THE FRONTEND ENVIRONMENT

HTML/CSS/JAVASCRIPT

10 Minute HTML

OVERVIEW OF KEY HTML CONCEPTS

Every documents begins thus:
`<!DOCTYPE html>`

General syntax
`<tag>...</tag>`

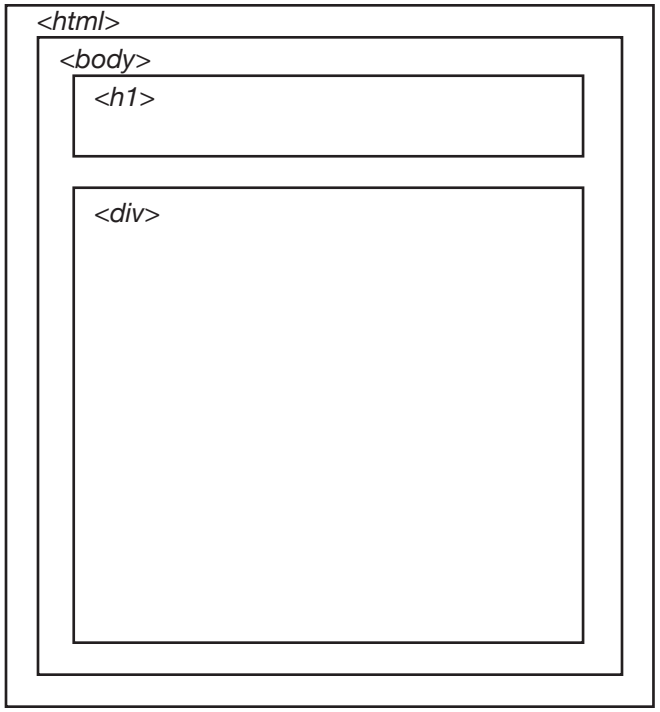
...where there is no content between tags
`<tag />`

Comments are ignored by browser for rendering
`<!-- ... -->`

Tags are nested to create hierarchy in the document

```
<!DOCTYPE html>
<html>
  <head>
  ...
  </head>
  <body>
    <h1>Hello World</h1>
     <div>...</div>
  </body>
</html>
```

OVERVIEW OF KEY HTML CONCEPTS

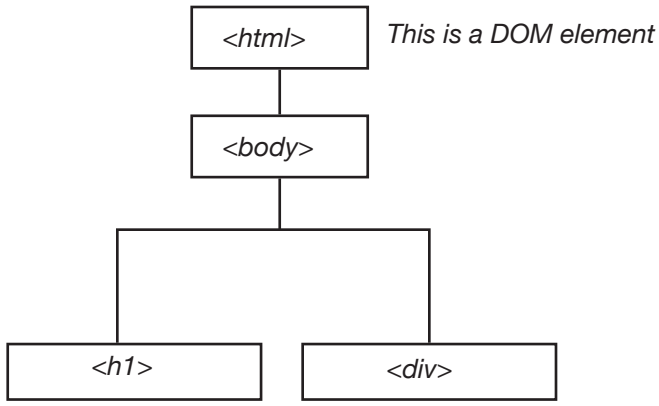


Tags are nested to create hierarchy in the document

```
<!DOCTYPE html>
<html>
  <head>
  ...
  </head>
  <body>
    <h1>Hello World</h1>
    <div>...</div>
  </body>
</html>
```

OVERVIEW OF KEY HTML CONCEPTS

The same document hierarchy can be visualized like this--commonly called the **DOM tree**:




```
<!DOCTYPE html>
<html>
  <head>
  ...
  </head>
  <body>
    <h1>Hello World</h1>
    <div>...</div>
  </body>
</html>
```

The DOM tree is made up of **DOM elements**.

OVERVIEW OF KEY HTML CONCEPTS

Tags can have attributes, class, and/or id

attribute *class*

```
<a href="http://www.github.com" class="button"
id="special"> Link to Github </a>
```

id

attribute Defines a key property for an element e.g. where does a link take you to

class Defines a group of elements with similar styles and/or semantic role

id Defines a specific element; only one allowed per document

HTML IN ACTION

- `<header class="mast-head" id="mast-head" role="banner">...</header>`

HTML IN ACTION

- `<li class="shortcuts">...</li>`
- `<h2 class="story-heading">...</h2>`

OVERVIEW OF KEY HTML CONCEPTS

Tags can have attributes, class, and/or id

attribute *class*

```
<a href="http://www.github.com" class="button"
id="special"> Link to Github </a>
```

id

Comprehensive reference here:
http://www.w3schools.com/tags/default.asp

LET'S RUN THROUGH SOME COMMON TAGS

`<body>`

`< >` Defines a hyperlink

`< >` Contains elements like `<script>` or `<link>`

`< >` Contains introductory content, such as navigation

`< >` Body paragraph text

`<ul>`

`<li>`

`< >` A grouping of elements; a section or division in the document

`< >` A grouoping of in-line elements

`<img>`

LET'S RUN THROUGH SOME COMMON TAGS

`<body>` Contains all the contents of the page

`<a>` Defines a hyperlink

`<head>` Contains elements like `<script>` or `<link>`

`<header>` Contains introductory content, such as navigation

`<p>` Body paragraph text

`<ul>` Unordered (bulleted) list

`<li>` Item in a list

`<div>` A grouping of elements; a section or division in the document

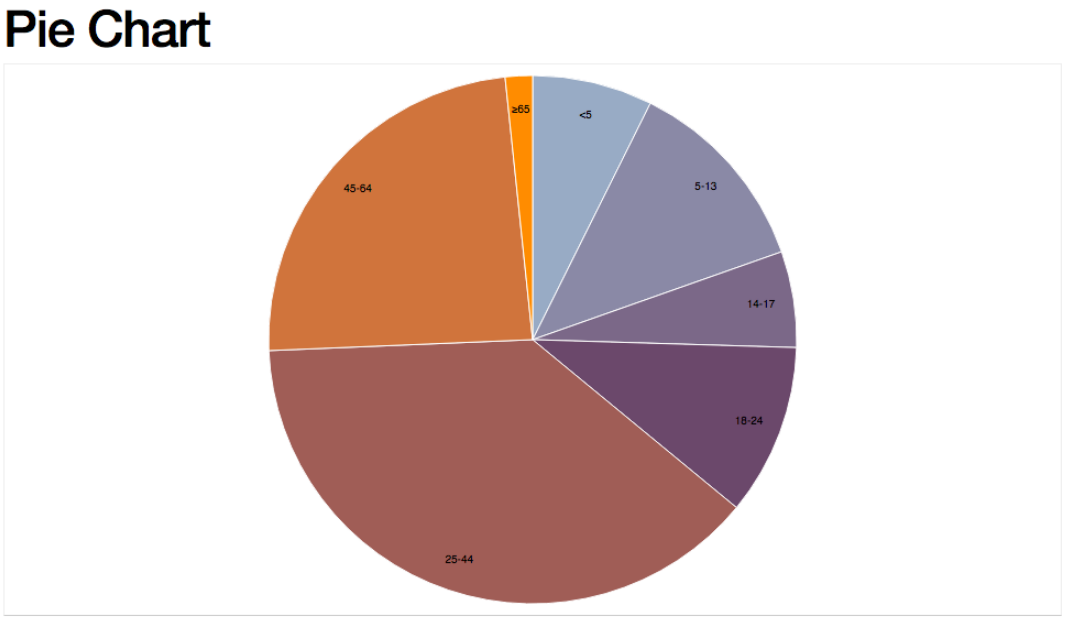
`<span>` A grouoping of in-line elements

`<img>` Image

In-class Exercise 1

Mark up a page yourself

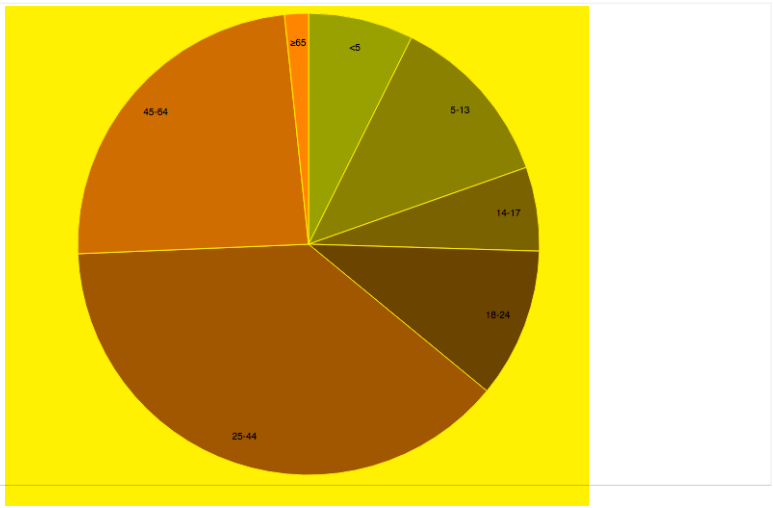
HOW IS THIS RELATED TO DATA VISUALIZATION?



http://bl.ocks.org/mbostock/3887235

HOW IS THIS RELATED TO DATA VISUALIZATION?

Pie Chart



In D3 visualization, data is represented by DOM elements.

```
<!DOCTYPE html>
<html>
  <head>
  ...
  </head>
  <body>
    <h1>Bubble Chart</h1>
    ...
    <div>
      <svg>
        ...
        <circle />
        <circle />
        ...
      </svg>
    </div>
  </body>
</html>
```

PRACTICAL CSS

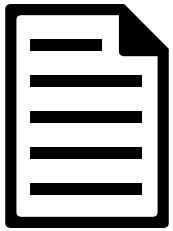
HOW IS EVERYTHING RELATED?

JavaScript

“Behavior”

All the dynamic stuff, such as animation, user interaction, manipulating DOM elements...

HTML



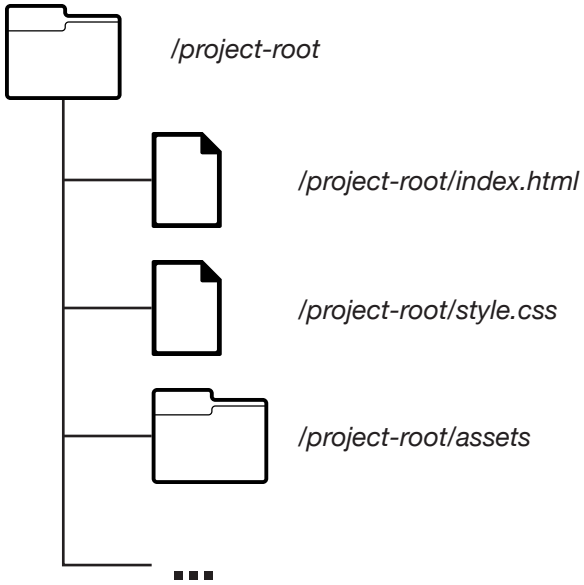
“Content”

CSS

“Style”

Controls the appearance of HTML DOM elements

ORGANIZING THE DIRECTORY



INCLUDING THE STYLESHEET

/project-root/index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello World</title>
    <meta charset="utf-8" />
    <link href="style.css"
rel="stylesheet" />
  </head>
  <body>
    ...
  </body>
</html>
```

/project-root/style.css

```
/*style.css*/
```

NOT A COMPREHENSIVE CSS COURSE, BUT...

Basic CSS syntax

Selectors

Inheritance and specificity

Basic styling

The box model
Size and position
Font and color

Best practice hints and tips

BASIC CSS SYNTAX

```
[selector]{
    [property-name] : [property-value];
}
```

selector

```
body {
    background: rgb(250,250,250);
    font-size: 14px;
    width: 100%;
    height: 100%;
    margin: 0;
    padding: 0;
}
```

SELECTORS

by element

```
p{
    font-family: Helvetica, Arial, sans-serif;
    font-size: 0.8em;
}
```

by class

```
.sub-heading{
    font-size: 1.2em;
}
```

by id

```
#mast-head{
    width: 800px;
}
```

SELECTORS

HTML

```
<h1 class=“intro” id=“header”>Hello World</h1>
```

CSS

```
h1{
    color: #03afeb;
    margin-bottom: 10px;
}
```

SELECTORS

HTML

```
<h1 class=“intro” id=“header”>Hello World</h1>
```

CSS

```
.intro{
    color: #03afeb;
    margin-bottom: 10px;
}
```

```
#header{
    color: #03afeb;
    margin-bottom: 10px;
}
```

SELECTORS

HTML

```
<h1 class=“intro” id=“header”>Hello World</h1>
```

CSS

```
h1.intro{
    color: #03afeb;
    margin-bottom: 10px;
}
```

LET'S GET OUR HANDS DIRTY: COLOR, BACK-GROUND, FONTS, BORDER

SELECTORS

HTML

```
<h1 class=“intro” id=“header”>Hello World</h1>
```

CSS

```
h1{
    margin-bottom: 10px;
}
...
.intro{
    color: #03afeb;
}
```

SELECTORS

Non-conflicting properties will combine.

But what if multiple selectors apply to the same object, and they conflict?

SELECTORS: INHERITANCE & SPECIFICITY

HTML

```
<div class= “featured”>
    <h2>Featured product</h2>
    <p>This product is made from...</p>
</div>
```

CSS

```
.featured{
    color: rgb(255,0,0);
}
```

Everything under .featured, including <h2> and <p>, will inherit this property

INHERITANCE & SPECIFICITY

HTML

```
<div class= “featured”>
    <h2>Featured product</h2>
    <p>This product is made from...</p>
</div>
```

CSS

```
.featured{
    color: rgb(255,0,0);
}
.featured p{
    color: rgb(0,0,0);
}
```

This will override the color on featured

WHAT ABOUT THIS?

HTML

```
<div class="featured" id="top-featured">
    <h2>Featured product</h2>
    <p>This product is made from...</p>
</div>
```

CSS

```
.featured{
    color: rgb(255,0,0);
}
#top-featured{
    color: rgb(0,0,0);
}
```

In general, the more specific selector will override the less specific selector.

But how is this actually determined?

PRIORITY OF SELECTORS (SPECIFICITY)

# of id selectors *# of element selectors*

0, 0, 0, 0

inline or dynamic styles *# of class selectors*

CSS

```
.featured{
    color: rgb(255,0,0);
}
#top-featured{
    color: rgb(0,0,0);
}
```

PRIORITY OF SELECTORS (SPECIFICITY)

HTML

```
<div class="featured" id="top-featured">
    <h2>Featured product</h2>
    <p>This product is made from...</p>
</div>
```

CSS

```
.featured{
    color: rgb(255,0,0);
}
#top-featured{
    color: rgb(0,0,0);
}
```

0, 0, 1, 0

0, 1, 0, 0

ONE MORE EXAMPLE

HTML

```
<div class="featured" id="top-featured">
    <h2 class="featured-heading">Featured
product</h2>
    <p>This product is made from...</p>
</div>
```

CSS

```
#top-featured h2{
    color: rgb(255,0,0);
}
.featured-heading{
    color: rgb(0,0,0);
}
```

?

ONE MORE EXAMPLE

HTML

```
<div class="featured" id="top-featured">
    <h2 class="featured-heading">Featured
product</h2>
    <p>This product is made from...</p>
</div>
```

CSS

```
#top-featured h2{
    color: rgb(255,0,0);
}
.featured-heading{
    color: rgb(0,0,0);
}
```

0, 1, 0, 1

0, 0, 1, 0

ONE MORE EXAMPLE

HTML

```
<div class="featured" id="top-featured">
    <h2 class="featured-heading">Featured
product</h2>
    <p>This product is made from...</p>
</div>
```

CSS

```
#top-featured h2{
    color: rgb(255,0,0);
}
#top-featured .featured-heading{
    color: rgb(0,0,0);
}
```

0, 1, 0, 1

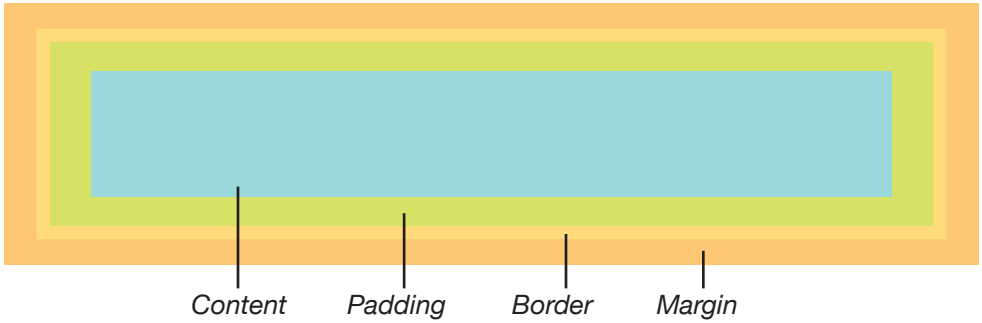
0, 1, 1, 0

BACK TO THE CONSOLE: SEE INHERITANCE IN ACTION

THE BOX MODEL

Every DOM element is a box!

```
<h1>Hello World</h1>
```



THE BOX MODEL

HTML

```
<div class="featured" id="top-featured">
    ...
</div>
```

CSS

```
#top-featured{
    width: 100px;
    border: 1px solid #000;
    padding-left: 5px;
    padding-right: 5px;
}
```

Total box width = width + padding + border

THE BOX MODEL

HTML

```
<div class="container">
    <div class="featured" id="top-featured">
        ...
    </div>
</div><!-- .container-->
```

CSS

```
.container{
    width: 100px;
    border: 1px solid #000;
    padding: 0 5px 0 5px;
}
.container .featured{ width: 100%; }
```

POSITIONING THESE BOXES

CSS

```
.container{
    width: 100px;
    border: 1px solid #000;
    padding: 0 5px 0 5px;
    position: relative;
}
```

OBSERVE THE NATURAL STACKING ORDER

Inspect your unstyled document for its document flow

WHAT OTHER POSSIBILITIES ARE THERE?

relative Position according to normal document flow, then shift using positioning properties e.g. `top` or `left`

absolute Take out of normal flow, and manually position against the containing element

fixed Take out of normal flow, and manually position against the window

In-class Exercise 2

Styling the page you've just marked up.

OK, WHAT HAVE WE LEARNED

Basic CSS syntax

Selectors

Elements inherit properties from parent.

Non-conflicting properties combine; conflicts are resolved based on rules of specificity.

Basic styling

Every DOM element is a box (“the box model”).

Possible positions (absolute, relative, fixed).

Best practice hints and tips

CSS BEST PRACTICE 1

Don’t Repeat Yourself

CSS BEST PRACTICE 1

Use inheritance wisely

HTML

```
<body>
    <div class= “container”>
        <h1>Title</h1>
        <p>Some text</p>
        <a href= “...”>Go somewhere</a>
    </div>
</body>
```

CSS BEST PRACTICE 1

When you find yourself writing the same style over and over again...combine selectors

CSS

```
p{
    font-size:12px;
}
h5{
    font-size:12px;
}
.featured-text{
    font-size:12px;
}
```

CSS

```
p, h5, .featured-text{
    font-size:12px;
}
```

CSS BEST PRACTICE 1

What is they are only mostly the same?

HTML

```
<div class=“nav-buttons”>
    <a href= “#” class= “button” id= “left”>Left</a>
    <a href= “#” class= “button” id= “right”>right</a>
</div><!-- .container-->
```

CSS

```
.nav-buttons .buttons{
    width: 50px;
    height: 50px;
    position: absolute;
}
```

```
.nav-buttons #left{
    left:0;
}
.nav-buttons #right{
    left: 50px;
}
```

CSS BEST PRACTICE 2

Using shorthands

CSS BEST PRACTICE 3

Centering an element

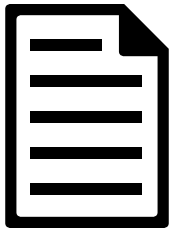
HOW IS EVERYTHING RELATED?

JavaScript

“Behavior”

All the dynamic stuff, such as animation, user interaction, manipulating DOM elements...

HTML



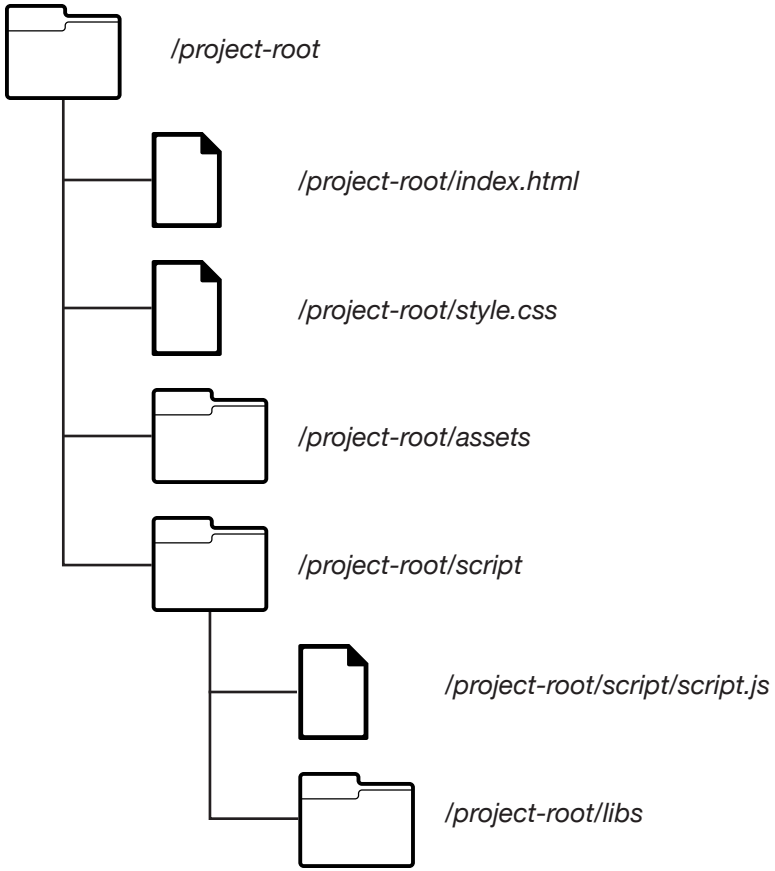
“Content”

CSS

“Style”

Controls the appearance of HTML DOM elements

ORGANIZING THE DIRECTORY



INCLUDING SCRIPTS

/project-root/index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello World</title>
    <meta charset="utf-8" />
    <link href="style.css"
rel="stylesheet" />
  </head>
  <body>
    ...
    <script ...></script>
  </body>
</html>
```

/project-root/script/script.js

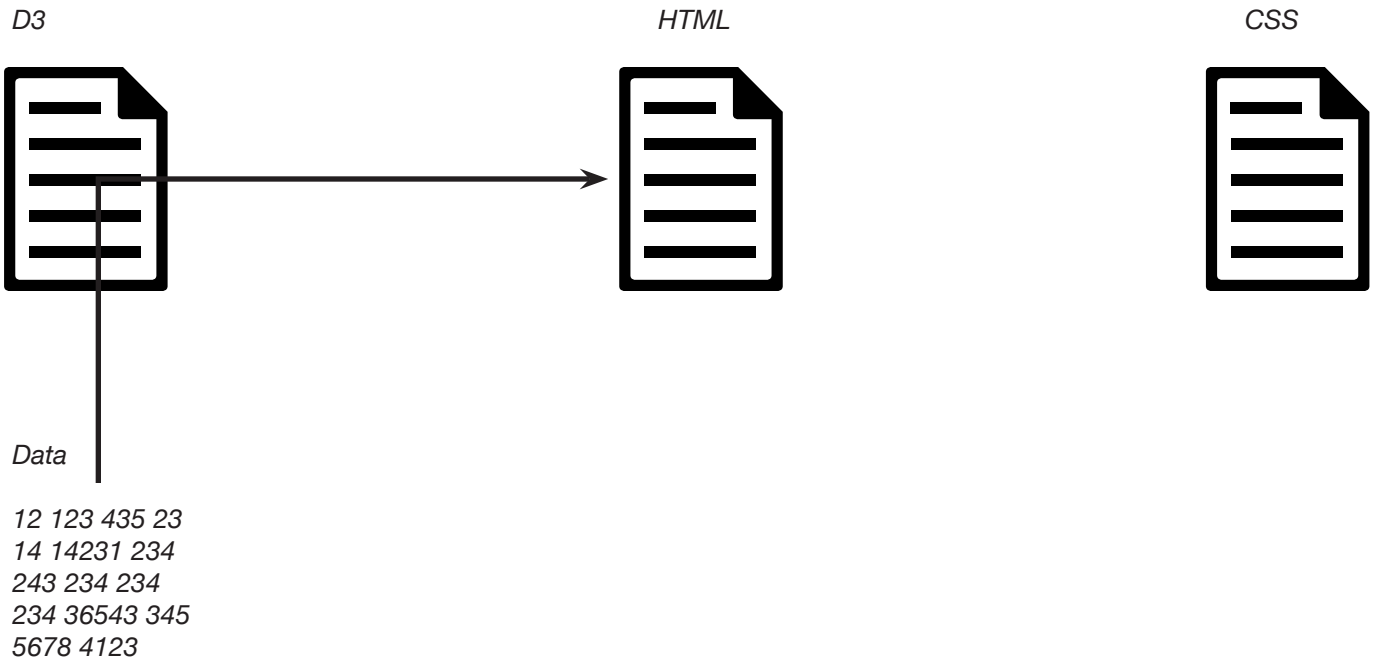
```
//script.js
```

```
<script src= "script/script.js"></script>
```

WHAT CAN A SCRIPT DO?

WHAT ARE LIBRARIES?

INTRO TO D3


D3
HTML
CSS
Data
12 123 435 23
14 14231 234
243 234 234
234 36543 345
5678 4123