributes and describe their roles
- Create a HTML template which follows W3C specifications
- Explain the importance of accessible and semantically meaningful markup
- Generate markup which meets accessibility standards

# Client-side web development

---

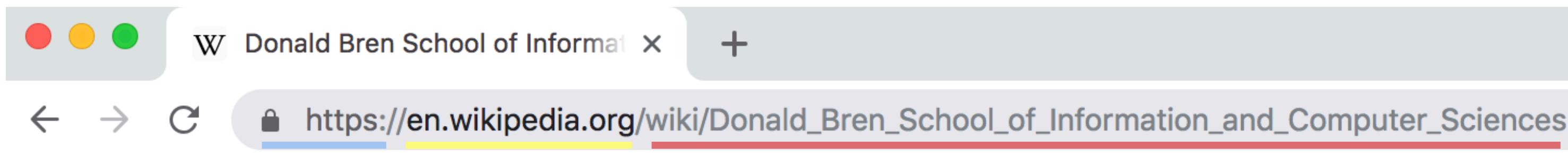


Your browser

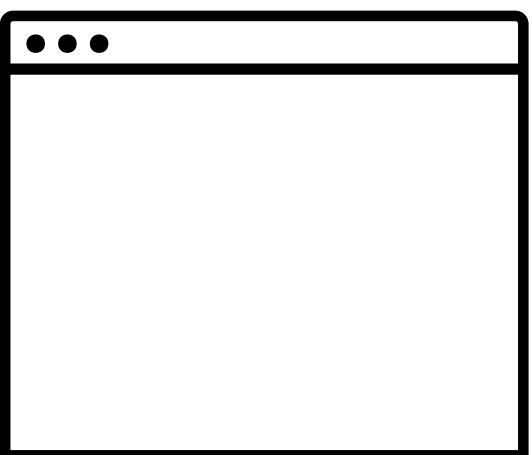


Web server

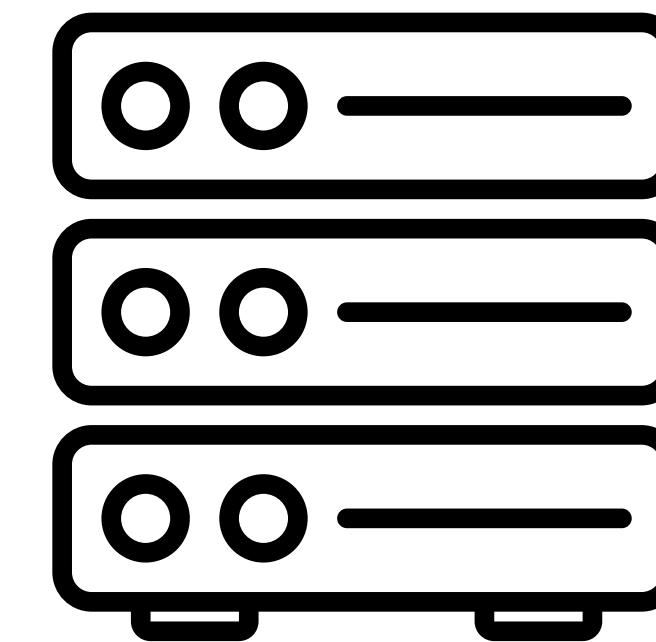
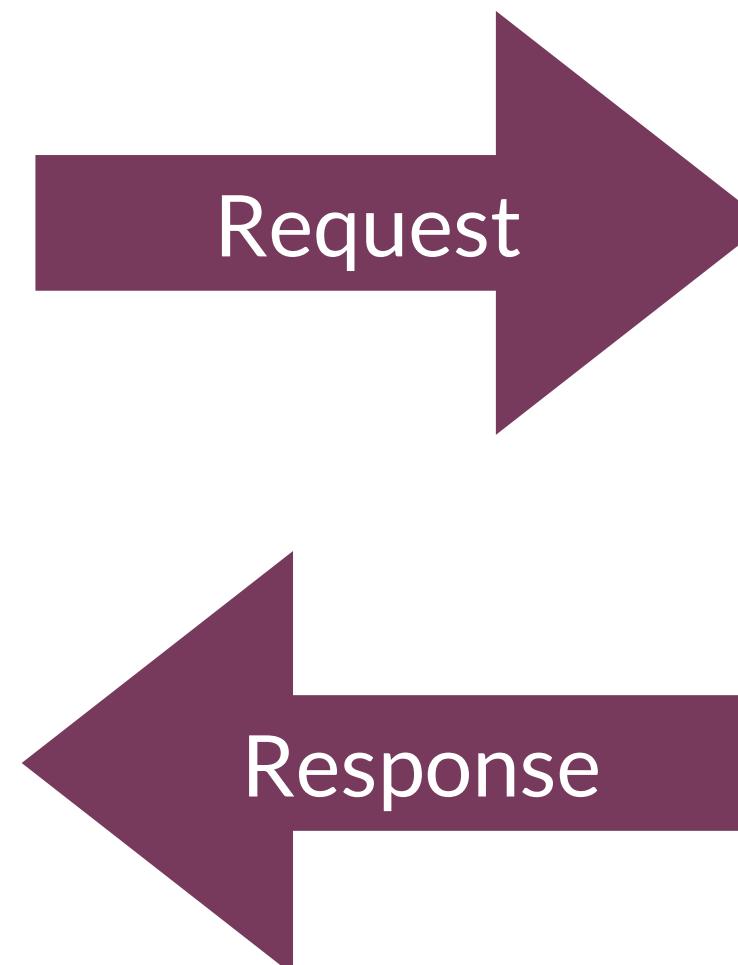
# Using the internet



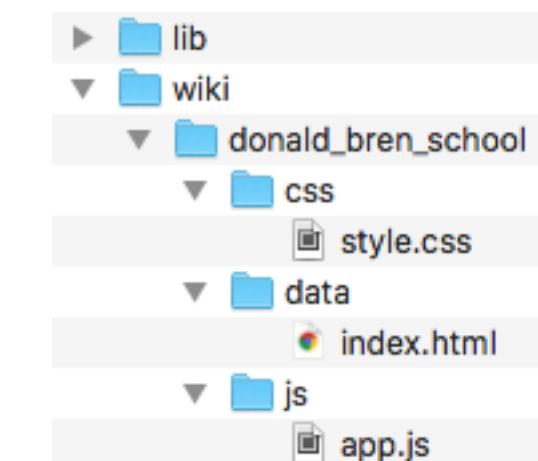
"Hey Wikipedia, I'd like to see the page for the school of ICS!"

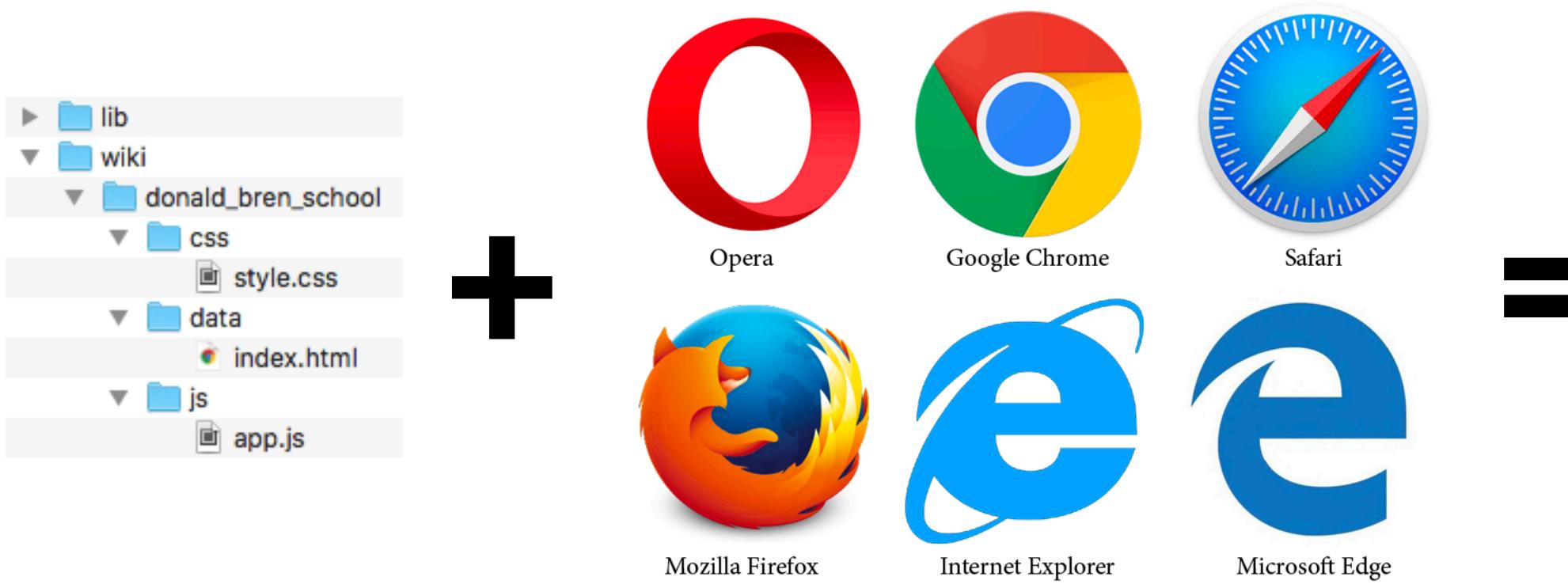


Your browser



Web server





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 [\[hide\]](#) [talk page](#). ([Learn how and when to remove these template messages](#))

- 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,<sup>[2]</sup> the school maintains three buildings in the South-East artery of UCI's undergraduate campus, and maintains student body and research affiliations throughout UCI.<sup>[3][4]</sup>

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.<sup>[5]</sup>



Donald Bren Hall, one of the buildings on the campus of the Bren School<sup>[1]</sup>

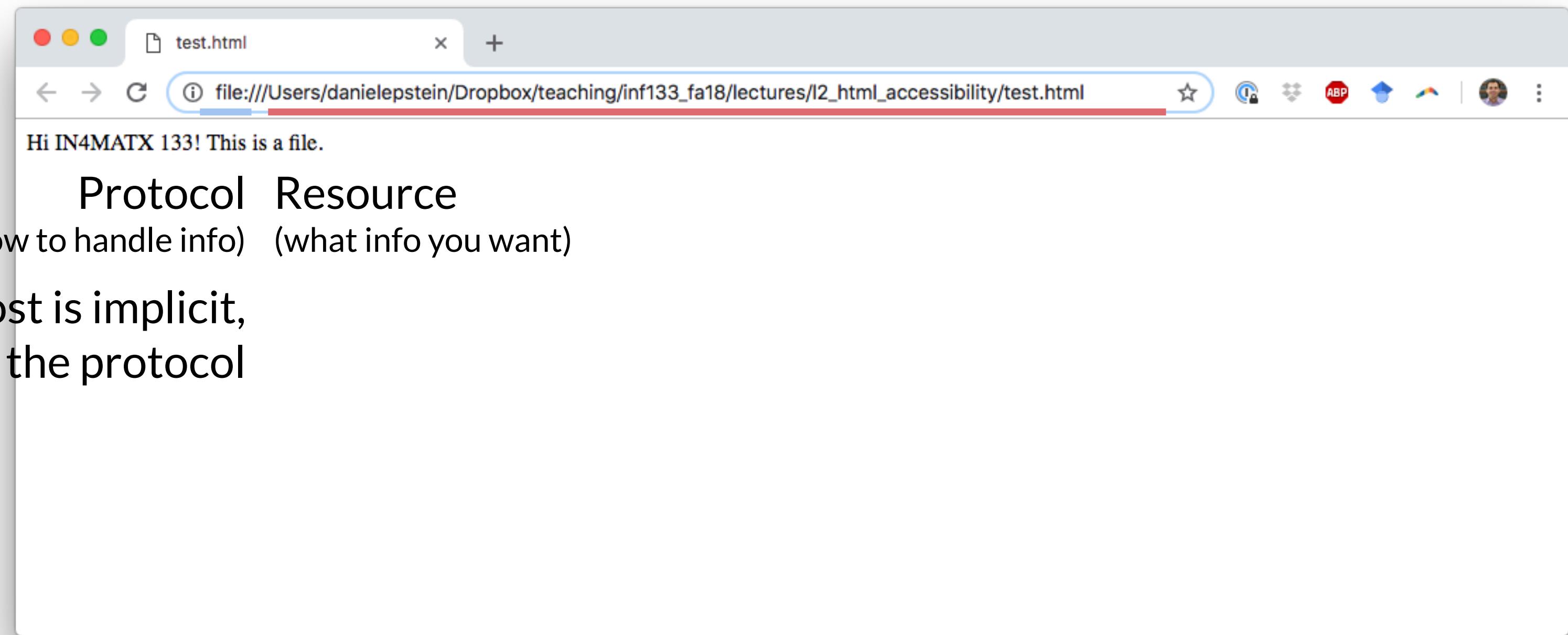
**Fundamentally, the web is  
designed to send files around**

**So what does a file on the web look like?**

A screenshot of a dark-themed code editor window titled "test.html". The editor displays a single line of code: "1 Hi IN4MATX 133! This is a file.". The status bar at the bottom shows "Line 1, Column 32", "Tab Size: 4", and "HTML".

```
1 Hi IN4MATX 133! This is a file.
```

# Same file, opened in Chrome



**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>` ← Open/start tag

Content goes here. ← Content

`</div>` ← 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

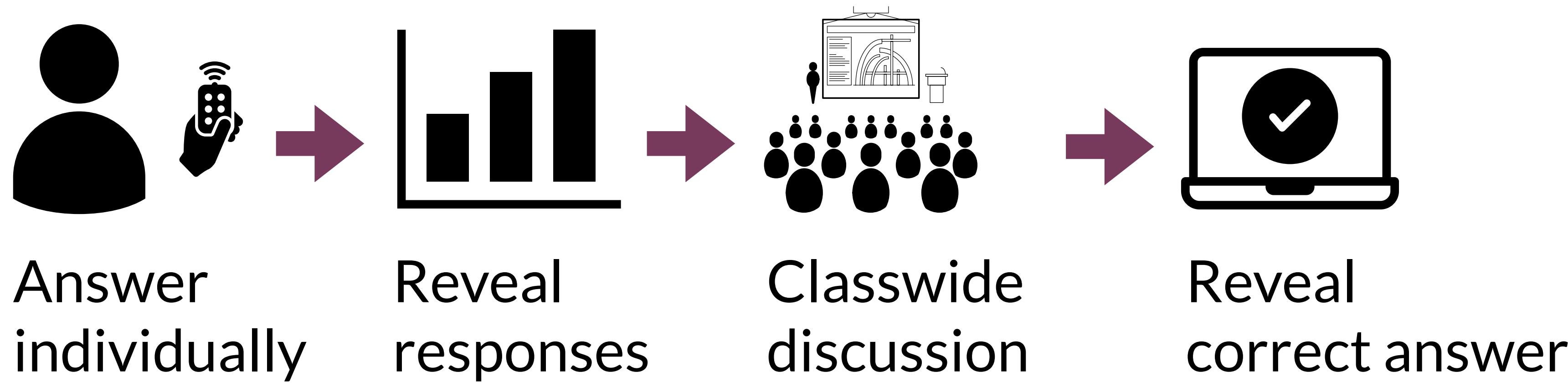
- There are hundreds of tags!
- You may not use them all, but it's good to explore them
- Search on Google or W3C to understand each tag's purpose
- <https://www.w3schools.com/tags/>



| HTML 5 NEW TAG                     |  |
|------------------------------------|--|
| <b>TAG NOT SUPPORTED IN HTML 5</b> |  |
| <!--...-->                         | Define a comment   |
| <!DOCTYPE>                         | Defines the document type  |
| <a>                                | Defines a hyperlink<br>href, hreflang, media, ping, rel, target, type  |
| <abbr>                             | Defines an abbreviation  |
| <acronym>                          | Used to define an embedded acronym   |
| <address>                          | Defines an address element   |
| <applet>                           | Used to define an embedded applet  |
| <area>                             | Defines an area inside an image map<br>alt, coords, href, hreflang, media, ping, rel, shape, target, type  |
| <article>                          | Defines an article<br>cite, pubdate  |
| <aside>                            | Defines content aside from the page content  |
| <audio>                            | Defines sound content<br>autobuffer, autoplay, controls, src   |
| <b>                                | Defines bold text  |
| <base>                             | Defines a base URL for all the links in a page<br>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<br>dir   |
| <big>                              | Used to make text bigger   |
| <blockquote>                       | Defines a long quotation<br>cite   |
| <body>                             | Defines the body element   |
| <br>                               | Inserts a single line break  |
| <button>                           | Defines a push button<br>autofocus, disabled, form, formaction, formenctype, formmethod, formnovalidate, formtarget, name, type, value   |
| <canvas>                           | Defines graphics<br>height, width  |
| <caption>                          | Defines a table caption  |
| <center>                           | Used to center align text and content  |
| <cite>                             | Defines a citation   |
| <code>                             | Defines computer code text<br>autobuffer, autoplay, controls, src  |
| <col>                              | Defines attributes for table columns   |
| <colgroup>                         | Defines groups of table columns<br>span  |
| <command>                          | Defines a command button<br>checked, disabled, icon, label, radiogroup, type   |
| <datalist>                         | Defines a dropdown list  |
| <dd>                               | Defines a definition description   |
| <del>                              | Defines deleted text<br>cite, datetime   |
| <details>                          | Defines details of an element<br>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<br>height, src, type, width   |
| <fieldset>                         | Defines a fieldset<br>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<br>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<br>manifest, xhtml  |
| <i>                                | Defines italic text  |
| <iframe>                           | Defines an inline sub window<br>height, name, sandbox, seamless, src, width  |
| <img>                              | Defines an image<br>alt, src, height, ismap, usemap, width   |
| <input>                            | Defines an input field<br>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<br>cite, datetime  |
| <keygen>                           | Defines a generated key in a form<br>autofocus, challenge, disabled, form, keytype, name   |
| <kbd>                              | Defines keyboard text  |
| <label>                            | Defines an inline sub window<br>for, form  |
| <legend>                           | Defines a title in a fieldset  |
| <li>                               | Defines a list item<br>value   |
| <link>                             | Defines a resource reference<br>href, hreflang, media, rel, sizes, type  |
| <map>                              | Defines an image map<br>name   |
| <mark>                             | Defines marked text  |
| <menu>                             | Defines a menu list<br>label, type   |
| <meta>                             | Defines meta information<br>charset, content, http-equiv, name   |
| <meter>                            | Defines measurement within a predefined range<br>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<br>data, form, height, name, type, usemap, width  |
| <ol>                               | Defines an ordered list<br>reversed, start   |
| <optgroup>                         | Defines an option group<br>label, disabled   |
| <option>                           | Defines an option in a drop-down list<br>disabled, label, selected, value  |
| <output>                           | Defines some types of output<br>for, form, name  |
| <p>                                | Defines a paragraph  |
| <param>                            | Defines a parameter for an object<br>name, value   |
| <pre>                              | Defines preformatted text  |
| <progress>                         | Defines progress of a task of any kind<br>max, value   |
| <q>                                | Defines a short quotation<br>cite  |
| <rp>                               | Used in ruby annotations to define what to show browsers that do not support the ruby element  |
| <rt>                               | Defines explanation to ruby annotations  |
| <ruby>                             | Defines ruby annotations   |
| <s>, <strike>                      | Used to define strikethrough text  |

HTML5 TAG CHEAT SHEET  
Created by WebsiteSetup.org

# Question



# Question



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

- A `<div><p>I <strong>love HTML !`
- B `<div><p>I <strong>love</strong> HTML !</p>`
- C `<div><p>I <strong>love<strong> HTML !<p><div>`
- D `<div><p>I <strong>love</strong> HTML !</p></div>`
- E `<div><p>I </p><strong>love</strong><p> HTML !</p></div>`

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

- A `<div><p>I <strong>love HTML!`
- B `<div><p>I <strong>love</strong> HTML!</p>`
- C `<div><p>I <strong>love<strong> HTML!<p><div>`
- D `<div><p>I <strong>love</strong> HTML!</p></div>`
- E `<div><p>I </p><strong>love</strong><p> HTML!</p></div>`

A

0%

B

0%

C

0%

D

0%

E

0%

# Question



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

- A `<div><p>I <strong>love HTML !`
- B `<div><p>I <strong>love</strong> HTML !</p>`
- C `<div><p>I <strong>love<strong> HTML !<p><div>`
- D `<div><p>I <strong>love</strong> HTML !</p></div>`
- E `<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>
```

# Let's make a shopping list

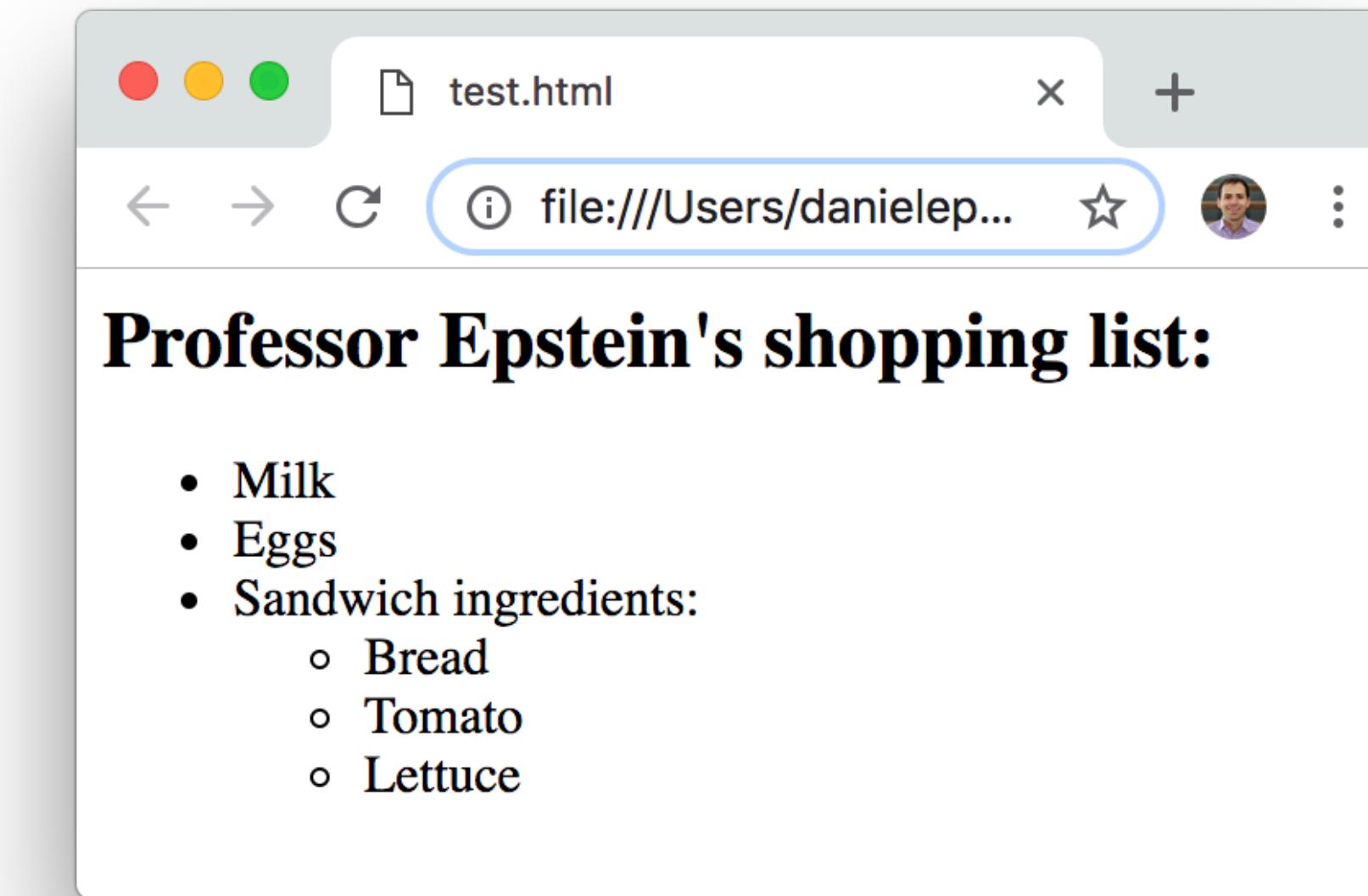
## Professor Epstein's shopping list

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



# Nesting: lists

```
<h2>Professor Epstein's shopping list:</h2>
<ul>
  <li>Milk</li>
  <li>Eggs</li>
  <li>Sandwich ingredients:
    <!--Lists can be nested!-->
    <ul>
      <li>Bread</li>
      <li>Tomato</li>
      <li>Lettuce</li>
    </ul>
  </li>
</ul>
```



# Nesting: HTML

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

```
<html> ← Start of HTML document  
<body> ← Start of body (visible) content  
  <h1>Hello, IN4MATX 133 !</h1>  
  <p>HTML is <em>great</em>!</p>  
</body> ← End of body content  
</html> ← 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-wi25.depstein.net/">IN4MATX 133</a>
```



anchor hypertext  
(hyperlink) reference

```

```



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

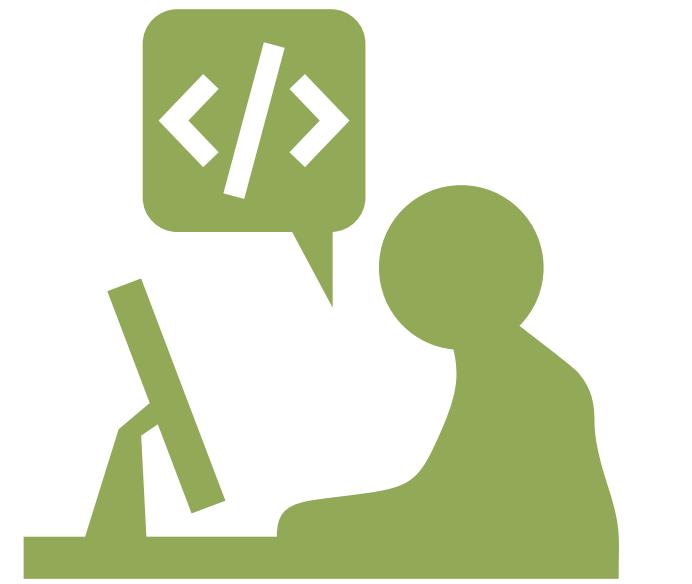
```
<!DOCTYPE html> ← Document format
<html lang="en"> ← Specify language
<head> ← Document header (content that's not shown)
    <meta charset="UTF-8"> ← Character set (for non-latin characters)
    <meta name="author" content="your name"> ← For search engines
    <title>My Webpage</title> ← Webpage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

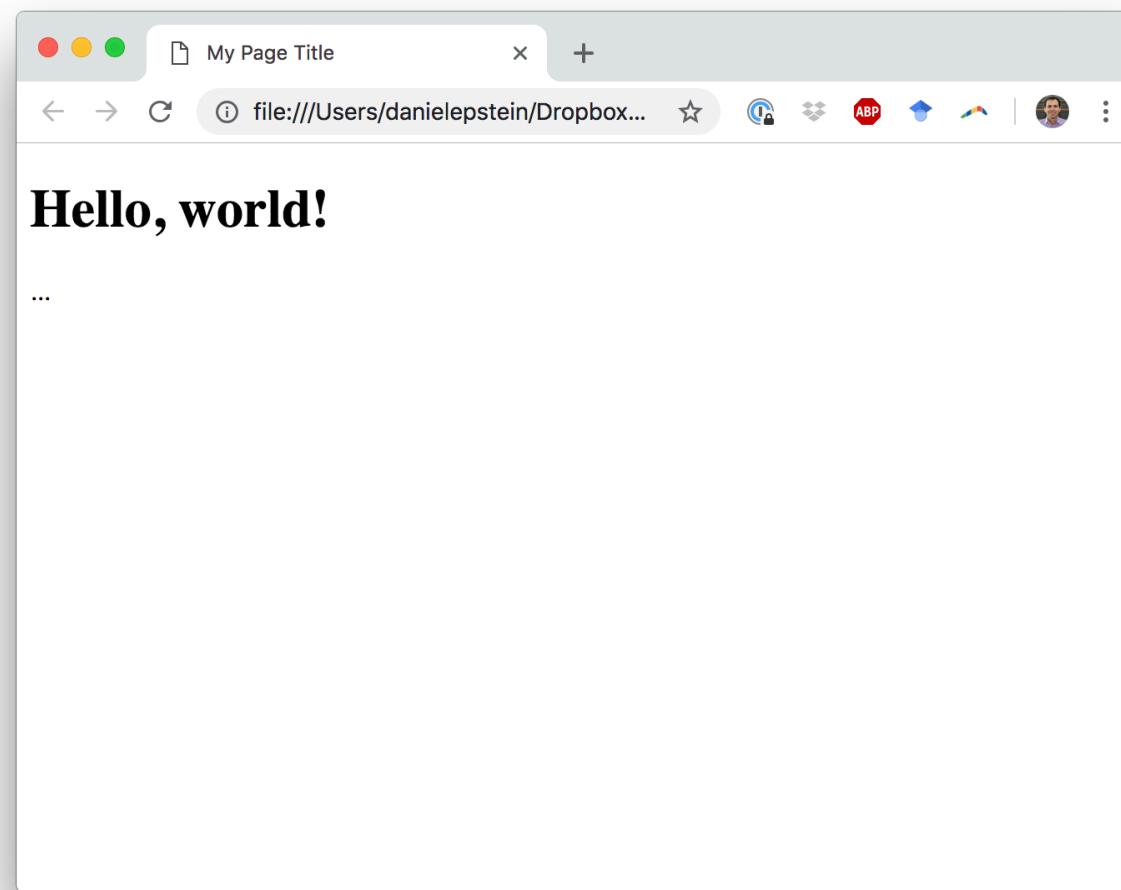
```
<h1>Hello, world!</h1>
```



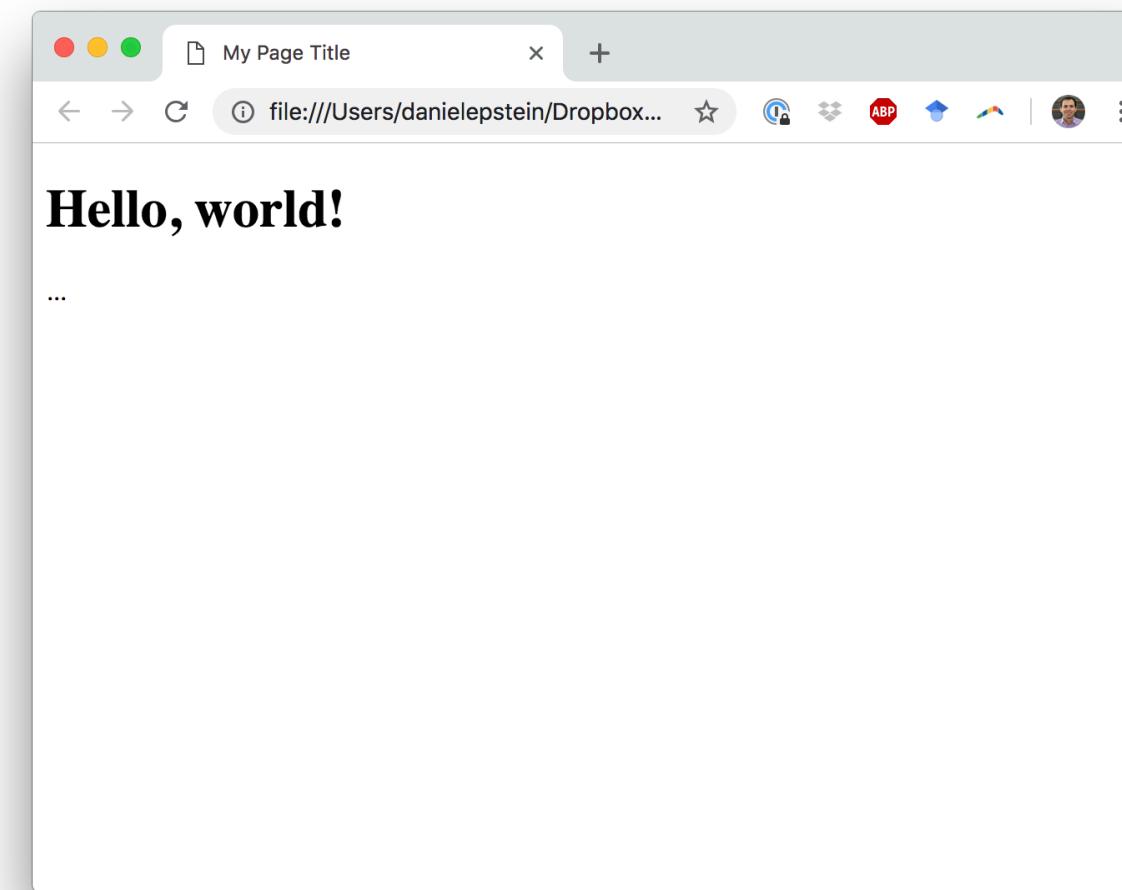
(live demo!)

# HTML structure

```
<!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>
```



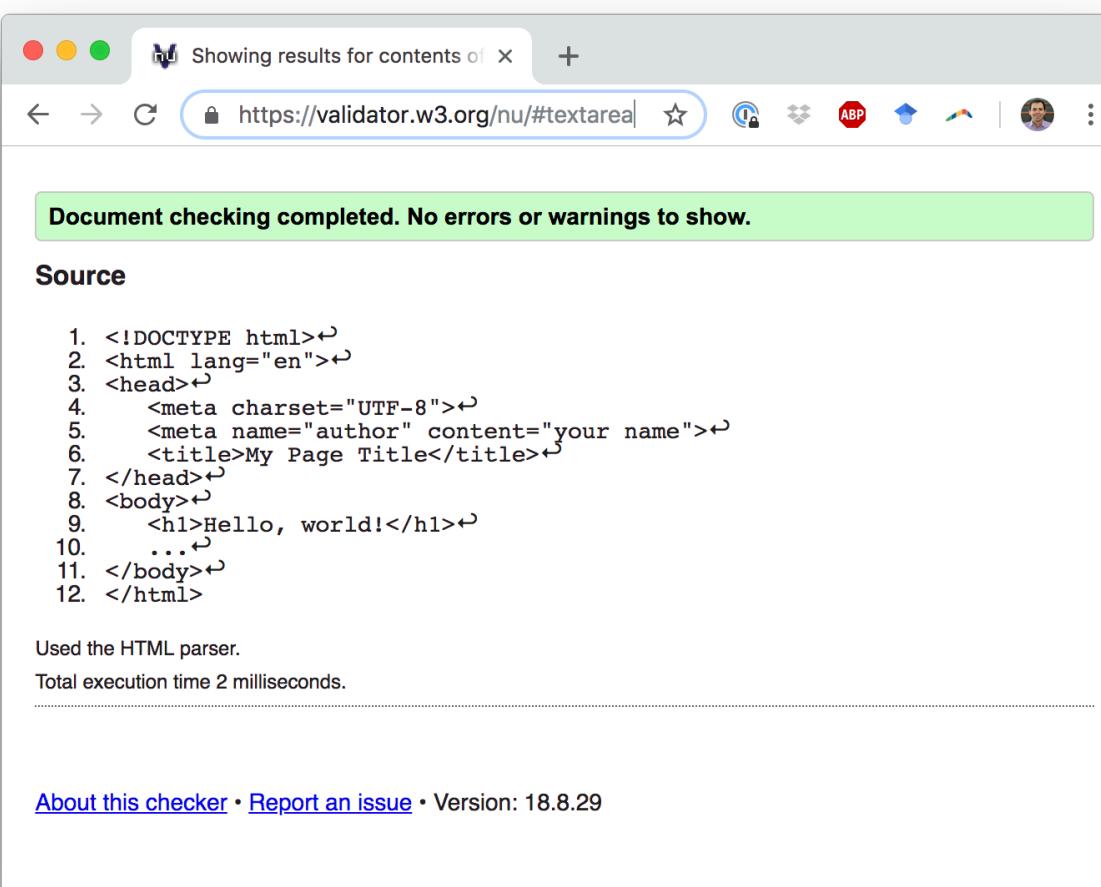
```
<html>
<head>
  <title>My Page Title</title>
</head>
<body>
  <h1>Hello, world!</h1>
  <p>...</p>
```



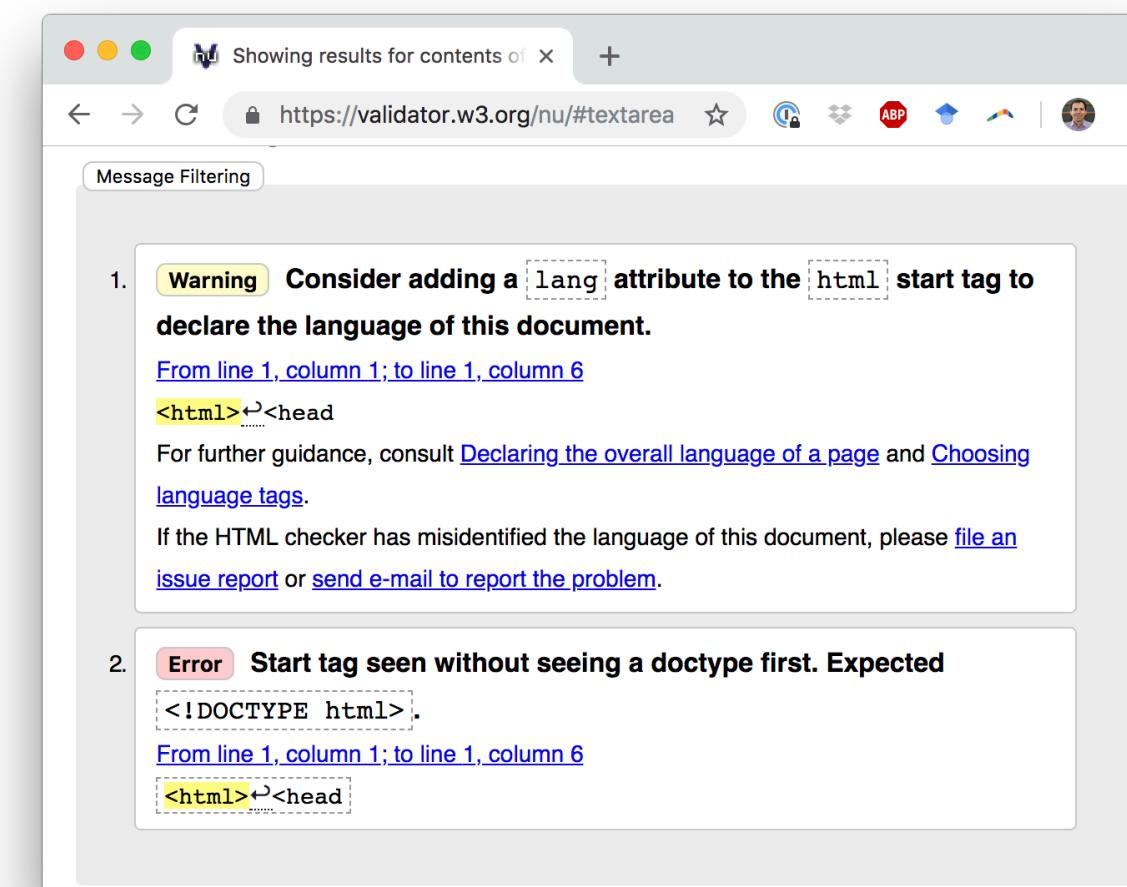
# W3C validator

<https://validator.w3.org/>

```
<!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>
```



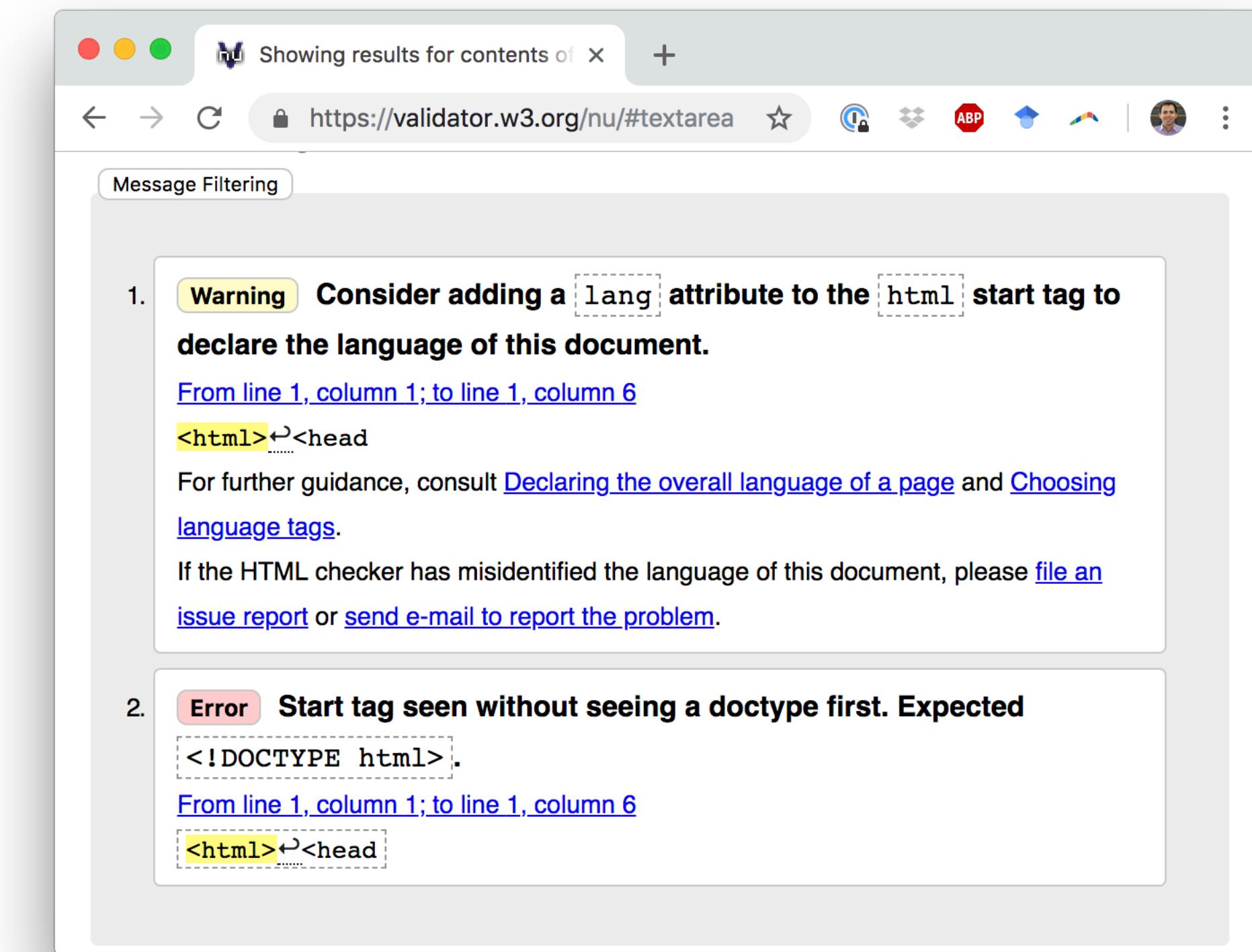
```
<html>
<head>
  <title>My Page Title</title>
</head>
<body>
  <h1>Hello, world!</h1>
  <p>...</p>
```



# W3C validator

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

- Even the validators are surprisingly accommodating!
- I count 5 structural errors that the validator did not report



# **Why does HTML structure matter?**

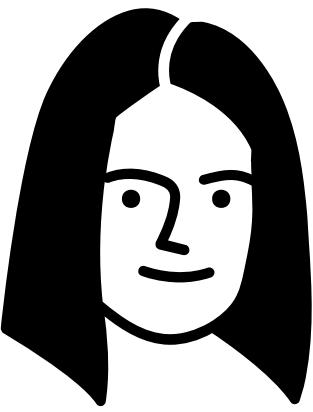
**Taking a step back:**  
**Web access is important**

**“The power of the Web is in its universality.  
Access by everyone regardless of disability is an essential aspect.”**

*-Tim Berners-Lee, inventor of the World Wide Web and 2016 Turing award winner  
<https://www.w3.org/WAI/fundamentals/accessibility-intro/>*

**All sorts of people  
will use the webpage you create**

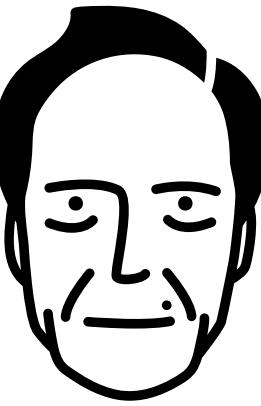
# Meet Tracy



Tracy Young is 28 years old and was born blind. She did well in school, getting support from audio tapes and books and the support of tutors. She never bothered really to learn Braille. She holds a college degree in English literature and is very fond of writing poems and short stories. When using her computer for work, she uses the JAWS software, which reads out aloud the content of the computer screen in an artificial voice (screen reader). JAWS runs only on Internet Explorer, which is the standard browser in Tracy's company.

- Adapted from [https://publikationen.sulb.uni-saarland.de/bitstream/20.500.11880/25641/1/personas\\_access.pdf](https://publikationen.sulb.uni-saarland.de/bitstream/20.500.11880/25641/1/personas_access.pdf)

# Meet Gerald



Gerald Oldman is 68 years old, a retired investment banker. He spends several hours a week on the Internet to manage his personal investments and pension funds. Gerald has some impairments which are quite common with senior citizens. His vision has reduced with age. The letters on the screen start to blur after reading for a while, so he needs an overhead light and a magnifying glass. His hands tend to be shaky, so that he has some difficulties making exact movements with a computer mouse. He therefore prefers keyboard controls.

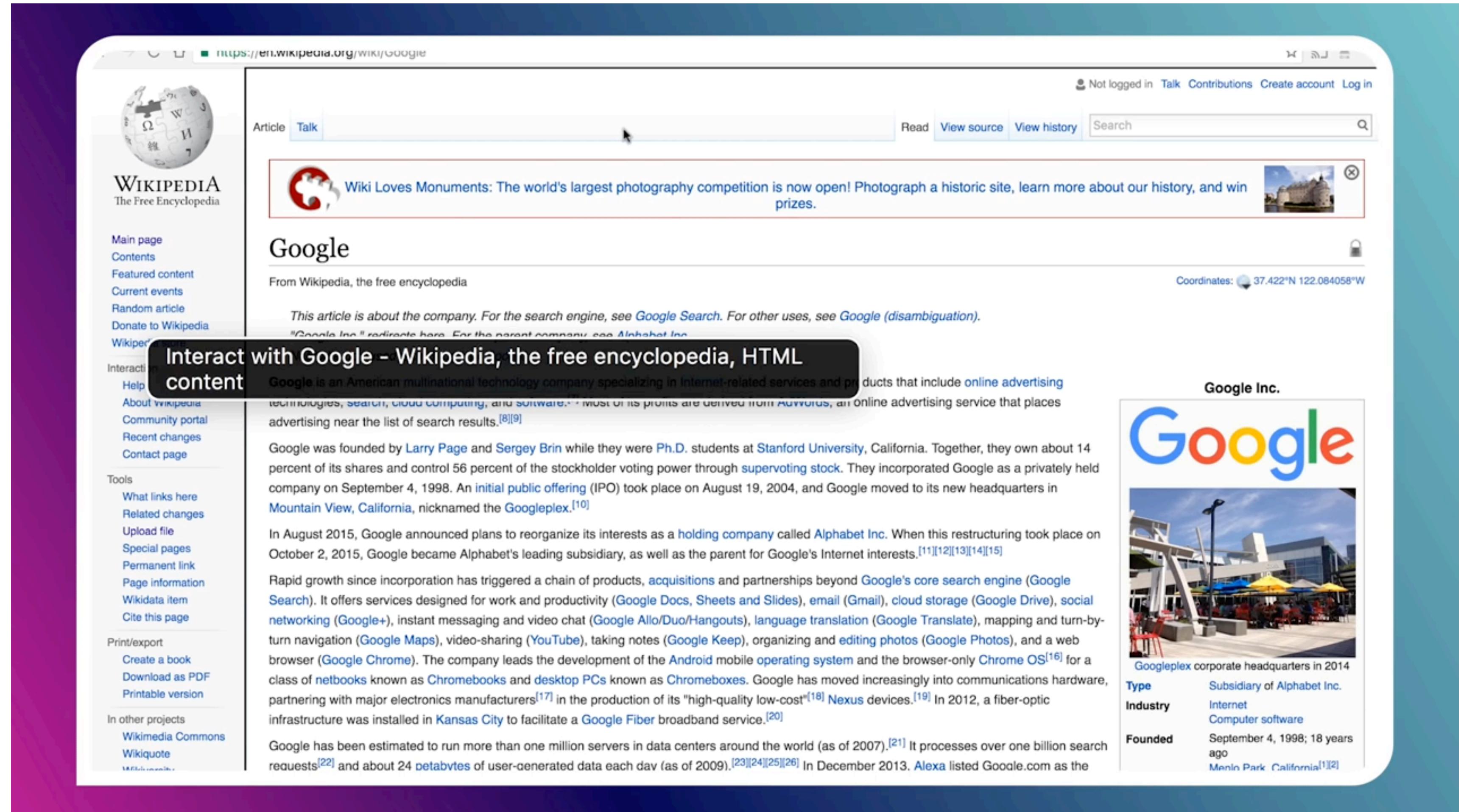
- Adapted from [https://publikationen.sulb.uni-saarland.de/bitstream/20.500.11880/25641/1/personas\\_access.pdf](https://publikationen.sulb.uni-saarland.de/bitstream/20.500.11880/25641/1/personas_access.pdf)

# Common impairments

- Vision
  - Blind, low vision, colorblind
- Motor impairments
  - Arthritis, cerebral palsy, tremors, paralysis
- Cognitive impairments
  - Autism, dyslexia, language barriers
- Much more

# One accommodation: Apple VoiceOver

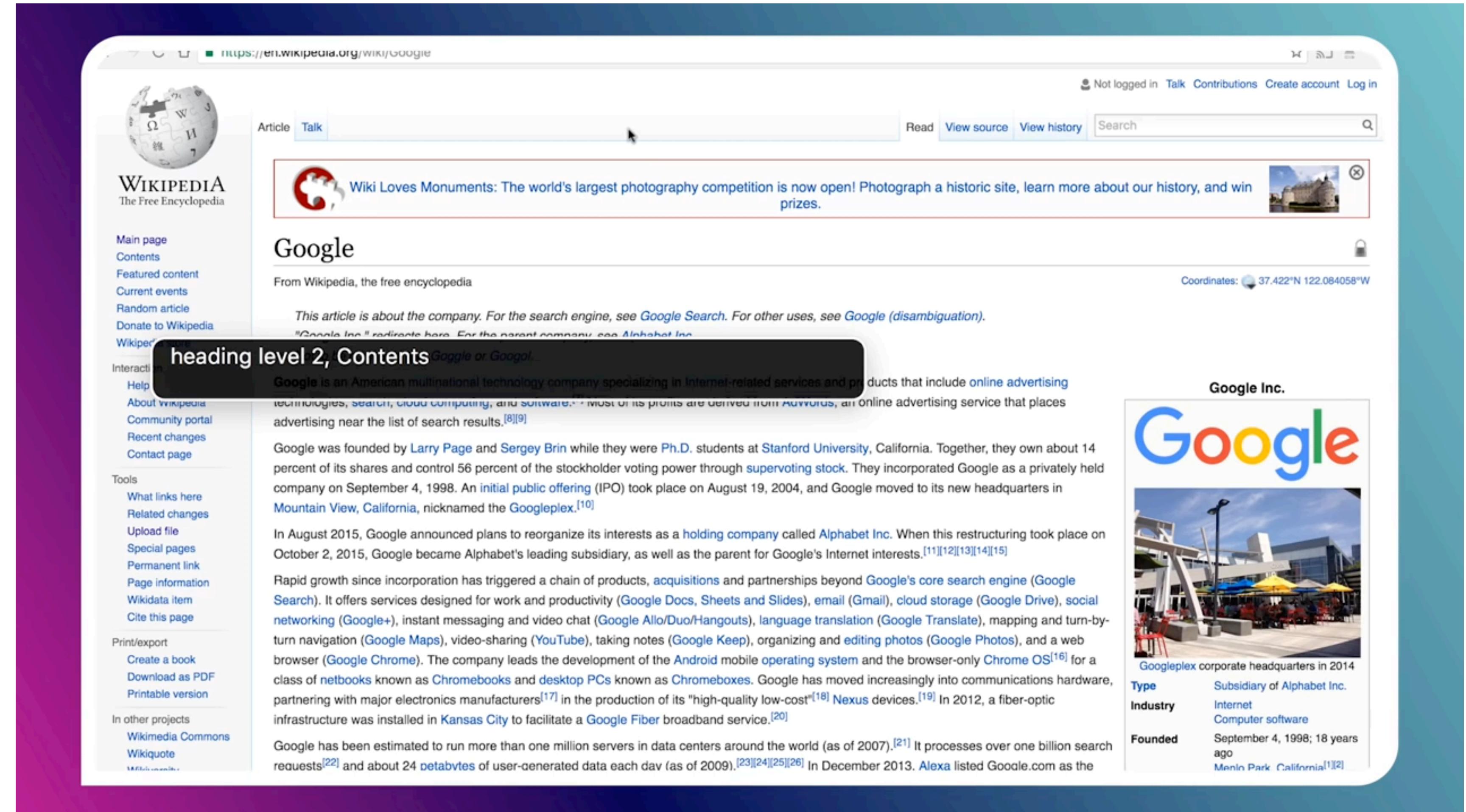
Enables navigating visual content via the keyboard



<https://youtu.be/5R-6WvAihms>

# One accommodation: Apple VoiceOver

Navigating via headings



<https://youtu.be/5R-6WvAihms>

# One accommodation: Apple VoiceOver

Navigating  
via specific types  
of content



<https://youtu.be/5R-6WvAihms>

# **How do we support easy navigation with a screen reader?**

**How do we support easy navigation  
with a screen reader?**

**Add semantic meaning to tags**

# Semantic (landmark) elements

## ARIA roles—the “old” way

- Give non-semantic elements (like <div>s) a role attribute to provide semantic meaning

```
<div role="main">
```

```
<div role="navigation">
```

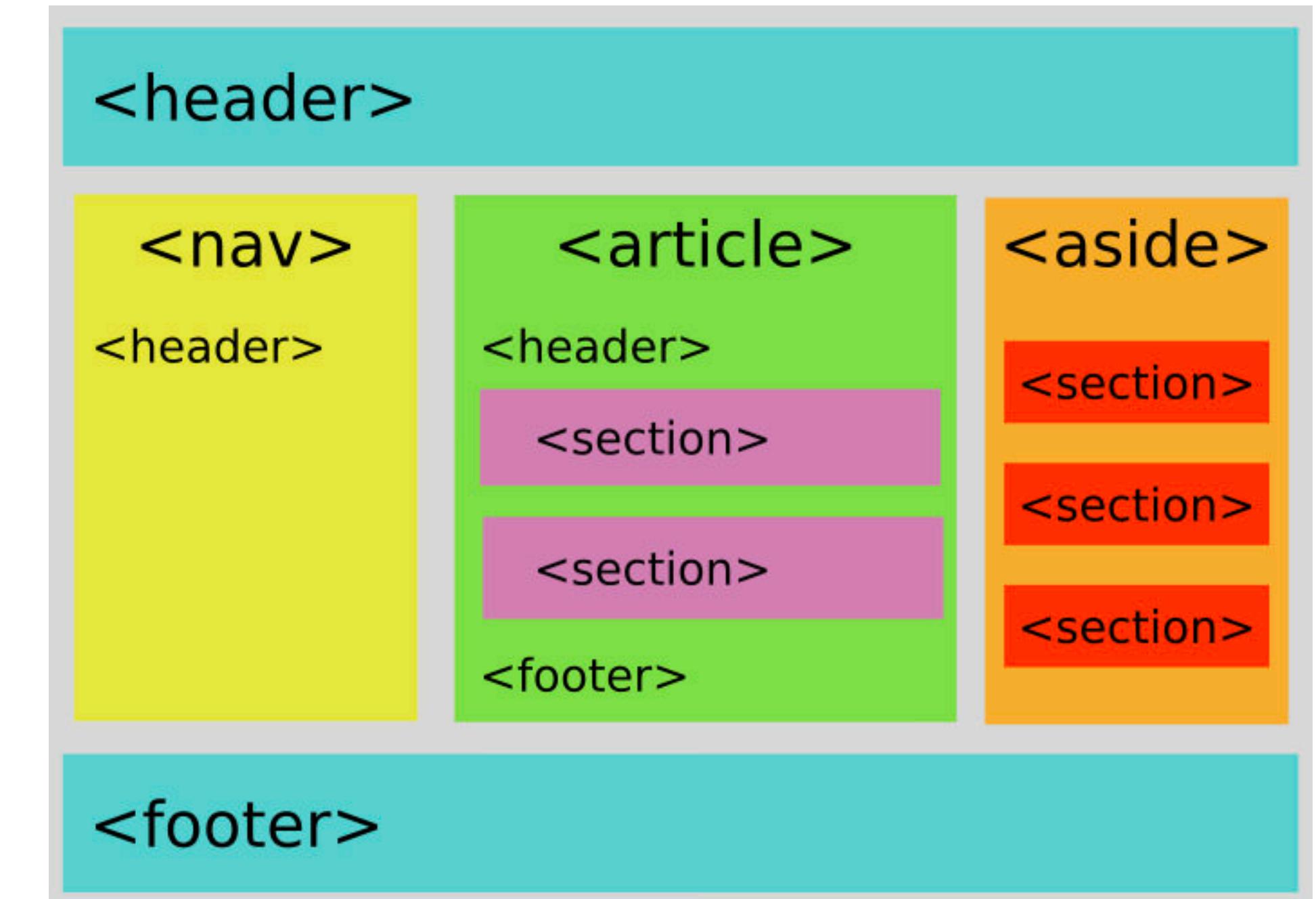
```
<div role="form">
```

- <https://www.w3.org/WAI/ARIA/apg/>

# Semantic (landmark) elements

## HTML5 tags—the “new” way

- Dedicated semantic tags
- [https://www.w3schools.com/html/html5\\_semantic\\_elements.asp](https://www.w3schools.com/html/html5_semantic_elements.asp)



# A few other accessibility examples

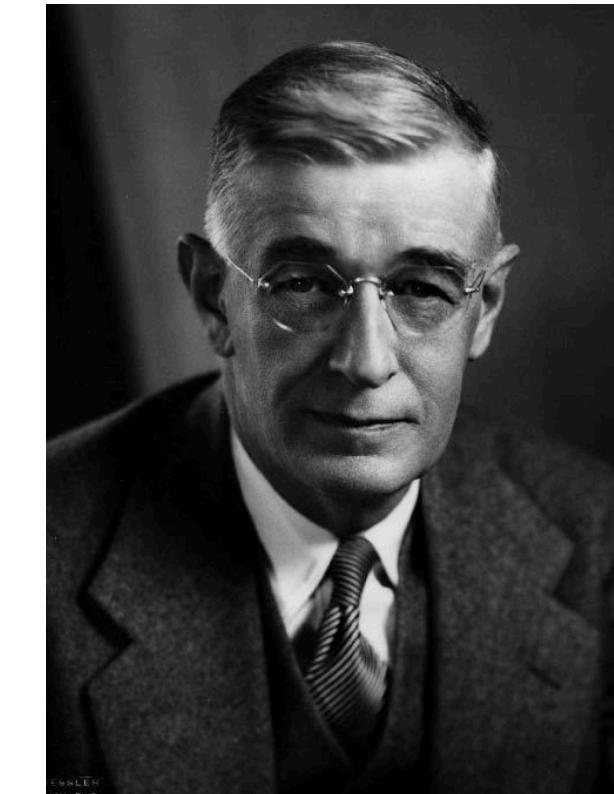
- “alt” attributes in images
- “aria-label” attributes to describe non-visual elements (like buttons)

```
<button aria-label="Close">x</button>
```

# Question



Which alt text would best describe this image?



- A ``
- B ``
- C ``
- D ``
- E ``

0%

Which alt text would best describe this image?



- A ``
- B ``
- C ``
- D ``
- E ``

A

0%

B

0%

C

0%

D

0%

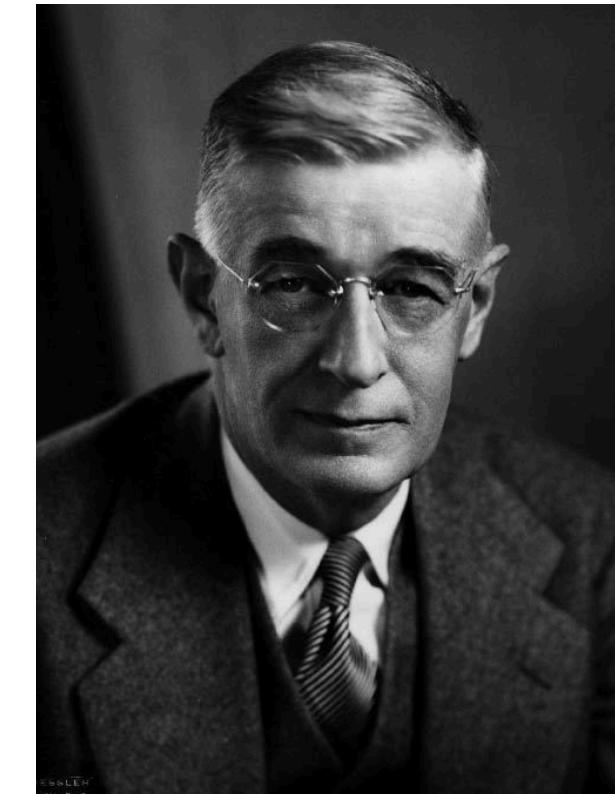
E

# Question



Which alt text would best describe this image?

(Depends on the context, really)

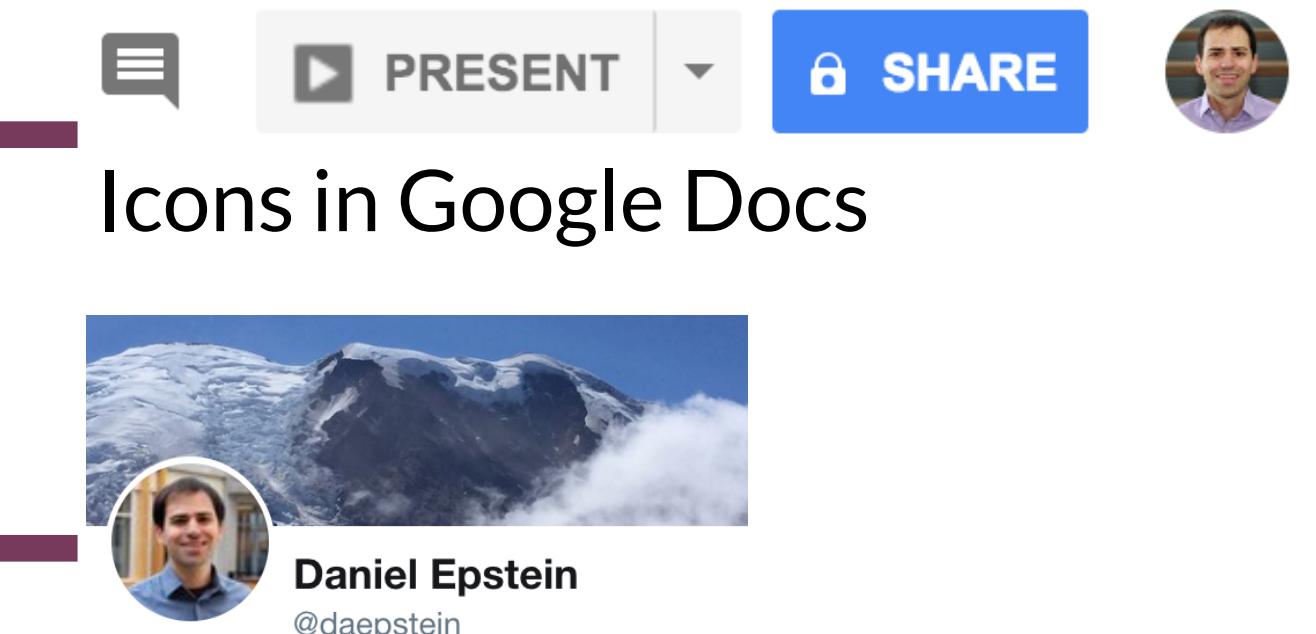


- A ``
- B ``
- C ``
- D ``
- E ``

# Alt text guidelines

1. Always include an alt attribute, even if it's empty
2. Describe the information, not the picture
3. “Active” images and images which contain information require descriptive alt text
4. Decorative images should have empty alt text
5. Be succinct, avoid being redundant with text

- <https://webaim.org/techniques/alttext/>
- <https://abilitynet.org.uk/news-blogs/five-golden-rules-compliant-alt-text>



Daniel Epstein  
@daepstein

Cover photos on Twitter/Facebook

# Accessibility validators

- WAVE <http://wave.webaim.org/>
- ACheker <https://achecks.org/achecker/>
- Both over-report problems, requires you to think through whether something is actually an accessibility issue
- Can try on your own with a screen reader
  - VoiceOver (Mac, under Settings -> Accessibility)
  - NVDA (Windows, requires download from <https://www.nvaccess.org/>)

**Wrap-up:**  
**Inclusive design**  
**is better for everyone**

# Inclusive design is better for everyone

- The HTML stands alone
  - Developers can glance at a page's source and have a good idea of what it renders
- Semantic HTML helps people identify the content they want
  - Accessibility benefits, as previously discussed
  - Interfaces can selectively remove or de-emphasize contextually unimportant content (e.g., footnotes on a small screen)
  - Search engines can index the important content (e.g., headings, articles) rather than UI content (e.g., nav, footers)

# A1

## Responsive portfolio in HTML and CSS

- Three requirements: content, responsiveness, validity
- *Content*: contains basic HTML and CSS (today & Tuesday)
- *Responsiveness*: adapts to a range of screen dimensions (next Thursday)
- *Validity*: must be well-formed HTML/CSS and pass accessibility guidelines
- Due at 11:59pm on Wednesday, January 22nd

# A1

## Responsive portfolio in HTML and CSS

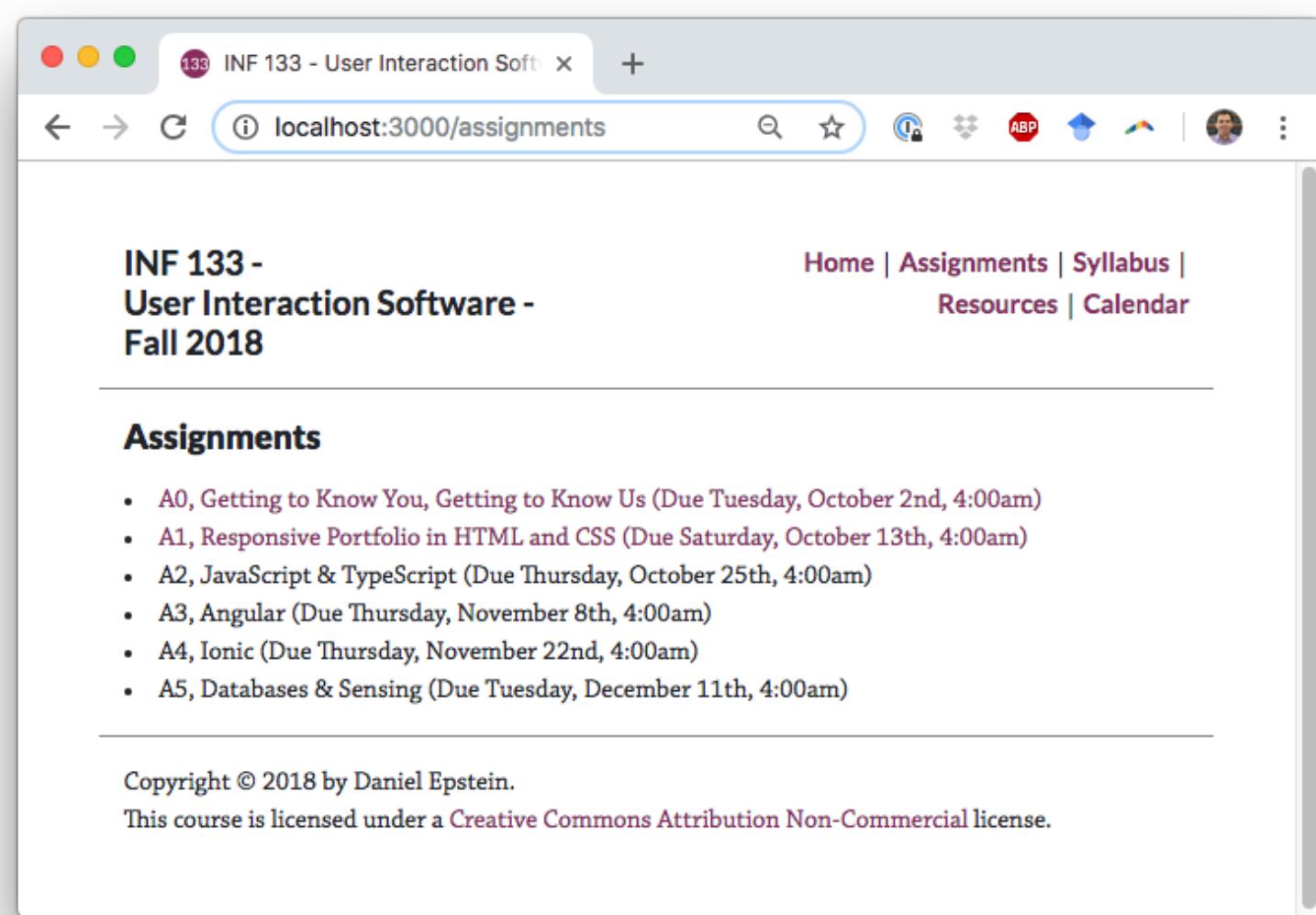
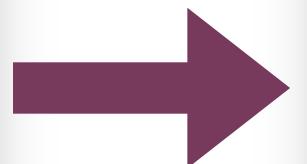
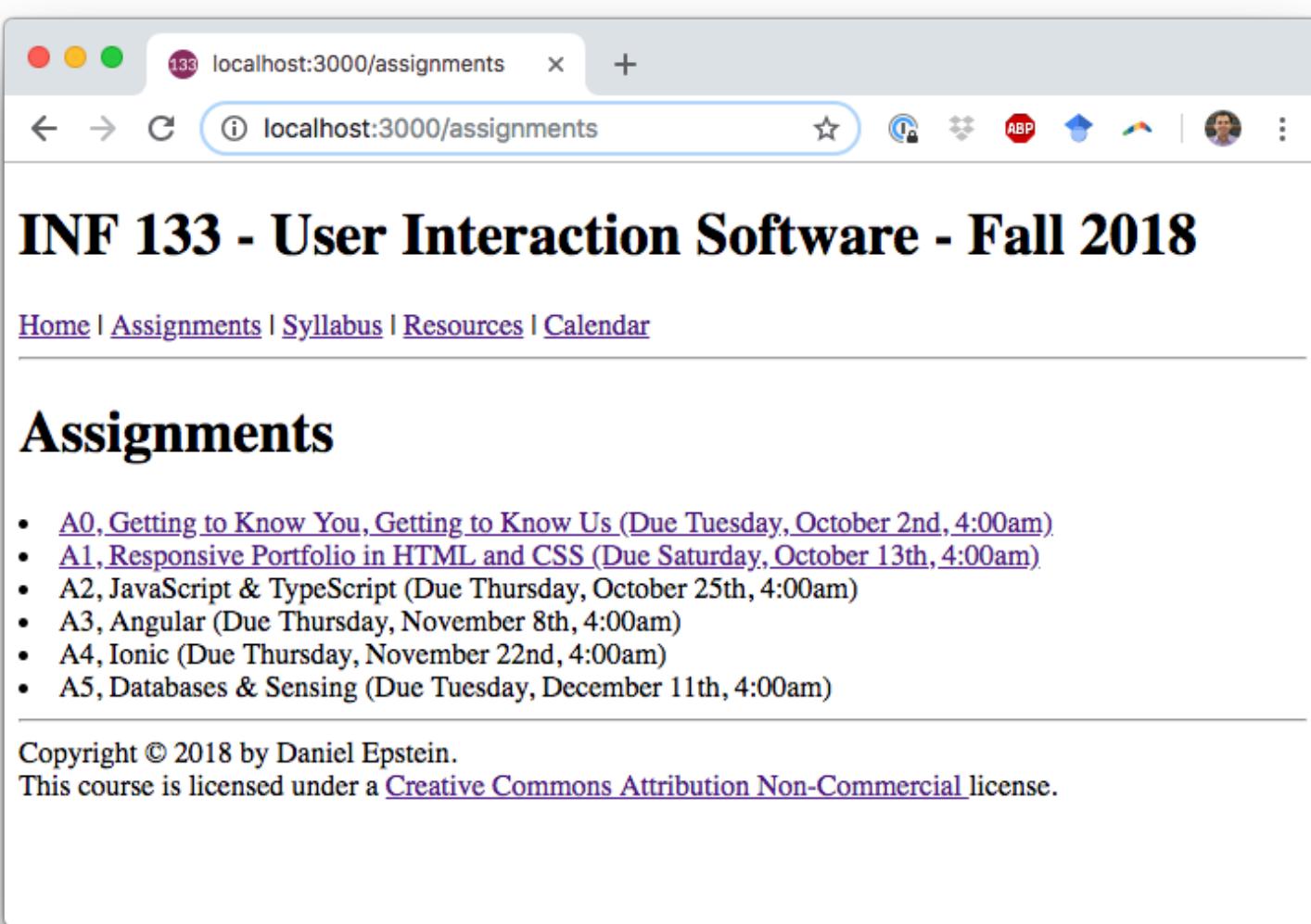
- This assignment is intentionally open-ended
- Many students use their A1 as the foundation for the portfolios they use when applying for jobs
- If you have already made a personal portfolio, you may re-use it for A1
  - But it must satisfy all of the other requirements
  - Consider adding a feature or two for the assignment

# Today's goals

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

- 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
- Explain the importance of accessible and semantically meaningful markup
- Generate markup which meets accessibility standards

# Next lecture: styling pages with CSS



# **IN4MATX 133: User Interface Software**

Lecture 2:  
HTML & Accessibility