



Introduction to HTML, CSS, and SVG

CS 4460

Feb 5, 2019

Computer

1

Human



About US

For years people have been trying to preserve the rarest of pepes from being stolen.

Thanks to Bitcoin and Counterparty we can associate these Rare Pepes with tokens to make them truly rare. Now Pepe's can preserve their rareness and actually be traded on the market.



Our experts at the Rare Pepe Directory certify Pepes rare and give that info to Pepe wallets such as Barepepwallet.com

When someone sends you a Pene token and it appears in one of these wallets, you can take comfort in the fact that it is indeed rare.

A sophisticated text editor for code, markup and prose

[DOWNLOAD FOR MAC](#)

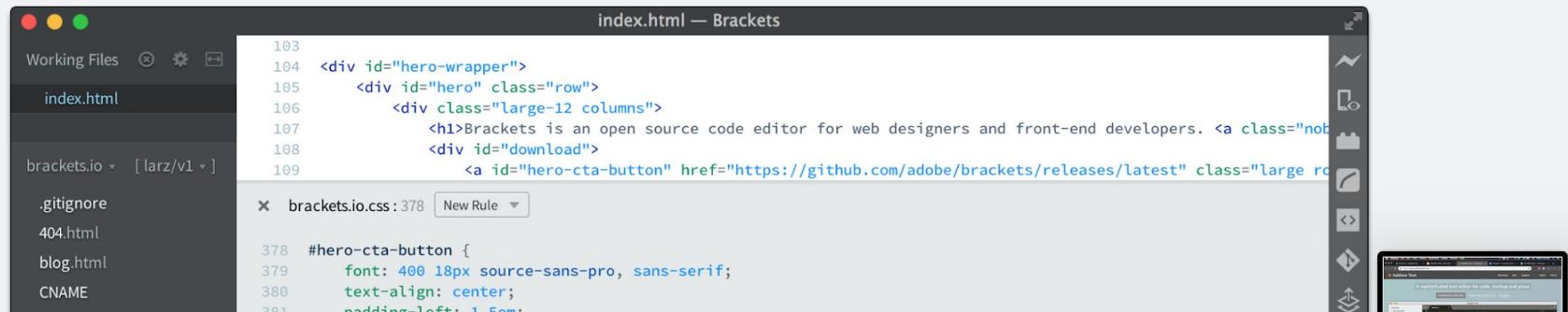
Sublime Text 3 (Build 3176)

[Changelog](#)

```
base64.cc

34
35 void base64_encode(const uint8_t * data, size_t length, char * dst,
36                     base64_charset variant)
37 {
38     const char * charset = (variant == base64_charset::URL_SAFE)
39             ? URL_SAFE_CHARSET
40             : STANDARD_CHARSET;
41
42     size_t src_idx = 0;
43     size_t dst_idx = 0;
44     for (; (src_idx + 2) < length; src_idx += 3, dst_idx += 4)
45     {
46         uint8_t s0 = data[src_idx];
47         uint8_t s1 = data[src_idx + 1];
48         uint8_t s2 = data[src_idx + 2];
49
50         dst[dst_idx + 0] = charset[(s0 & 0xfc) >> 2];
51         dst[dst_idx + 1] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];
52         dst[dst_idx + 2] = charset[((s1 & 0x0f) << 2) | (s2 & 0xc0) >> 6];
53         dst[dst_idx + 3] = charset[(s2 & 0x3f)];
54     }
55
56     if (src_idx < length)
57     {
58         uint8_t s0 = data[src_idx];
59
60         dst[dst_idx + 0] = charset[(s0 & 0xfc) >> 2];
61         dst[dst_idx + 1] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];
62         dst[dst_idx + 2] = charset[((s1 & 0x0f) << 2) | (s2 & 0xc0) >> 6];
63         dst[dst_idx + 3] = charset[(s2 & 0x3f)];
64     }
65 }
```

A modern, open source text editor that understands web design.

[Download Brackets 1.13](#)[Other Downloads](#)

<http://brackets.io/>

Teletype for Atom

Great things happen when developers work together—from teaching and sharing knowledge to building better software.

Teletype for Atom makes collaborating on code just as easy as it is to code alone, right from your editor.

Share your workspace and edit code together in real time. To start collaborating, open [Teletype in Atom](#) and install the package.

[Learn more](#)

The screenshot shows the Atom code editor interface. On the left, there's a tree view of a project named "real-time". Inside "real-time", there are folders for ".git", "lib", "node_modules", "script", "styles", "test", and files like "buffer-binding.js", "editor-binding.js", etc. A file named "real-time-package.js" is selected in the tree view. On the right, a large code editor window displays the contents of "real-time-package.js". The code is a JavaScript class definition for "RealTimePackage". The URL "https://atom.io/" is visible at the bottom right of the slide.

```
Project
real-time
  .git
  lib
    buffer-binding.js
    editor-binding.js
    guest-portal-binding.js
    join-portal-dialog.js
    normalize-uri.js
    real-time-package.js
  node_modules
  script
  styles
  test
    .gitignore
    .travis.yml
    index.js
    package-lock.json
    package.json
  README.md

lib/real-time-package.js

JS real-time-package.js

1 const {CompositeDisposable} = require('atom')
2 const {allowUnsafeNewFunction} = require('allow-unsafe-new-function')
3
4 let Client
5 allowUnsafeNewFunction(() => {
6   Client =
7     const BufferBinding = require('./buffer-binding')
8     const EditorBinding = require('./editor-binding')
9
10 module.exports =
11   class RealTimePackage {
12     constructor (options) {
13       cons|
```

<https://atom.io/>

Version 1.30 is now available! Read about the new features and fixes from November.

Code editing. Redefined.

Free. Open source. Runs everywhere.

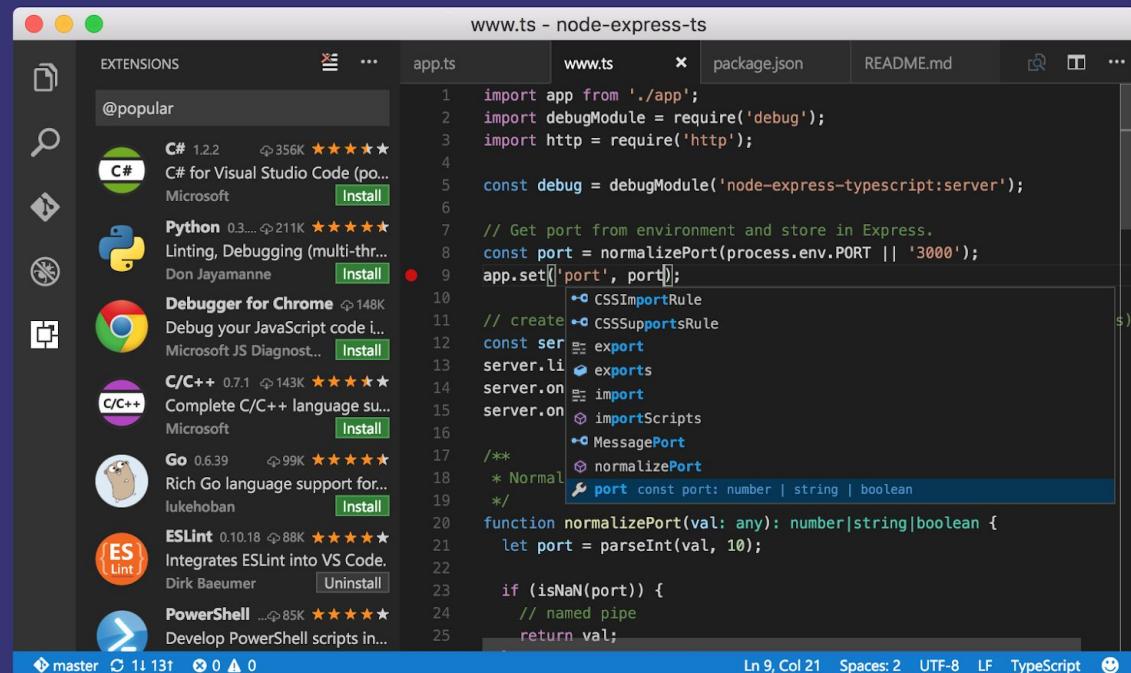
[Download for Mac](#)

Stable Build



[Other platforms and Insiders Edition](#)

By using VS Code, you agree to its
license and privacy statement.



The screenshot shows the Visual Studio Code interface. On the left, the Extensions sidebar is open, displaying popular extensions like C#, Python, and Debugger for Chrome. The main workspace shows a file named 'www.ts' with code related to node-express-ts. The status bar at the bottom provides information about the current file ('www.ts - node-express-ts'), the commit ('master'), and the editor settings ('Ln 9, Col 21 Spaces: 2 UTF-8 LF TypeScript').

```
import app from './app';
import debugModule = require('debug');
import http = require('http');

const debug = debugModule('node-express-typescript:server');

// Get port from environment and store in Express.
const port = normalizePort(process.env.PORT || '3000');
app.set('port', port);

// create
const server = exports;
server.listen(port, () => {
  console.log(`Server running on port ${port}`);
});

// exports
server.on('error', (err) => {
  if (err.syscall === 'listen') {
    let bind = `Port ${port}`;
    if (err.address) {
      bind = `Address ${err.address} port ${port}`;
    }
    console.error(`Error: ${bind} ${err.message}`);
    process.exit(1);
  } else {
    throw err;
  }
});

// imports
server.on('SIGINT', () => {
  process.exit();
});

// normalizePort
function normalizePort(val: any): number|string|boolean {
  let port = parseInt(val, 10);
  if (isNaN(port)) {
    // named pipe
    return val;
  }
  if (port < 0) {
    throw new Error(`Port ${port} must be a positive integer`);
  }
  return port;
}

// MessagePort
server.on('MessagePort', (port) => {
  const portObj = {
    port: port,
    onmessage: (event) => {
      const message = event.data;
      if (message.type === 'normalizePort') {
        message.value = normalizePort(message.value);
      }
      port.postMessage(message);
    },
  };
  portObj.port.addEventListener('close', () => {
    portObj.port = null;
  });
  portObj.port.addEventListener('error', (err) => {
    if (portObj.port) {
      portObj.port.close();
    }
    portObj.port = null;
  });
  portObj.port.addEventListener('message', (event) => {
    const message = event.data;
    if (message.type === 'normalizePort') {
      message.value = normalizePort(message.value);
    }
    portObj.port.postMessage(message);
  });
});
```

<https://code.visualstudio.com/>



What is HTML?

- HTML = HyperText Markup Language
- Standardized by the World Wide Web Consortium (W3C)
 - Computer Scientists love their acronyms
 - Look up: Tim Berners-Lee, Robert Cailliau, Dan Connolly...
- Elements that build many of the web pages that we see
- Syntax: <**tagname**>Content</**tagname**>



Basic structure of an HTML page

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
</body>
</html>
```

← Define the file as an HTML document
← HTML element is the root element
← Metadata, general info about the doc
← Visible content to the humans

```
| <html>  
| |  
| |   <head>  
| |     Title, links to stylesheets, etc.  
| |   </head>  
| |  
| |  
| |  
| |
```

```
<body>  
  Visible content  
</body>
```

```
| </html>  
| |  
| |  
| |
```



Quick check #1

Which of the following element(s) are the bones of an HTML page?

- a. <!DOCTYPE html>
- b. Tail
- c. Head
- d. Body
- e. Memes



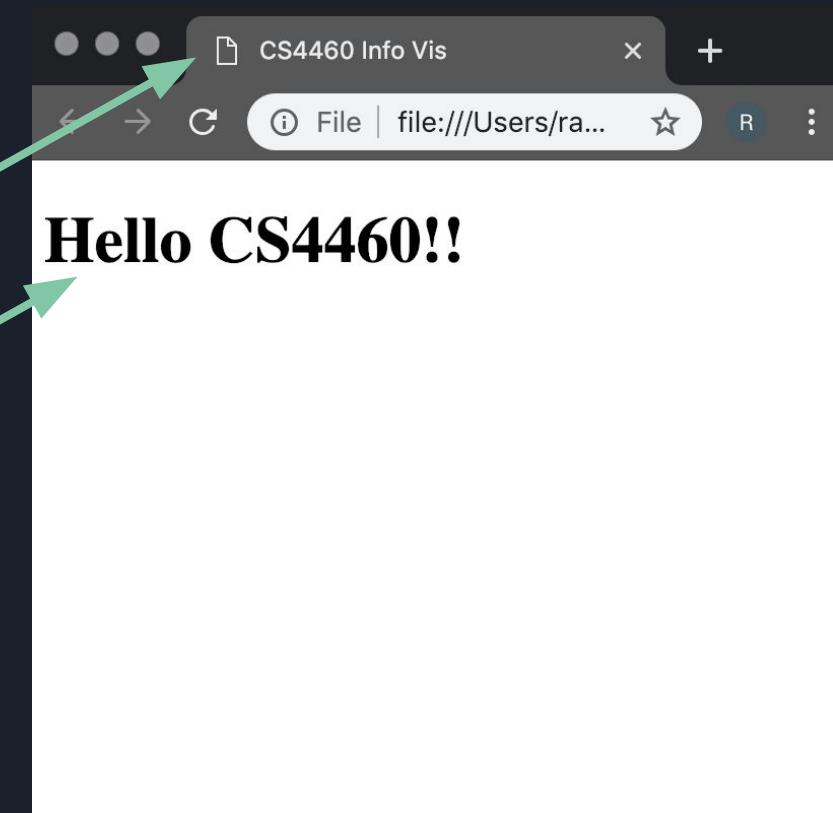
Quick check #1

Which of the following element(s) are the bones of an HTML page?

- a. **<!DOCTYPE html>**
- b. Tail
- c. Head
- d. Body
- e. Memes

```
<!DOCTYPE html>
<html>
<head>
    <title>CS4460 Info Vis</title>
</head>
<body>
    <h1>Hello CS4460!!</h1>
</body>
</html>
```

Text editor



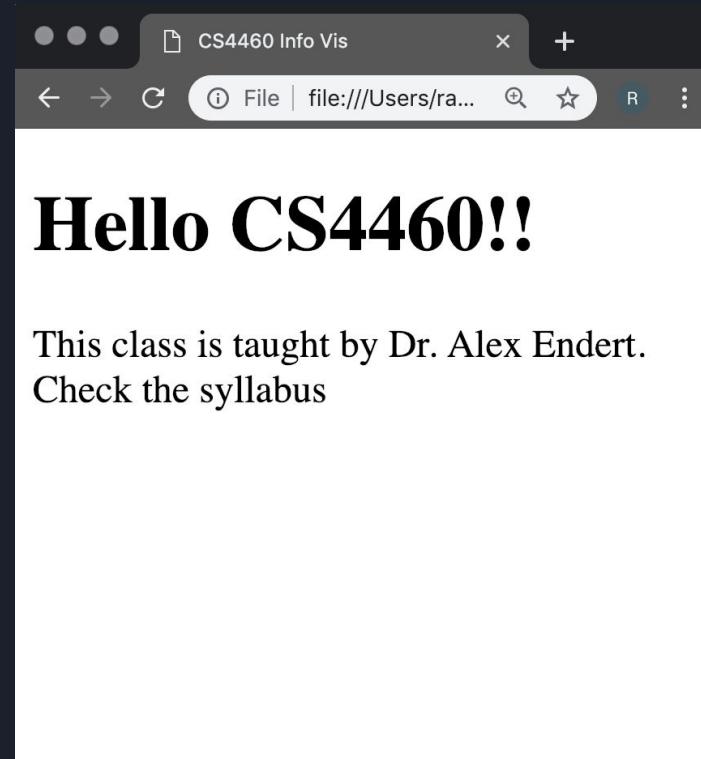
Browser



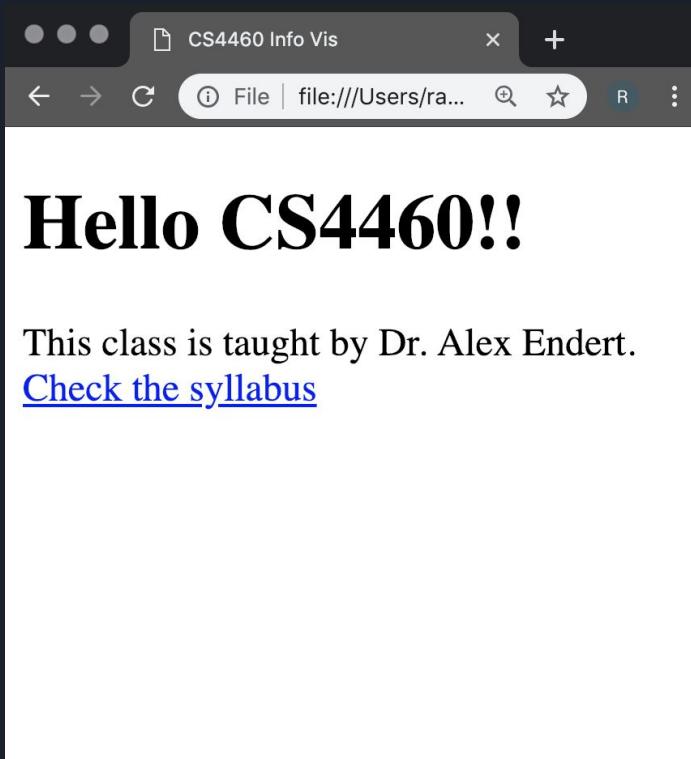
HTML attributes

- Attributes provide additional information about the tagged elements
- Syntax: **<tagname attribute="...">Content</tagname>**

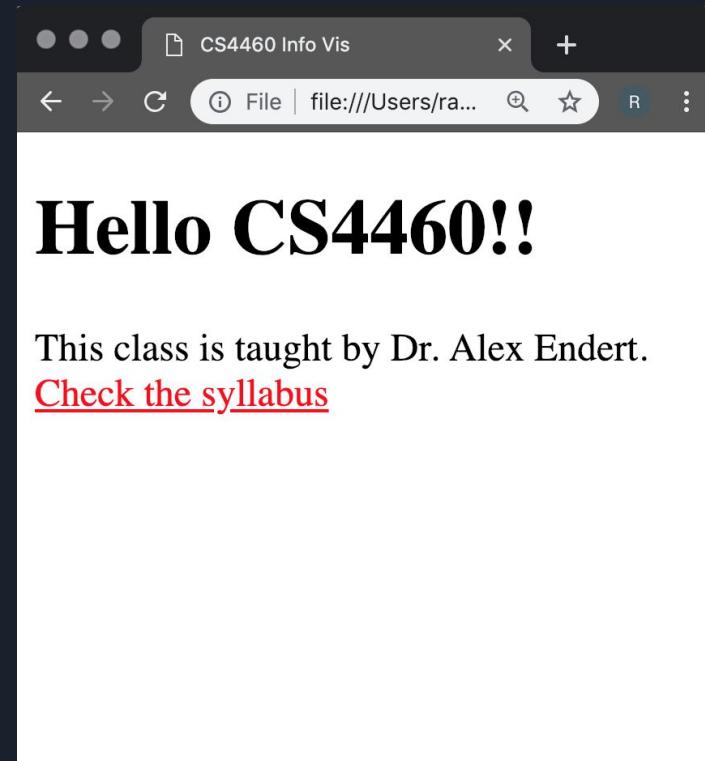
```
<!DOCTYPE html>
<html>
<head>
    <title>CS4460 Info Vis</title>
</head>
<body>
    <h1>Hello CS4460!!</h1>
    <p>This class is taught by Dr. Alex
    Endert. <a>Check the syllabus</a></p>
</body>
</html>
```



```
<!DOCTYPE html>
<html>
<head>
    <title>CS4460 Info Vis</title>
</head>
<body>
    <h1>Hello CS4460!!</h1>
    <p>This class is taught by Dr. Alex
    Endert. <a href =
    "http://va.gatech.edu/courses/cs4460/">C
    heck the syllabus</a></p>
</body>
</html>
```



```
<!DOCTYPE html>
<html>
<head>
    <title>CS4460 Info Vis</title>
</head>
<body>
    <h1>Hello CS4460!!</h1>
    <p>This class is taught by Dr. Alex
    Endert. <a href =
    "http://va.gatech.edu/courses/cs4460/">
    style="color:red;">Check the
    syllabus</a></p>
</body>
</html>
```





Try “Kuala Lumpur”

Saved

Trips

Messages

Help



Explore Airbnb



Homes



Experiences



Restaurants

More ways to stay in 2019



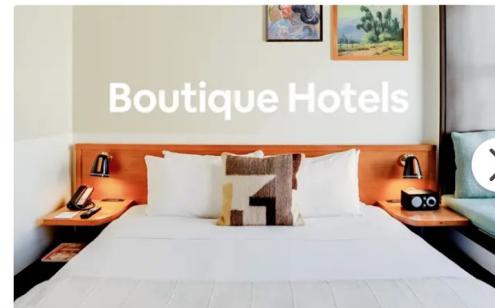
ENTIRE HOMES

Everything you need including a full range of amenities, with room for the group or whole family.



AIRBNB PLUS HOMES

A new selection of homes verified for quality & comfort.



BOUTIQUE HOTELS

Attentive service and comfort with local flavor.

Our community's favorite homes



Terms, Privacy, Currency & More



Try “Kuala Lumpur”

Saved

Trips

Messages

Help



Explore Airbnb



Homes



Experiences



Restaurants

More ways to stay in 2019



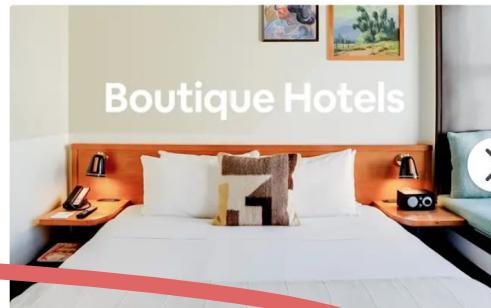
ENTIRE HOMES

Everything you need including a full range of amenities, with room for the group or whole family.



AIRBNB PLUS HOMES

A new selection of homes verified for quality & comfort.



BOUTIQUE HOTELS

BOUTIQUE HOTELS

Attentive service and comfort with local flavor.

Our community's favorite homes



Terms, Privacy, Currency & More

```
<h1 style="color: #31b569">ENTIRE HOMES</h1>
<span>Everything you need including a full range of amenities,
with room for the group or whole family.</span>

<h1 style="color: #31b569">AIRBNB PLUS HOMES</h1>
<span>A new selection of homes verified for quality &
comfort.</span>

<h1 style="color: #31b569">BOUTIQUE HOTELS</h1>
<span>Attentive service and comfort with local flavor.</span>
```

Text editor



What is CSS?

- CSS = Cascading Style Sheets
- Steps:
 1. Create a default.css file in the directory
 2. In <head> of index.html, insert a link to default.css
 3. In index.html, add a class attribute to the specific elements
 4. In default.css, specify the style

```
<p>  
    People in info vis you should know:<br>  
    Edward Tufte - data-ink ratio<br>  
    Ben Shneiderman - info vis mantra<br>  
    Jin Zhang - information retrieval  
</p>
```

The screenshot shows a presentation slide with the following content:

← → C ⓘ File | file:///Users/ra... ☆ R :

Hello CS4460!!

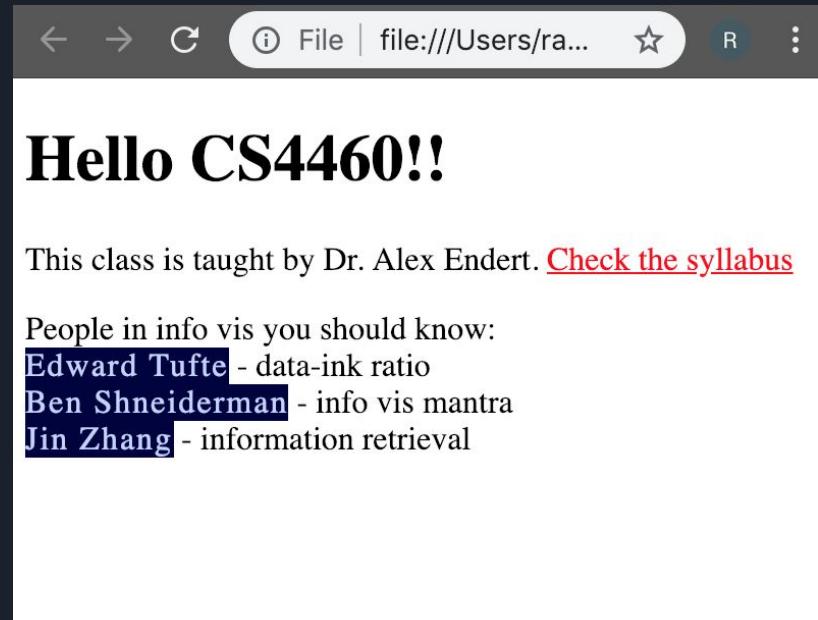
This class is taught by Dr. Alex Endert. [Check the syllabus](#)

People in info vis you should know:
Edward Tufte - data-ink ratio
Ben Shneiderman - info vis mantra
Jin Zhang - information retrieval

```
<p>  
    People in info vis you should know:<br>  
    <span class="key-names">Edward Tufte</span> - data-ink ratio<br>  
    <span class="key-names">Ben Shneiderman</span> - info vis mantra<br>  
    <span class="key-names">Jin Zhang</span> - information retrieval  
</p>
```

```
.key-names {  
    color: #c4d3ff;  
    background-color: #00003f;  
    letter-spacing: 1px;  
}
```

default.css



Browser



CSS selectors

Syntax	Example	What it does
.class	.key-names	Select all elements with class="key-names"
#id	#tufte	Select the element with id="tufte"
element	h1	Select all h1 elements
:hover	a:hover	Select links on mouse over

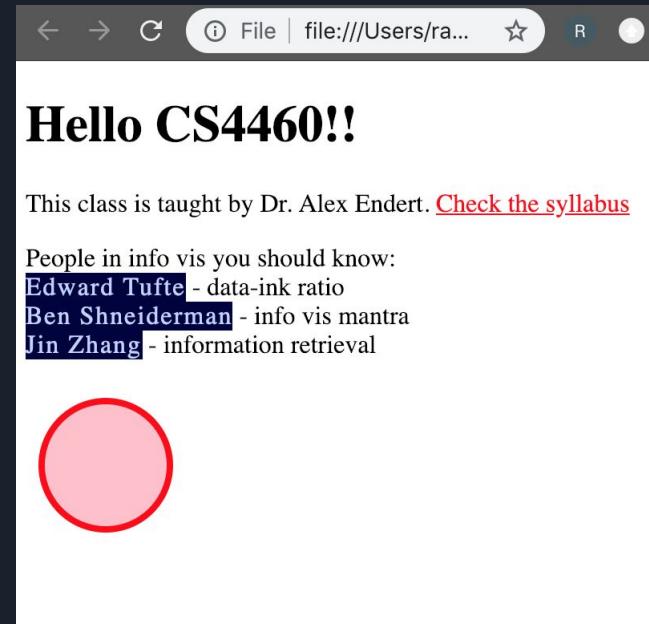


What is SVG?

- **SVG = Scalable Vector Graphics**
- Renders graphics with shapes, coordinates, etc.
- Useful for D3!

```
<svg width="100" height="100">  
  <circle cx="50" cy="50"  
    r="40" stroke="red"  
    stroke-width="4"  
    fill="#ffc0cb" />  
</svg>
```

index.html



Browser

```
<svg width="100" height="100">  
  
</svg>
```

index.html



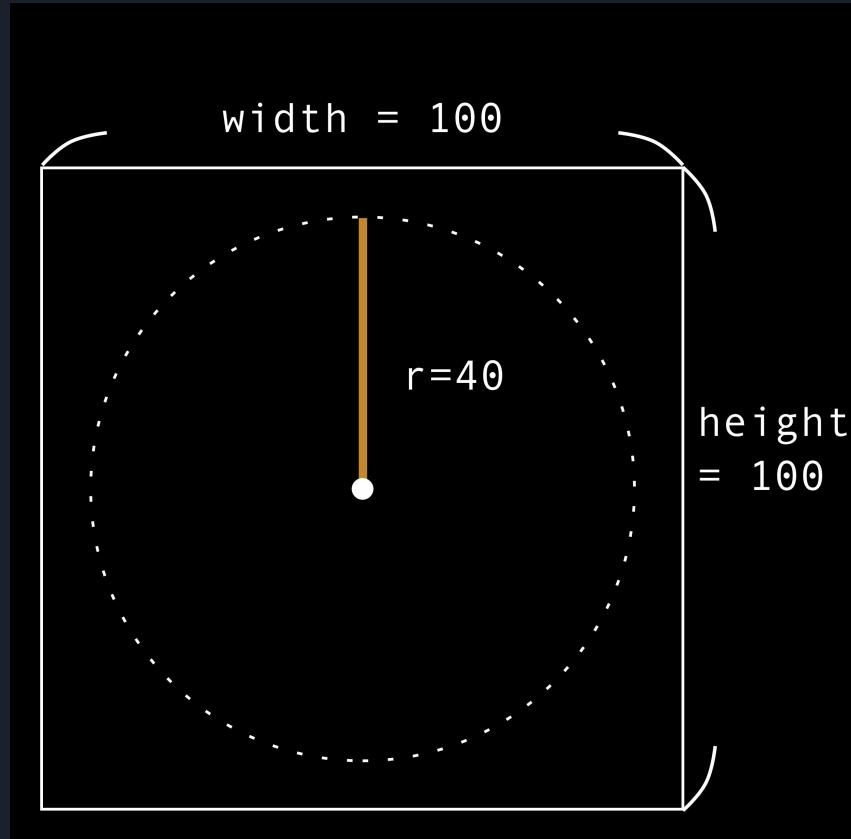
```
<svg width="100" height="100">  
  <circle cx="50" cy="50" />  
  
</svg>
```

index.html



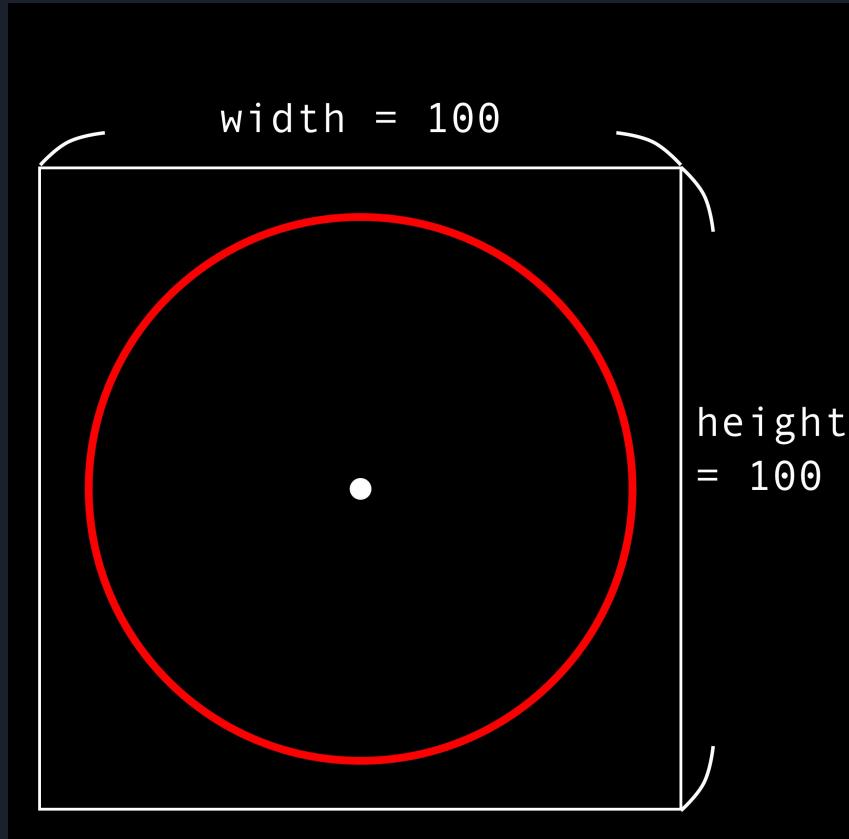
```
<svg width="100" height="100">  
  <circle cx="50" cy="50"  
    r="40" />  
  
</svg>
```

index.html



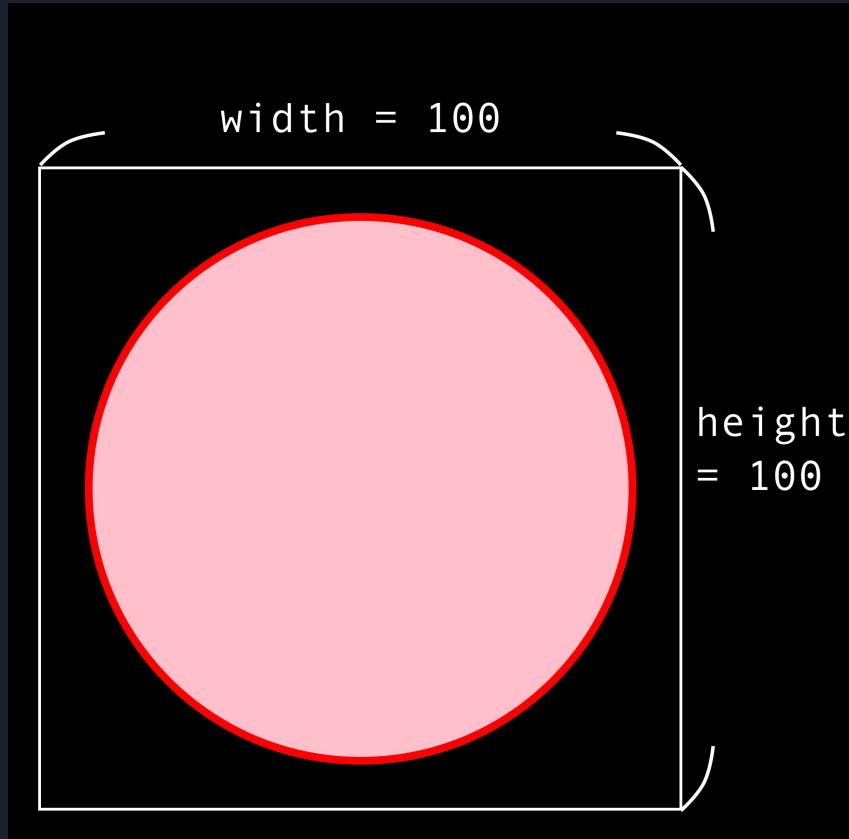
```
<svg width="100" height="100">  
  <circle cx="50" cy="50"  
    r="40" stroke="red"  
    stroke-width="4" />  
  
</svg>
```

index.html



```
<svg width="100" height="100">  
  <circle cx="50" cy="50"  
    r="40" stroke="red"  
    stroke-width="4"  
    fill="#ffc0cb" />  
</svg>
```

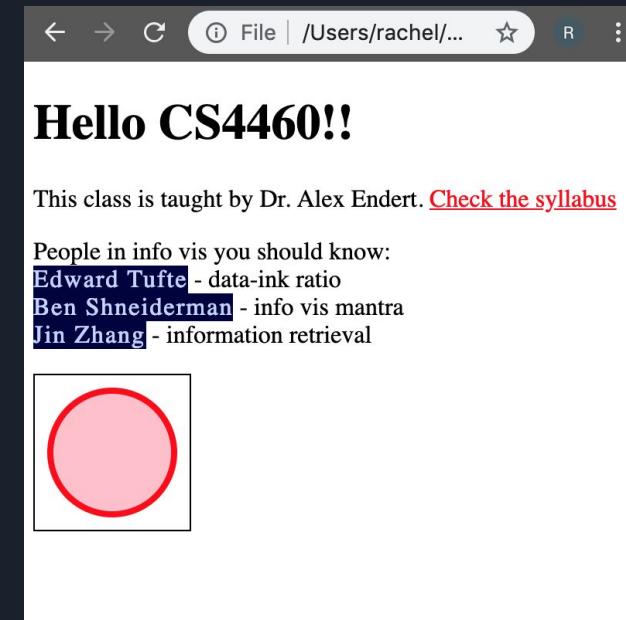
index.html



Trick: outline the border of the svg

```
<svg width="100" height="100"  
style="border: 1px solid black;">  
    <circle cx="50" cy="50"  
    r="40" stroke="red"  
    stroke-width="4"  
    fill="#ffc0cb" />  
</svg>
```

index.html

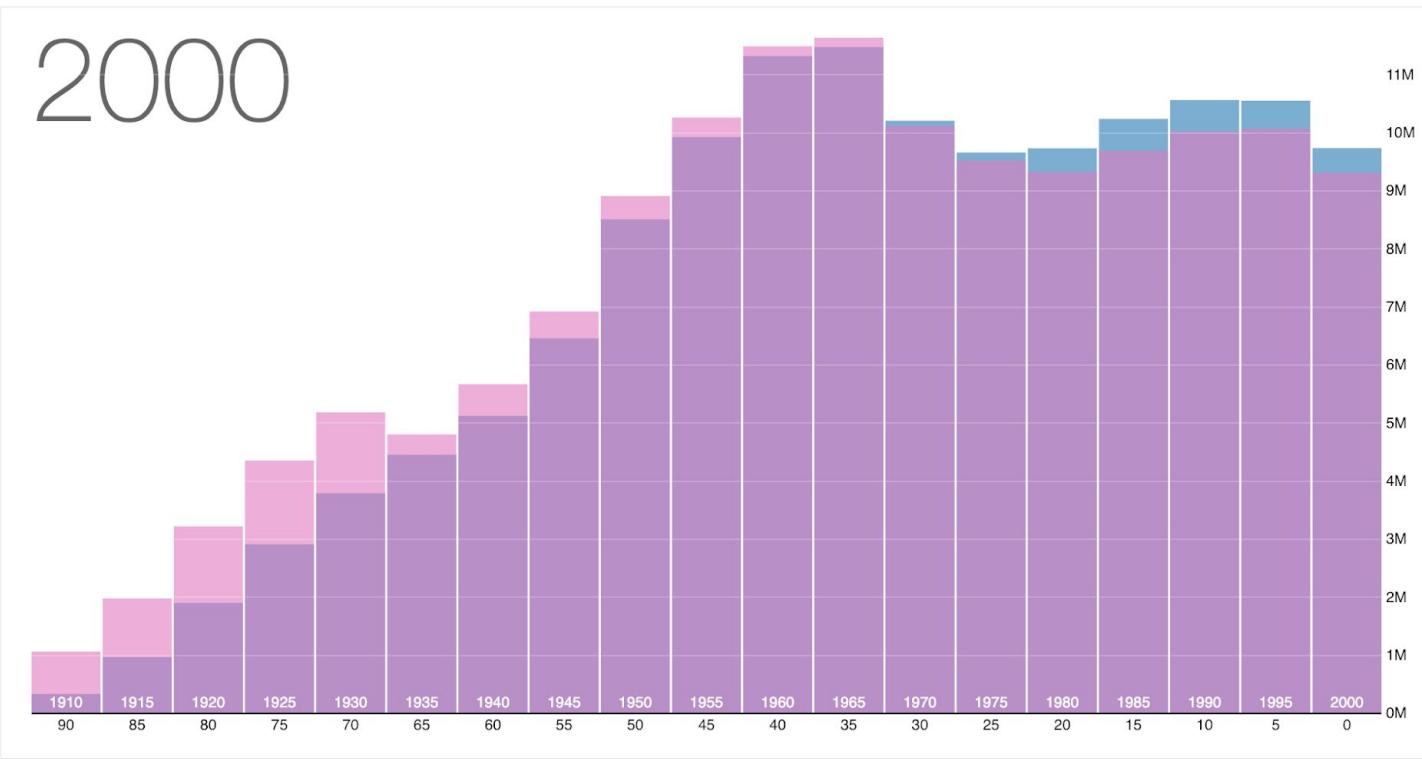


Browser



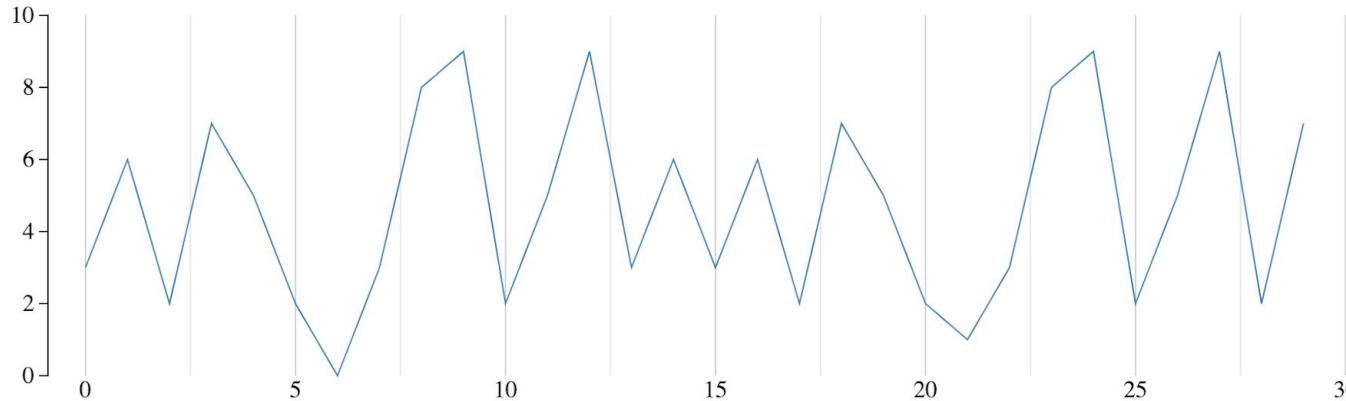
Population Pyramid

2000



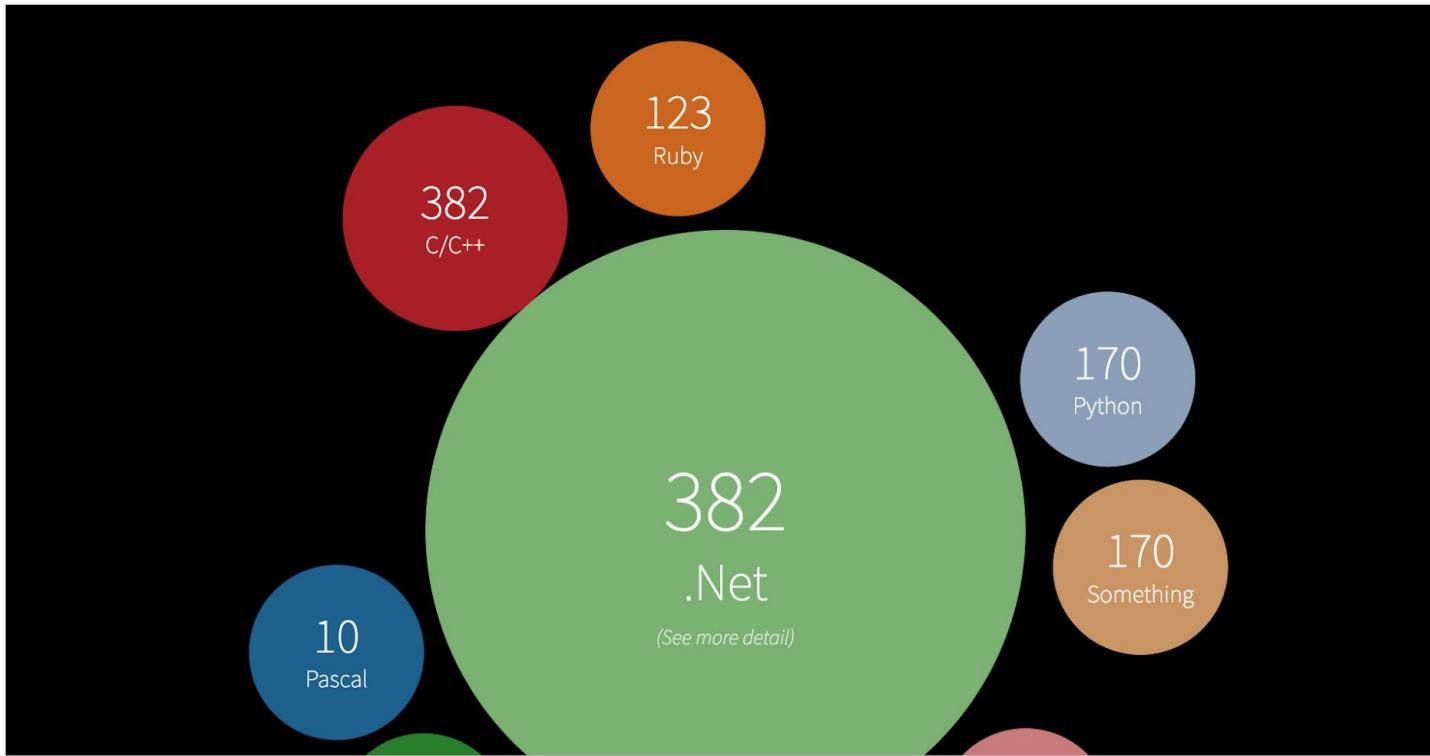


Simple Line Graph using SVG and d3.js





D3 Bubble chart





To learn more

https://www.w3schools.com/html/html_intro.asp

<https://developer.mozilla.org/en-US/docs/Web/HTML/Element>

https://www.w3schools.com/cssref/css_selectors.asp

<https://www.sitepoint.com/svg-101-what-is-svg/>

Feel free to email me at [rachelwchen@gatech.edu!](mailto:rachelwchen@gatech.edu)



Styling resources

<https://www.colorhexa.com/>

<https://webaim.org/resources/contrastchecker/>

<https://fonts.google.com/>