Lecture 3
HTML, SVG, JavaScript, D3.js

DTS204TC Data Visualisation

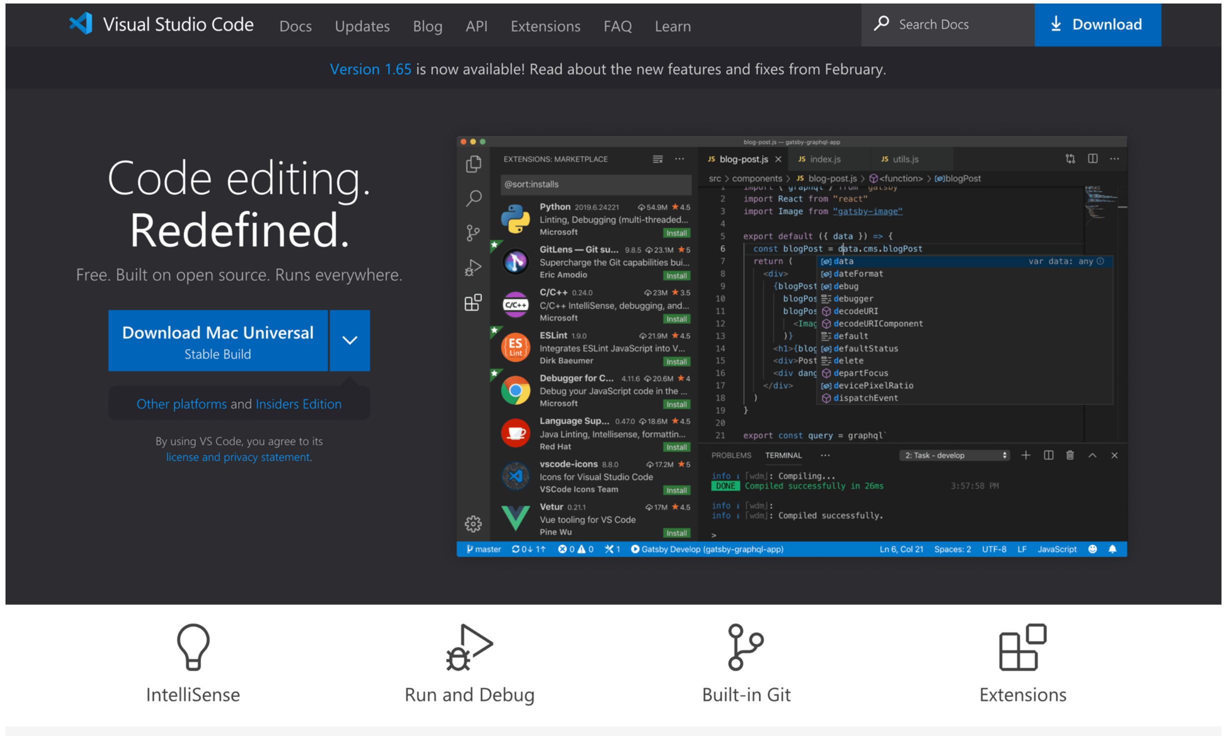
Xi'an Jiaotong-Liverpool University
西交利物浦大学

Outline

- Visual Studio Code (VS Code)
- HTML (HyperText Markup Language)
- SVG (Scalable Vector Graphics)
- Java Script
- D3.js

VS Code

- Download
  - Link: https://code.visualstudio.com/



Version 1.65 is now available! Read about the new features and fixes from February.

Code editing. Redefined.

Free. Built on open source. Runs everywhere.

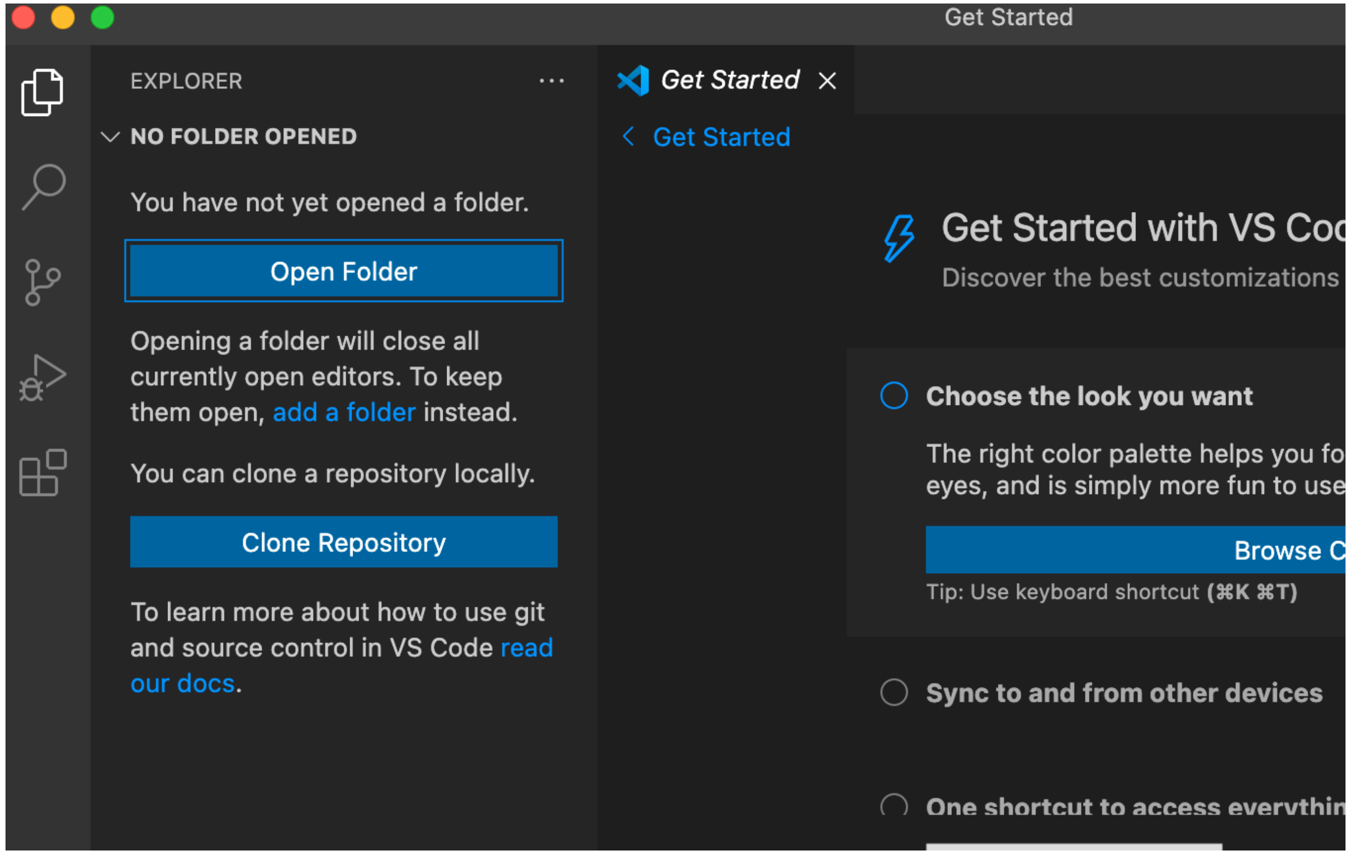
By using VS Code, you agree to its license and privacy statement.

```
import React from "react"
import Image from "gatsby-image"

export default ({ data }) => {
  const blogPost = data.cms.blogPost
  return (
    <div>
      {blogPost
        blogPos
        blogPos
          <Imag
        )}
      <h1>{blog
      <div>Post
      <div dang
    </div>
  )
}

export const query = graphql`
```

```
info ｉ ｢wdm｣: Compiling...
DONE  Compiled successfully in 26ms                3:57:58 PM

info ｉ ｢wdm｣:
info ｉ ｢wdm｣: Compiled successfully.
>
```

VS Code

- Create an HTML file
  - EXPLORER → Open Folder



Get Started

EXPLORER ···

NO FOLDER OPENED

You have not yet opened a folder.

Open Folder

Opening a folder will close all currently open editors. To keep them open, add a folder instead.

You can clone a repository locally.

Clone Repository

To learn more about how to use git and source control in VS Code read our docs.

Get Started ×

< Get Started

Get Started with VS Coc

Discover the best customizations

Choose the look you want

The right color palette helps you fo eyes, and is simply more fun to use

Browse C

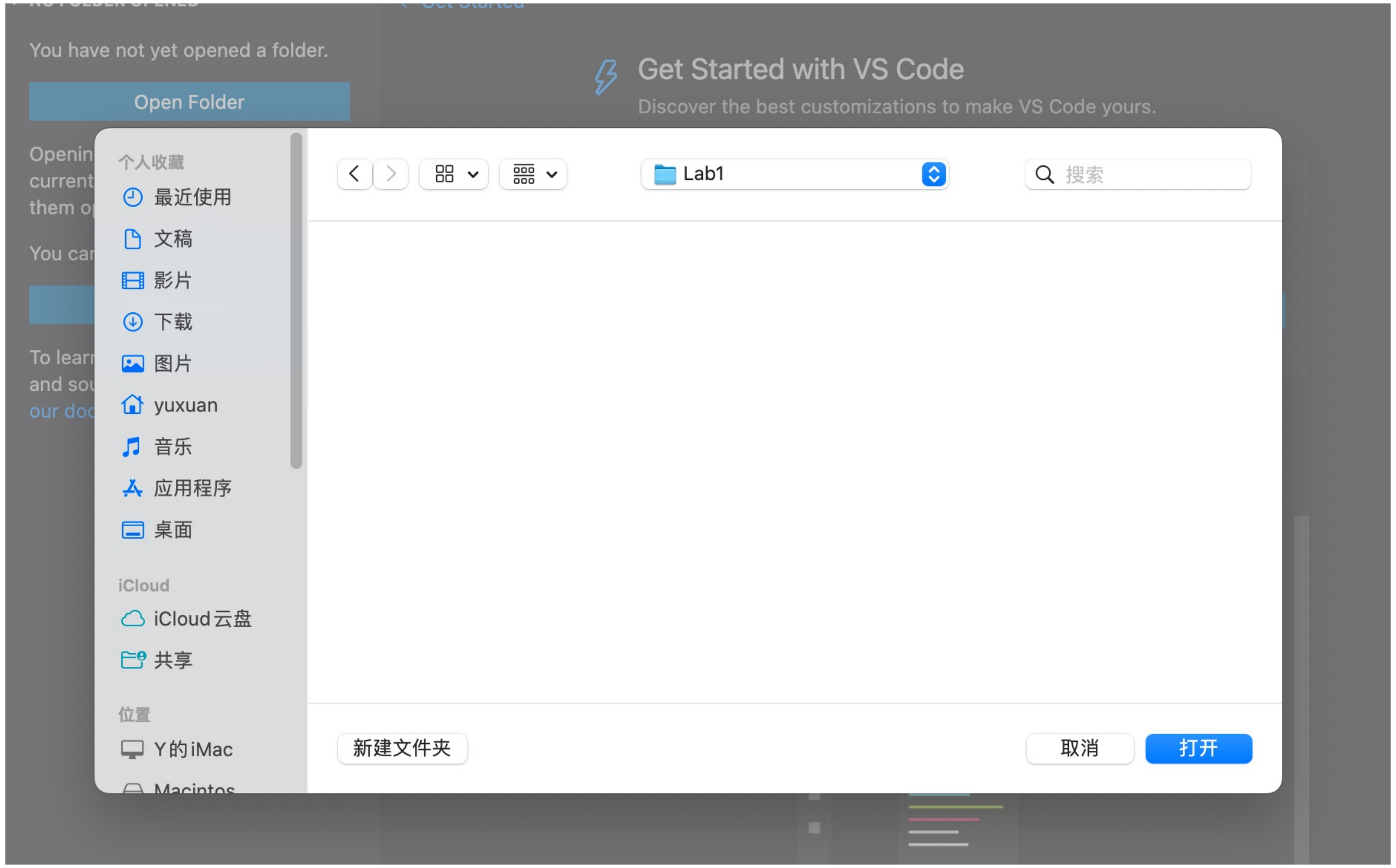
Tip: Use keyboard shortcut (⌘K ⌘T)

Sync to and from other devices

One shortcut to access everythin

VS Code

- Create an HTML file
  - Open the folder that you want to use for DTS204TC



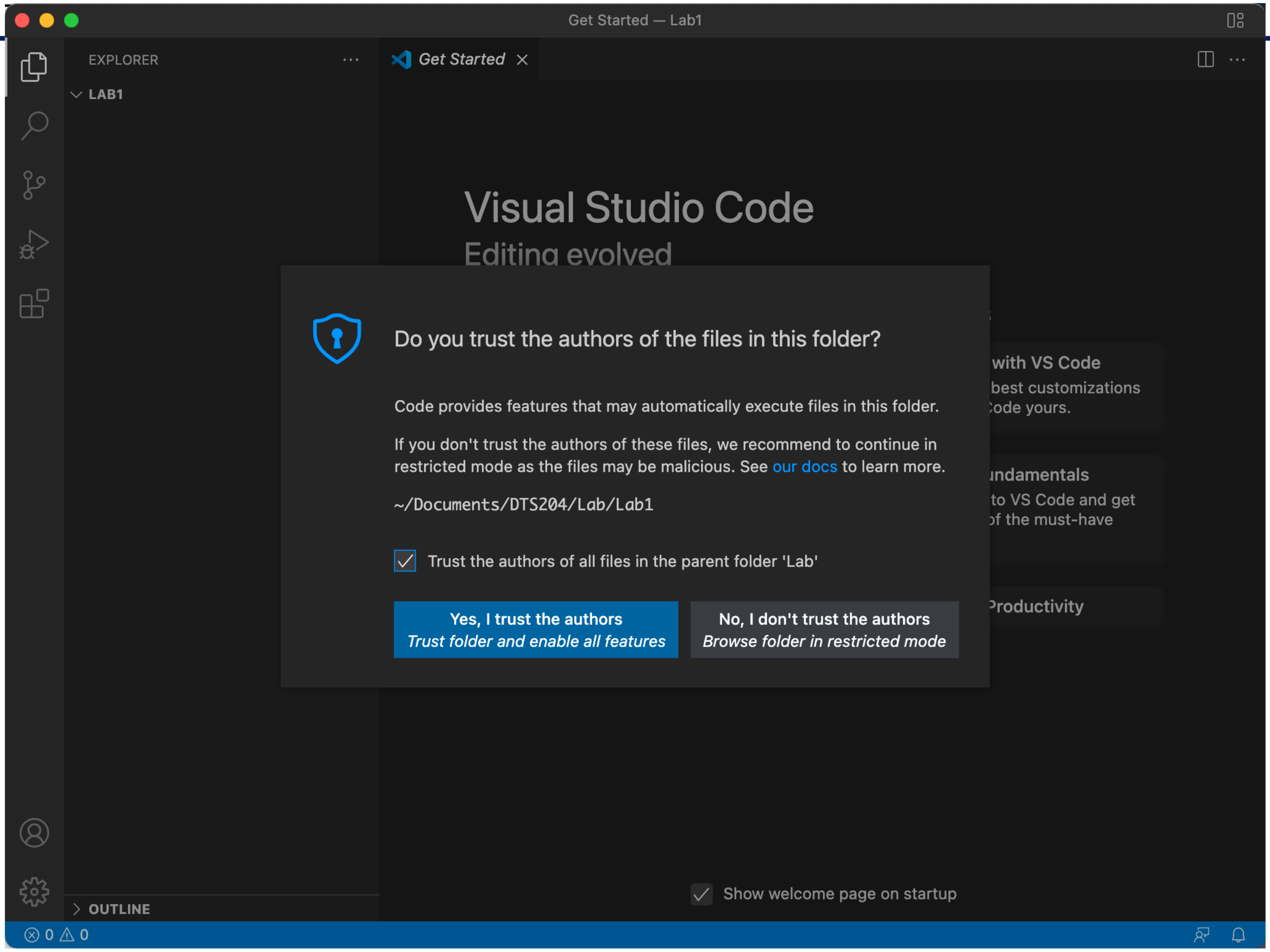
You have not yet opened a folder.

Get Started with VS Code

Discover the best customizations to make VS Code yours.

VS Code

- Create an HTML file
  - *select "Yes, I trust the authors"*



Do you trust the authors of the files in this folder?

Code provides features that may automatically execute files in this folder.

If you don't trust the authors of these files, we recommend to continue in restricted mode as the files may be malicious. See our docs to learn more.

~/Documents/DTS204/Lab/Lab1

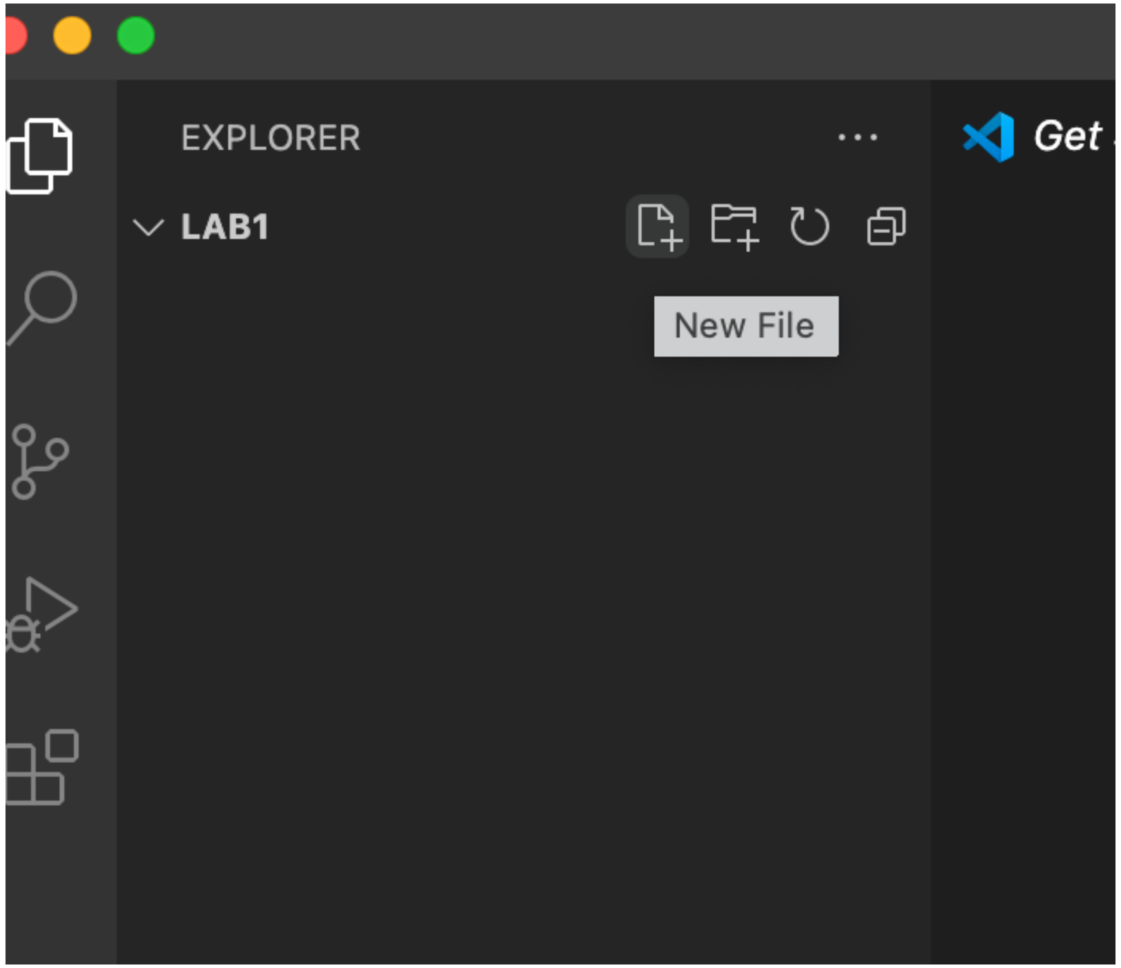
☑ Trust the authors of all files in the parent folder 'Lab'

Yes, I trust the authors
Trust folder and enable all features

No, I don't trust the authors
Browse folder in restricted mode

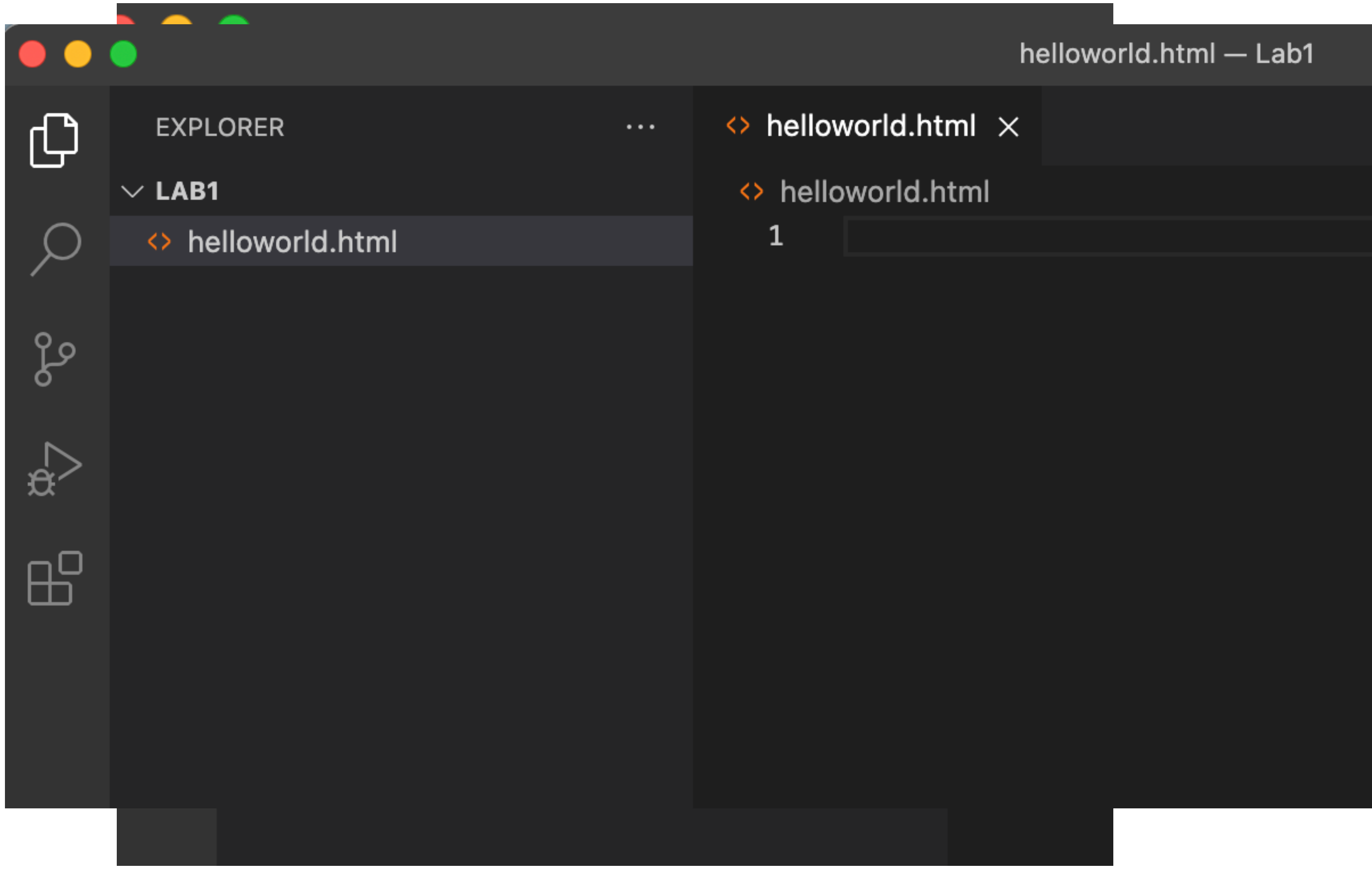
VS Code

- Create an HTML file
    - Create a new file called **“helloworld.html”**



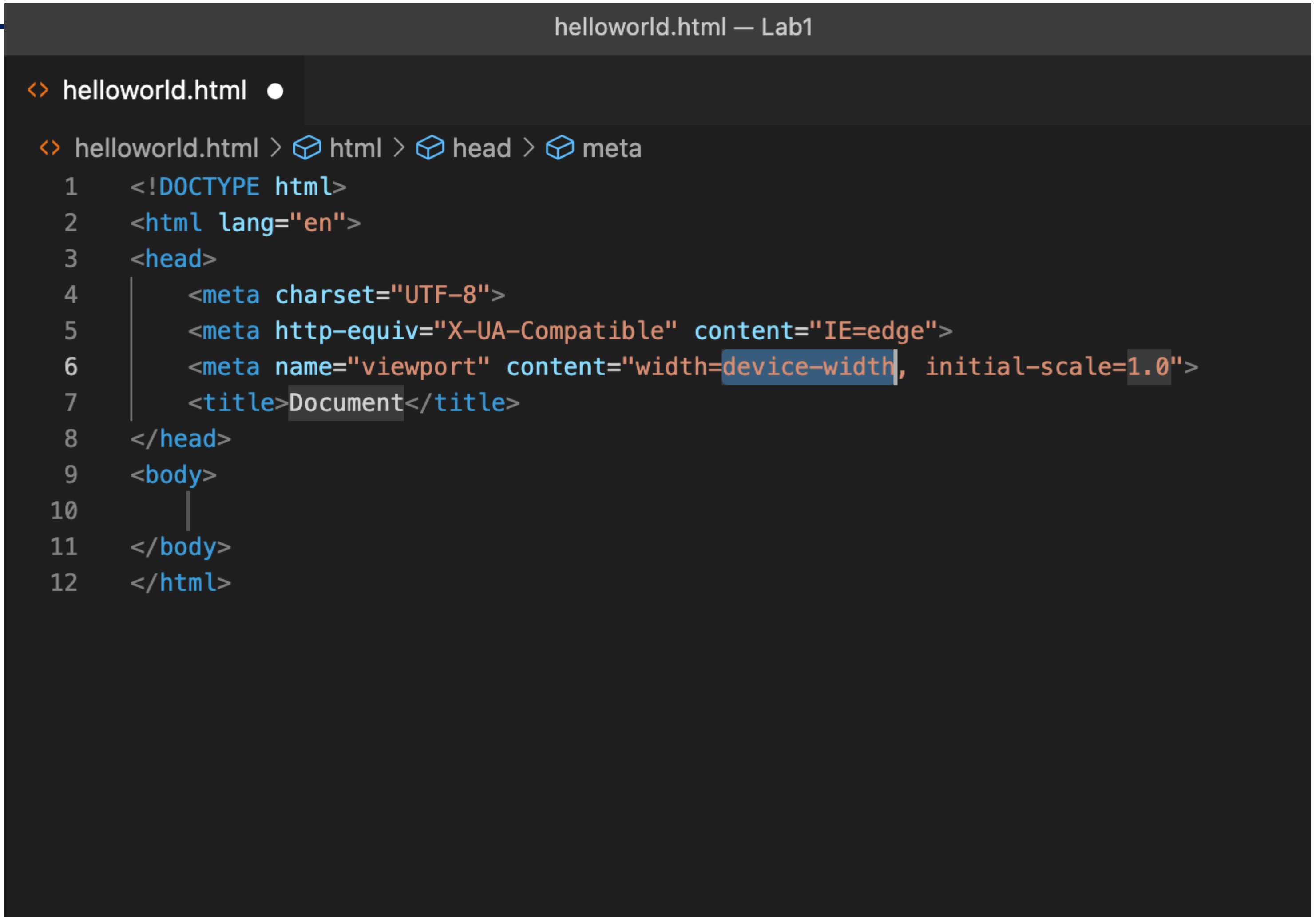
VS Code

- Create an HTML file
  - Create a new file called **"helloworld.html"**



VS Code

- Create an HTML file
    - type **“!”** and press **“Return”** on your keyboard. You will find a basic structure of a .html file.



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    
</body>
</html>
```

HTML

- HTML

HTML is used to structure the content of the web page. The current version is HTML 5. It is stored in a text file with the extension ".html".

- DOM (Document Object Model)

When you write html code for your page, it gets converted to a hierarchical structure on the browser. Every tag in html gets converted to an element in the DOM with a parent-child hierarchy.

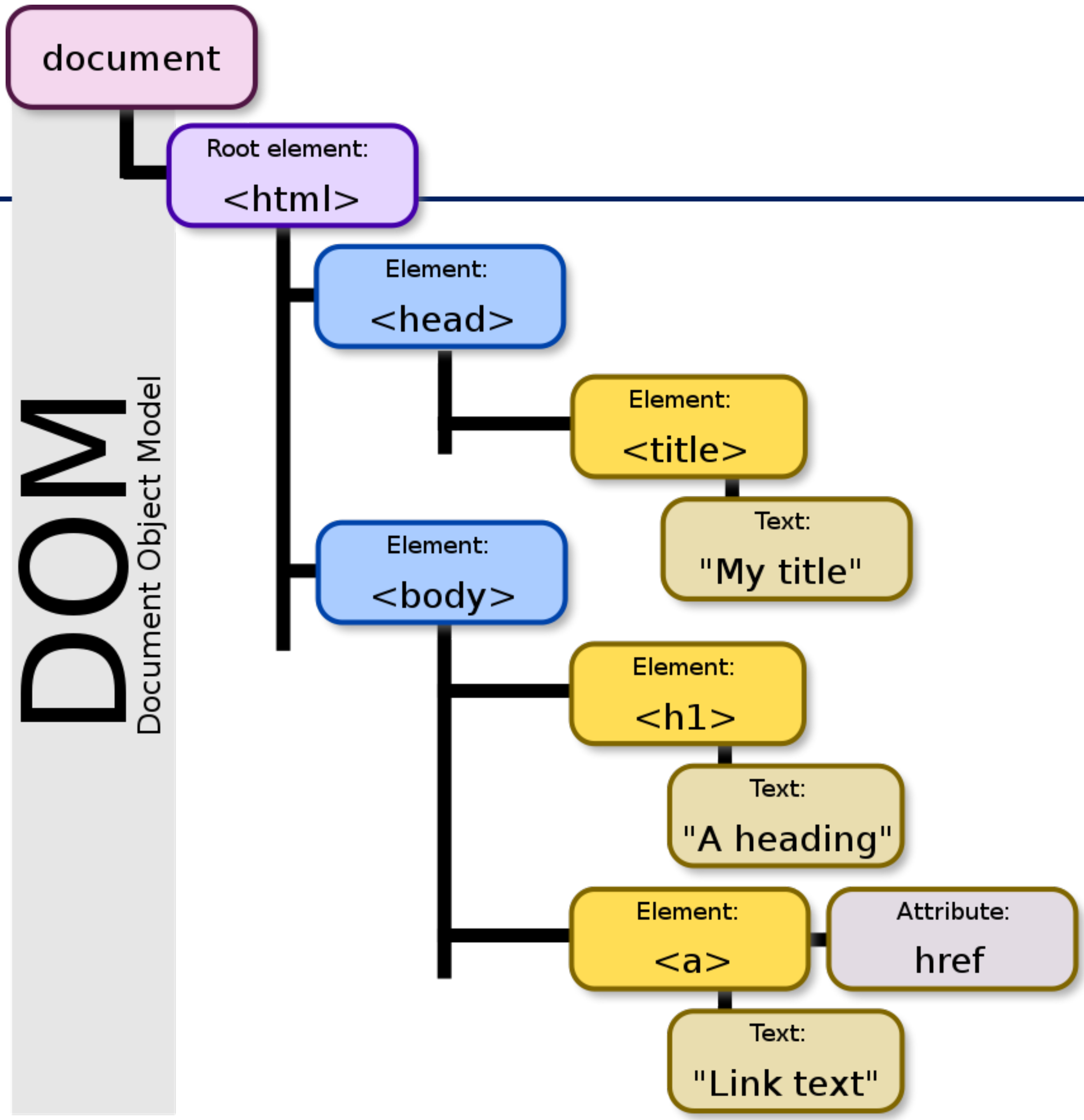
HTML

- HTML-Tags

  - <html>: Main Tag. Necessary for every HTML file.
  - <head>: May contains title, links…
  - <body>:main part of html
  - <title>:
  - **<script>: For JavaScript codes**
  - **<svg>: For SVG elements**

HTML

- DOM


DOM
Document Object Model
document
Root element:
<html>
Element:
<head>
Element:
<title>
Text:
"My title"
Element:
<body>
Element:
<h1>
Text:
"A heading"
Element:
<a>
Attribute:
href
Text:
"Link text"

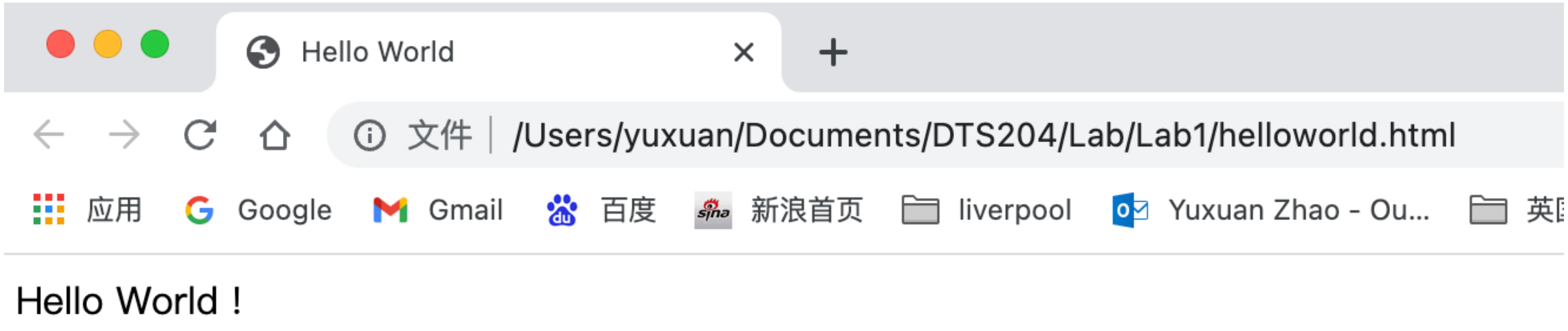
HTML

- Example: Hello World

```
<!DOCTYPE html>
<html lang="en">

<head>
		<title>Hello World</title>
</head>

<body>
		Hello World !
</body>

</html>
```

HTML

- Example: Hello World



Hello World !

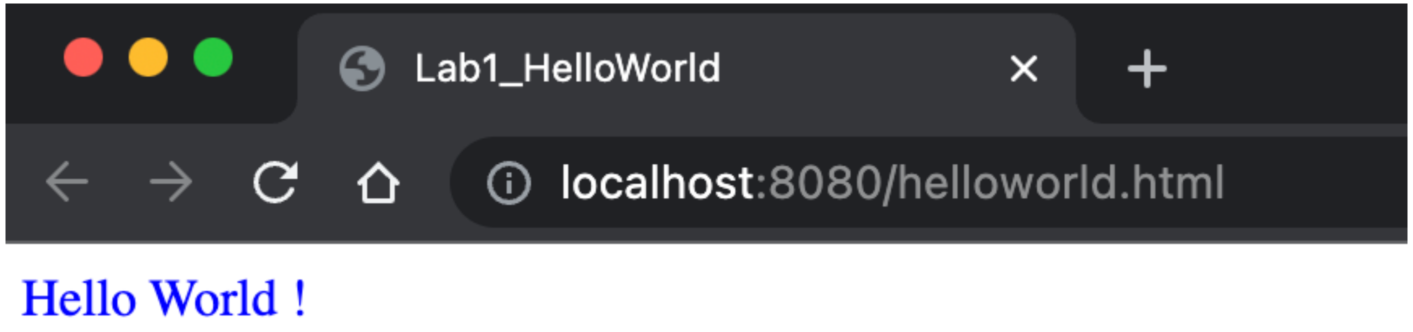
HTML

- CSS (**Cascading Style Sheets**)

Cascading Style Sheets (**CSS**) is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media. More information:

https://developer.mozilla.org/en-US/docs/Web/CSS

```
<style>
            body {
            color: blue;
            }
</style>

<body>
            Hello World !
</body>
```



Hello World !

SVG

- Scalable Vector Graphics

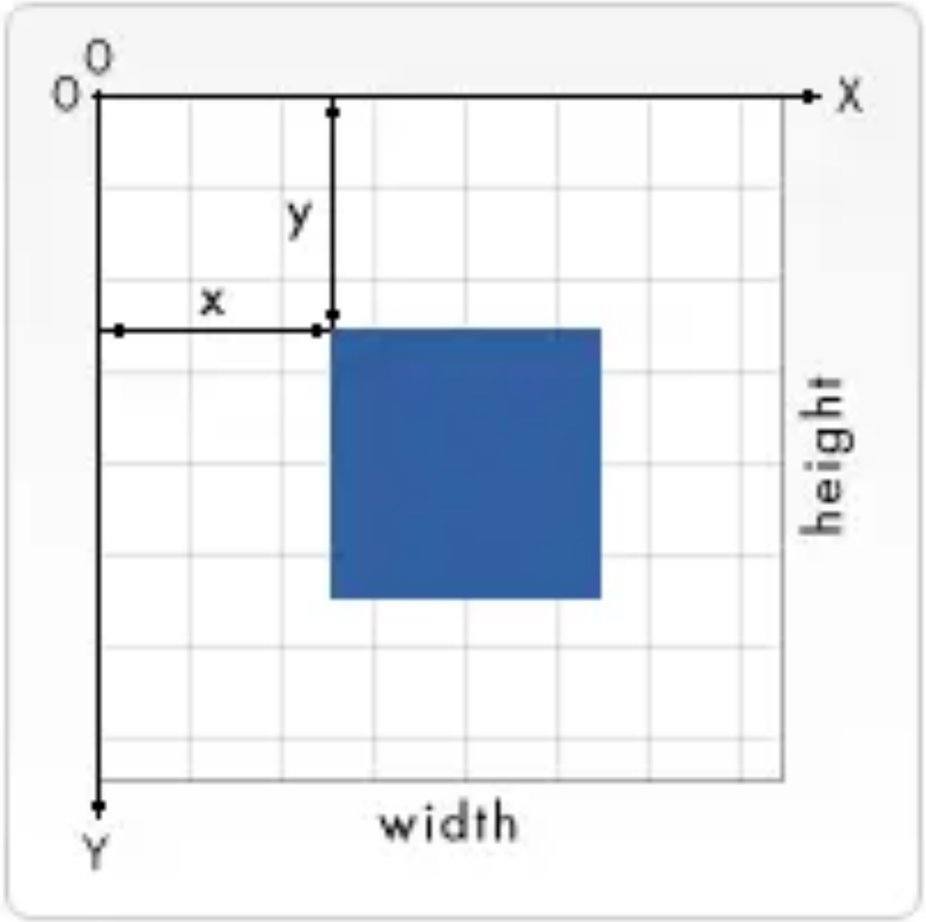
SVG is a way to render images on the web page. SVG is not a direct image but is just a way to create images using text. It scales itself according to the size of the browser, so resizing your browser will not distort the image.

To start off, create an SVG tag:

```
<svg width="500" height="500"> </svg>
```

SVG

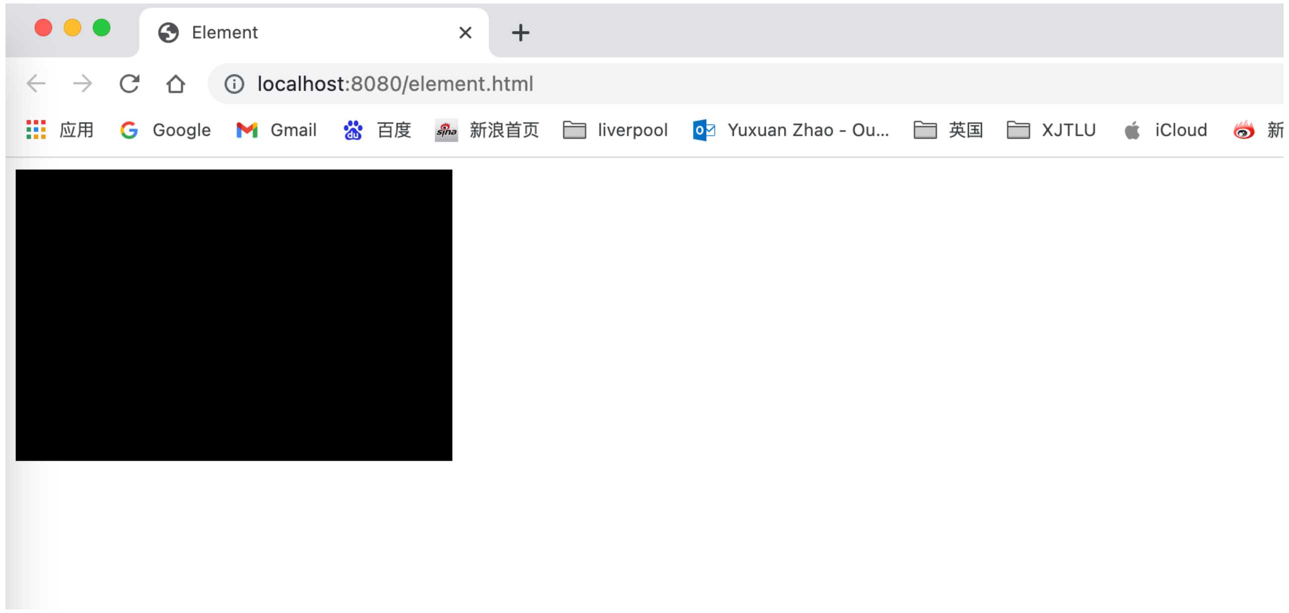
- Draw SVG elements



SVG

- Draw SVG elements

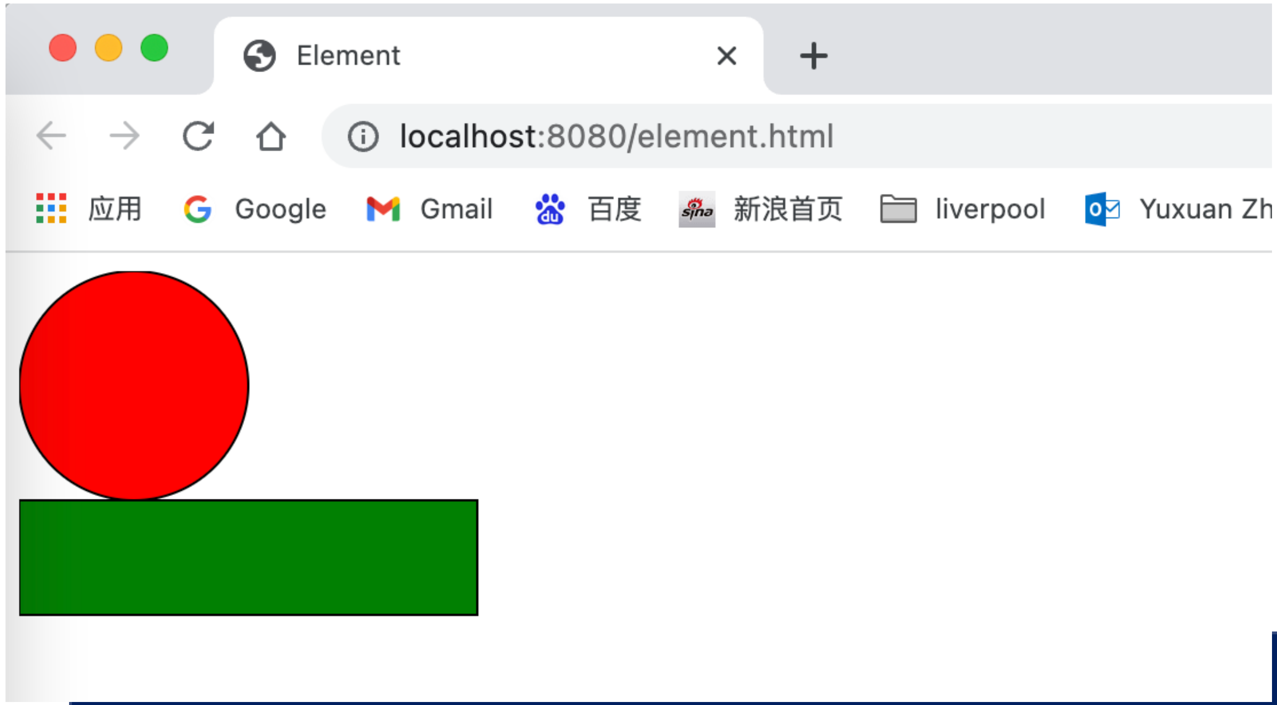
```
<svg width="500" height="500">
        <rect x="0" y="0" width="300" height="200"></rect>
</svg>
```



SVG

- Draw elements in SVG

```
<svg width = "500" height = "500">
        <circle cx = "50" cy = "50" r = "50" stroke = "black" fill = "red"></circle>
        <rect x = “0” y = “100” width = “200” height = “50” stroke = “black” fill = “green”> </rect>
</svg>
```



SVG

- Elements in SVG

You can find all shapes (‘**circle**’, ‘**ellipse**’, ‘**line**’, ‘**path**’, ‘**polygon**’, ‘**polyline**’ and ‘**rect**’.) that SVG supports from:

https://www.w3.org/TR/SVG/shapes.html

SVG

- Styling elements (attributes)

https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute

Style Attribute	Description
fill	This is the fill color for your element. It can be color name, hex value, or RGB or RGBA values.
stroke	This is the stroke color. Like our line example, we can specify a color for our element.
stroke-width	Stroke width specifies the width of our line or boundary. This is in pixels.
opacity	Opacity will specify an opacity/transparency number. 0 is completely transparent and 1 is completely opaque.
font-family	For text elements, we can specify the font-family. This works like CSS.
font-size	We can also specify the font-size for text elements.

SVG

- Styling elements (attributes)

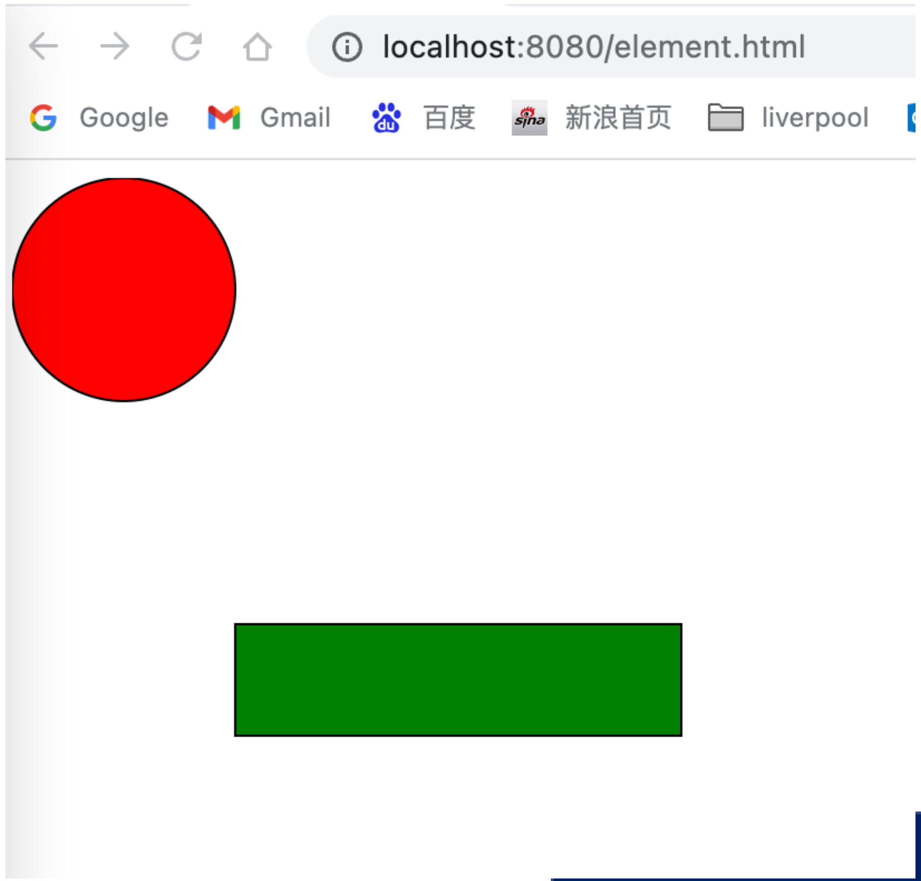
Transform: translate, rotate, scale

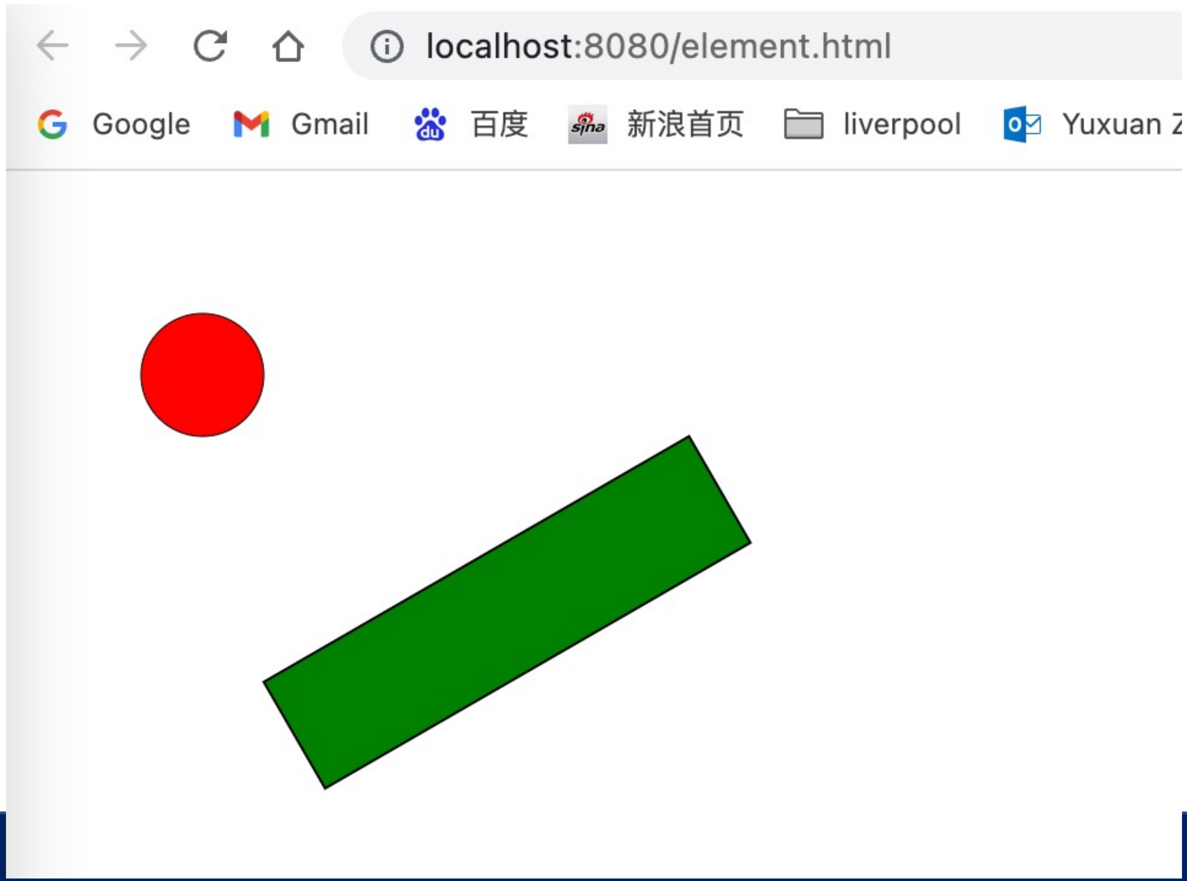
```
<svg width = "500" height = "500">
        <circle cx = "50" cy = "50" r = "50" stroke = "black" fill = "red"
        transform = "scale(0.5,0.5) translate(100,100)"></circle>

        <rect x = "100" y = "200" width = "200" height = "50" stroke = "black" fill = "green"
        transform = “rotate(-30,100 200)"></rect>
</svg>
```

SVG

- Styling elements (attributes)

Transform: translate, rotate, scale





Java Script

- JavaScript is a loosely-typed client side scripting language that executes in the user's browser. JavaScript interact with html elements (DOM elements) in order to make the web user interface interactive.

JavaScript

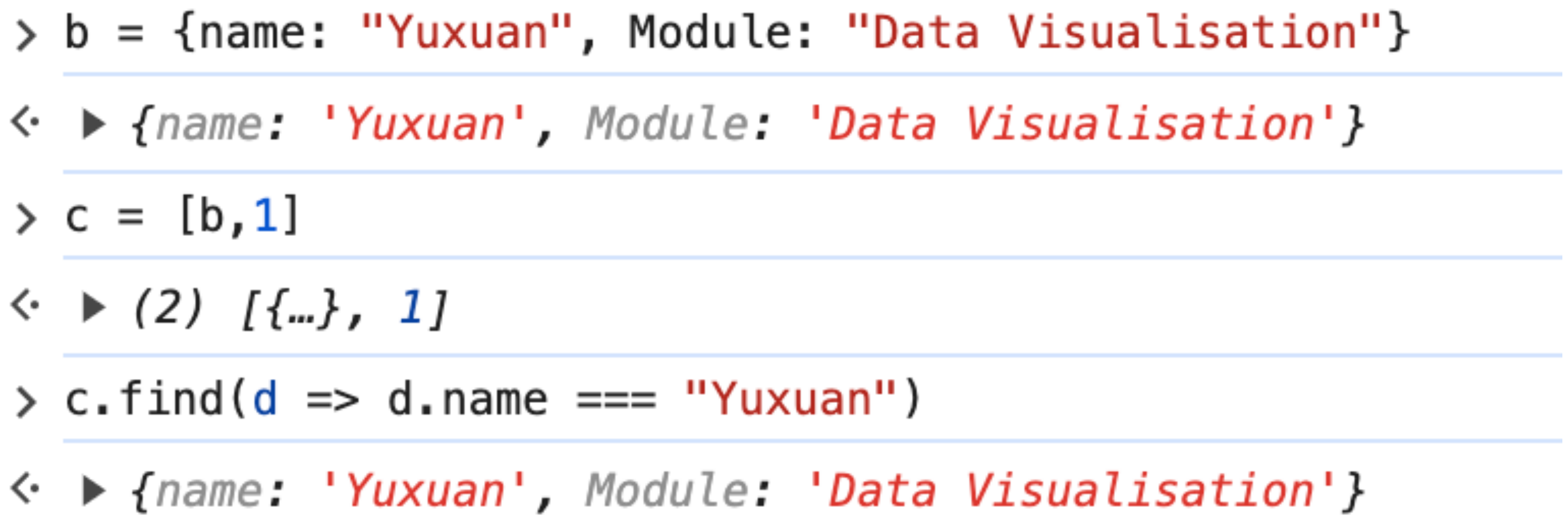
- JavaScript
  - Web Development Language
  - When you declare a variable, you do not need to specify its type (int, float…)
    - const (constant value), let (block scoped), var (globally scoped)
  - Functions
    - **function abc(a){ return a + 5; }** // declare a function named "abc" that takes a parameter "a" and returns "a + 5"
    - **let f = datum => datum.value;** // The arrow function syntax (=>) is used to declare an anonymous function assigned to the variable "f" that takes a parameter "datum" and returns "datum.value".
    - **const p = function(a, b) { return a + b; }** // a function expression to declare a function assigned to the constant "p". This function takes two parameters "a" and "b" and returns their sum.
    - **let myFunction = (a, b) => a + b** // Another arrow function syntax is used to declare a function assigned to the variable "myFunction", which takes two parameters "a" and "b" and returns their sum.
  - A function can be declared as a variable

JavaScript

- JavaScript - Web Development Language
  - Common interfaces
    - Console.log(“Hello World”)
    - Object: b = {name: “Yuxuan”, Module: ”Data Visualisation” }
    - Array: c = [b,1]
    - Sort: c.sort()
    - Query: c.find(d => d.name === “Yuxuan”)
  - String ➔ Number: +(“123”)
  - Template String
    - let a = 10;
    - let myString = `abc-${a}`; ➔ “abc-10”

JavaScript

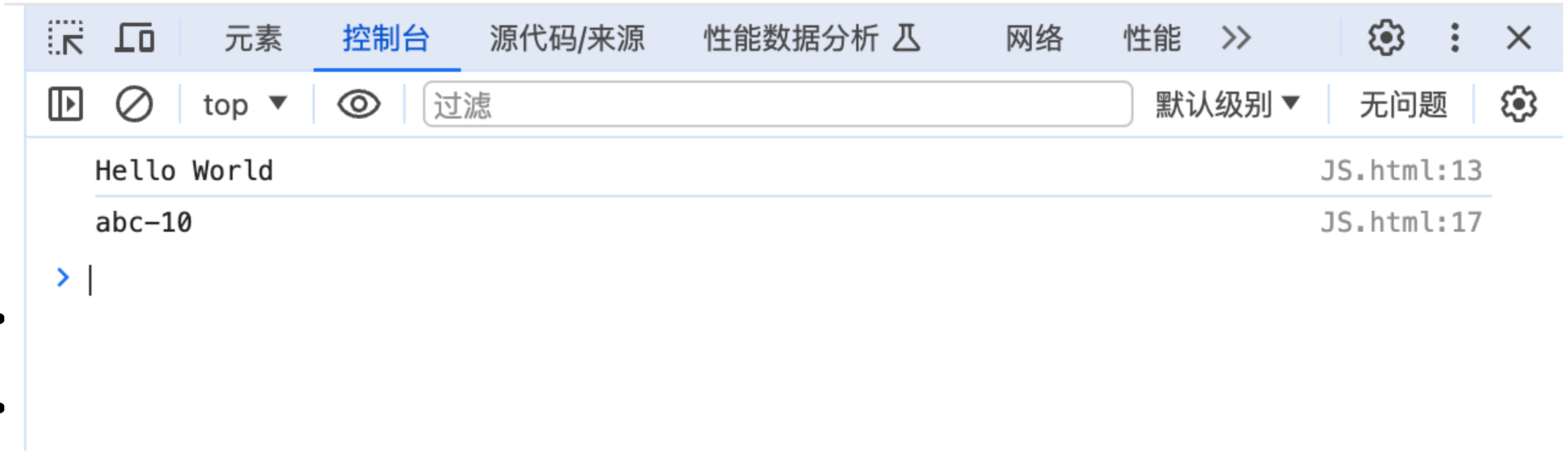
- JavaScript - Web Development Language
  - Common interfaces



```
> b = {name: "Yuxuan", Module: "Data Visualisation"}
<· ▶ {name: 'Yuxuan', Module: 'Data Visualisation'}
> c = [b,1]
<· ▶ (2) [{…}, 1]
> c.find(d => d.name === "Yuxuan")
<· ▶ {name: 'Yuxuan', Module: 'Data Visualisation'}
```

JavaScript

- JavaScript - Web Development Language
  - Common interfaces
    - Console.log(“Hello World”) //



```
Hello World                                              JS.html:13
abc-10                                                   JS.html:17
```

-
-

    - let a = 10;
    - let myString = `abc-${a}`; ➔ “abc-10”

D3.js

- What is D3

D3 stands for **Data-Driven Documents**. It is an open-source JavaScript library to create custom interactive data visualizations in the web browser using SVG, HTML and CSS.

- Official web site: d3js.org

D3.js

- Setup D3.js development environment

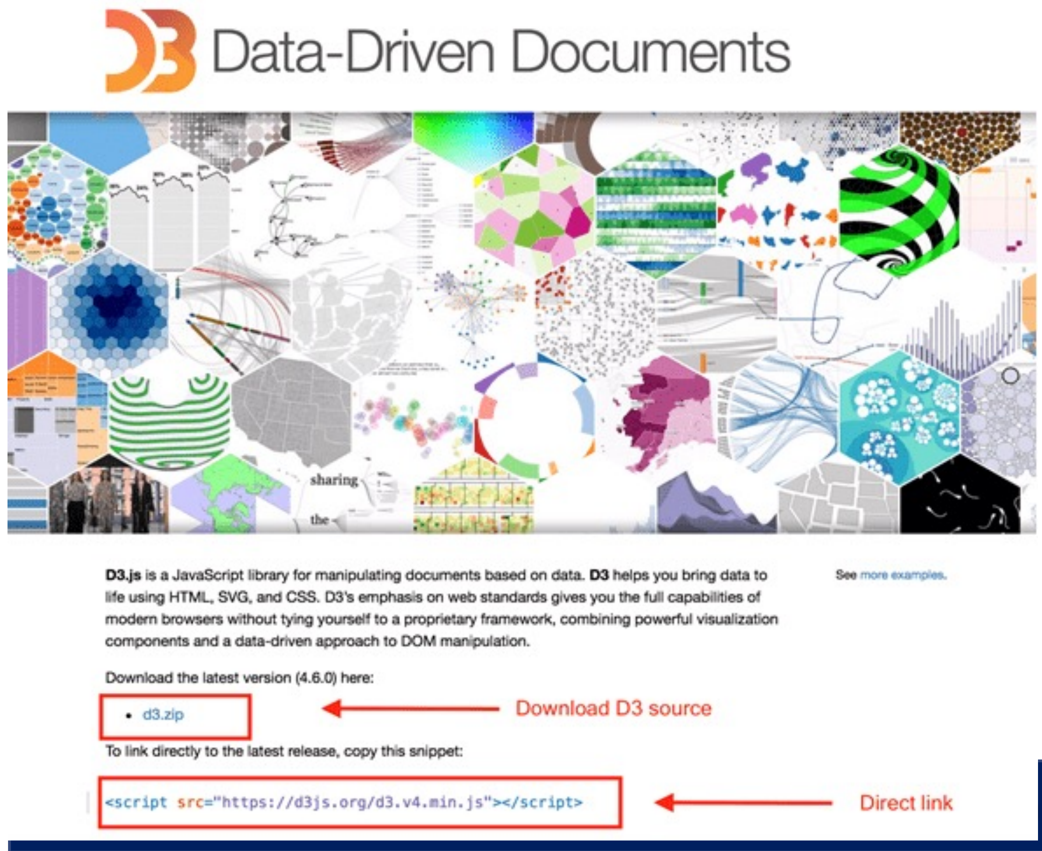
You need to include D3.js library into your HTML webpage in order to use D3 to create data visualization. You can do it in **TWO** ways:

1. Include D3 library from your project's folder

2. Include D3 library from CDN (Content Delivery Network)

D3.js

- Download D3 Library
  - Visit the D3 website: https://d3js.org
  - Download the latest version of d3 (d3.zip)
  - For coursework, please use the D3 in the LMO page



Data-Driven Documents

D3.js is a JavaScript library for manipulating documents based on data. **D3** helps you bring data to life using HTML, SVG, and CSS. D3's emphasis on web standards gives you the full capabilities of modern browsers without tying yourself to a proprietary framework, combining powerful visualization components and a data-driven approach to DOM manipulation.

See more examples.

Download the latest version (4.6.0) here:

- d3.zip

Download D3 source

To link directly to the latest release, copy this snippet:

```
<script src="https://d3js.org/d3.v4.min.js"></script>
```

Direct link

D3.js

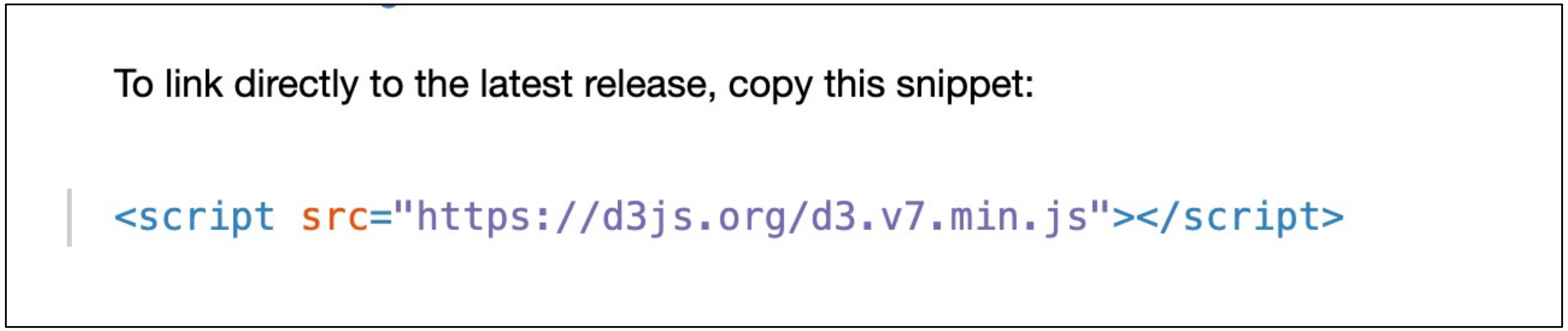
- Download D3 Library
  - After the download is complete, unzip the d3 folder and look for d3.min.js This is the minified version of the d3 source code. Copy d3.min.js file and paste it to your project's root folder or any other folder where you want to keep all your library files. Include d3.min.js file in your HTML page as shown below.

```
<head>
        <script src="../d3.min.js"></script>
</head>
```

D3.js

- Include D3 Library from CDN
    - Include D3 library using CDN url

```
<script src="https://d3js.org/d3.v7.min.js"></script>
```



To link directly to the latest release, copy this snippet:

```
<script src="https://d3js.org/d3.v7.min.js"></script>
```

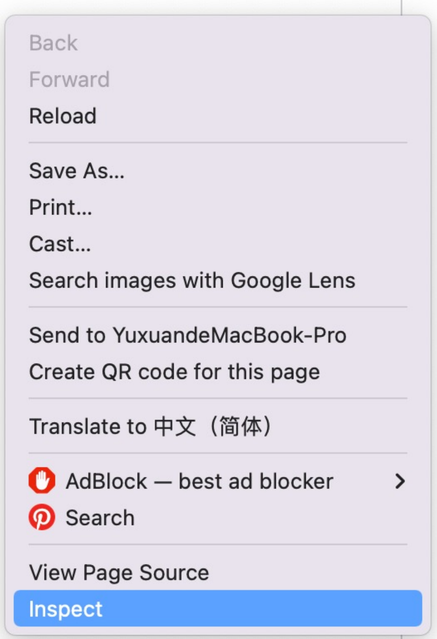
D3.js

- D3 is a JavaScript code so you can write all your D3 code within <script> tag. You may need to manipulate existing DOM elements, so it is advisable to write D3 code just before the end of </body> tag

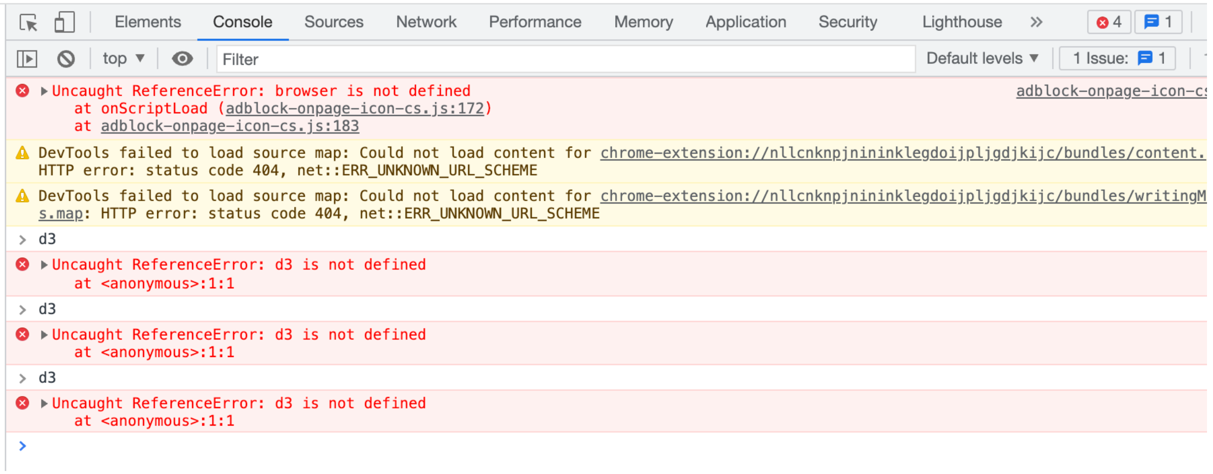
```
<!DOCTYPE html>
<html lang="en">
<head>
		<script src="../d3.min.js"></script>
</head>
<body>
		<script>
				// write your d3 code here..
		</script>
</body>
</html>
```

D3.js

- Check your set up:
  - To make sure that you have set up D3.js environment correctly, you can type **d3** in the Console of your browser:



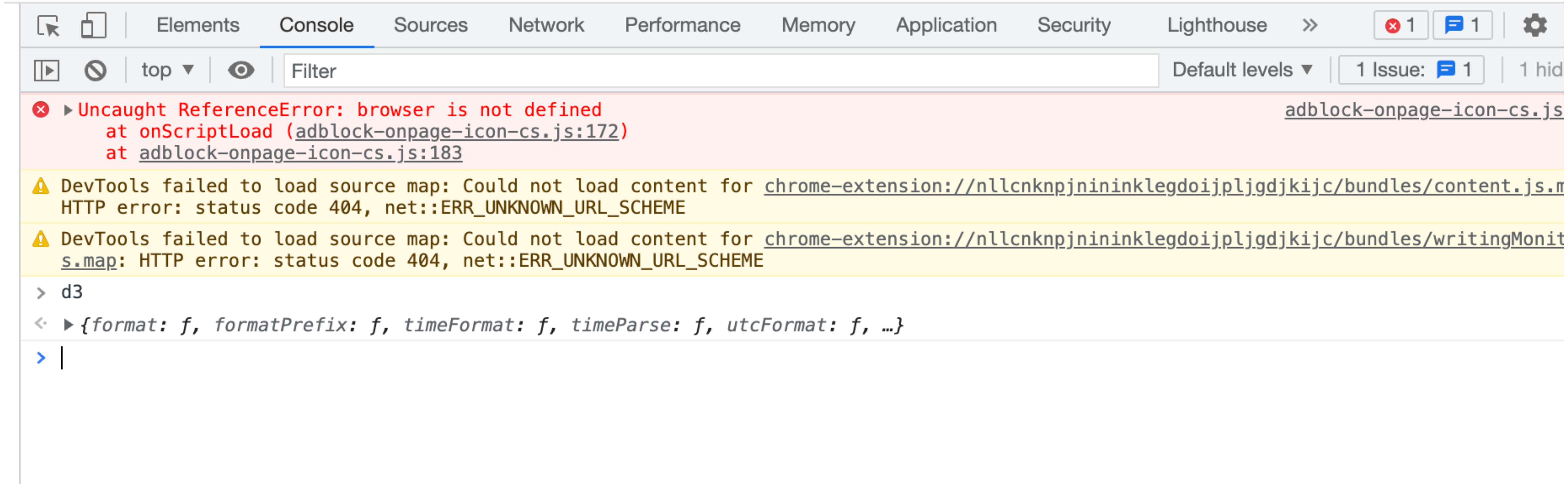
Back
Forward
Reload
Save As...
Print...
Cast...
Search images with Google Lens
Send to YuxuandeMacBook-Pro
Create QR code for this page
Translate to 中文（简体）
AdBlock — best ad blocker
Search
View Page Source
Inspect



```
Uncaught ReferenceError: browser is not defined                                   adblock-onpage-icon-cs
    at onScriptLoad (adblock-onpage-icon-cs.js:172)
    at adblock-onpage-icon-cs.js:183
DevTools failed to load source map: Could not load content for chrome-extension://nllcnknpjnininklegdoijpljgdjkijc/bundles/content.
HTTP error: status code 404, net::ERR_UNKNOWN_URL_SCHEME
DevTools failed to load source map: Could not load content for chrome-extension://nllcnknpjnininklegdoijpljgdjkijc/bundles/writingM
s.map: HTTP error: status code 404, net::ERR_UNKNOWN_URL_SCHEME
> d3
Uncaught ReferenceError: d3 is not defined
    at <anonymous>:1:1
> d3
Uncaught ReferenceError: d3 is not defined
    at <anonymous>:1:1
> d3
Uncaught ReferenceError: d3 is not defined
    at <anonymous>:1:1
>
```

D3.js

- Check your set up:
  - To make sure that you have set up D3.js environment correctly, you can type **d3** in the Console of your browser:



```
Uncaught ReferenceError: browser is not defined                                    adblock-onpage-icon-cs.js
    at onScriptLoad (adblock-onpage-icon-cs.js:172)
    at adblock-onpage-icon-cs.js:183
DevTools failed to load source map: Could not load content for chrome-extension://nllcnknpjnininklegdoijpljgdjkijc/bundles/content.js.m
HTTP error: status code 404, net::ERR_UNKNOWN_URL_SCHEME
DevTools failed to load source map: Could not load content for chrome-extension://nllcnknpjnininklegdoijpljgdjkijc/bundles/writingMonit
s.map: HTTP error: status code 404, net::ERR_UNKNOWN_URL_SCHEME
> d3
<· ▶{format: f, formatPrefix: f, timeFormat: f, timeParse: f, utcFormat: f, …}
> |
```

D3.js

- D3.js
  - D3 stands for **Data-Driven Documents**. Console.log(“Hello World”)
  - id and class
    - **id** is an unique identifier for an element.
    - **class** is a identifier for a set of elements (designed by you).
  - D3 Query
    - d3.select(“#id”) * by id
    - d3.select(“svg”) *by tag
    - d3.selectAll(“.class1”) *by class
    - d3.select(#secondgroup rect)

D3.js

- Select elements using D3:

Method	Description
d3.select(css-selector)	Returns the first matching element in the HTML document based on specified css-selector
d3.selectAll(css-selector)	Returns all the matching elements in the HTML document based on specified css-selector

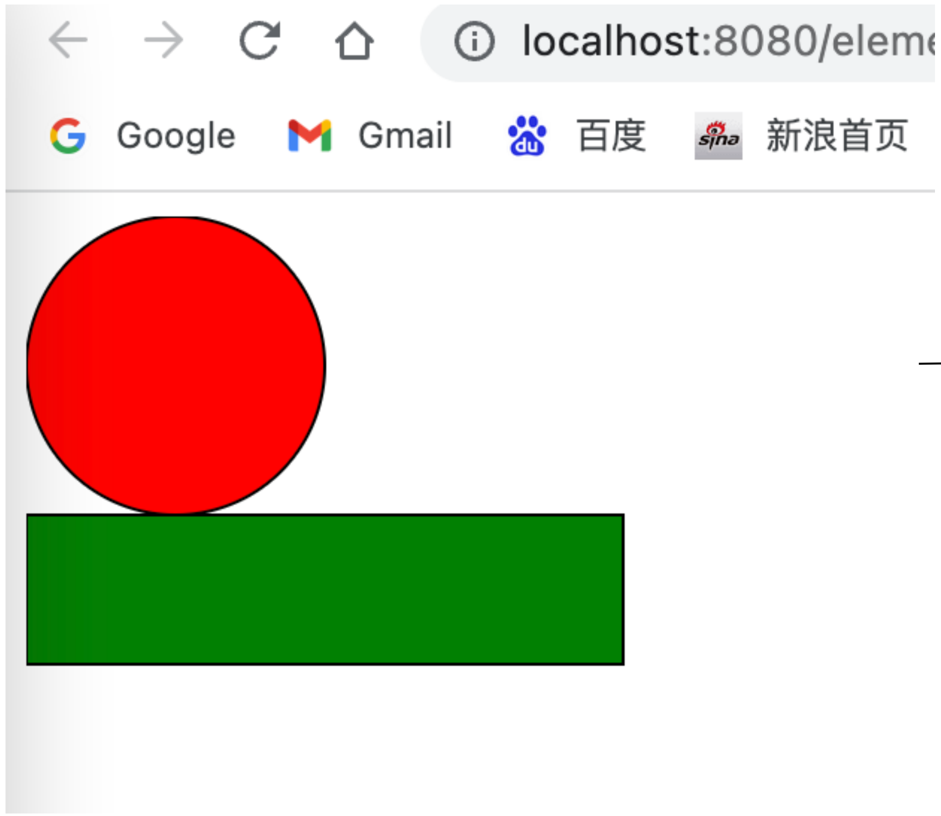
D3.js

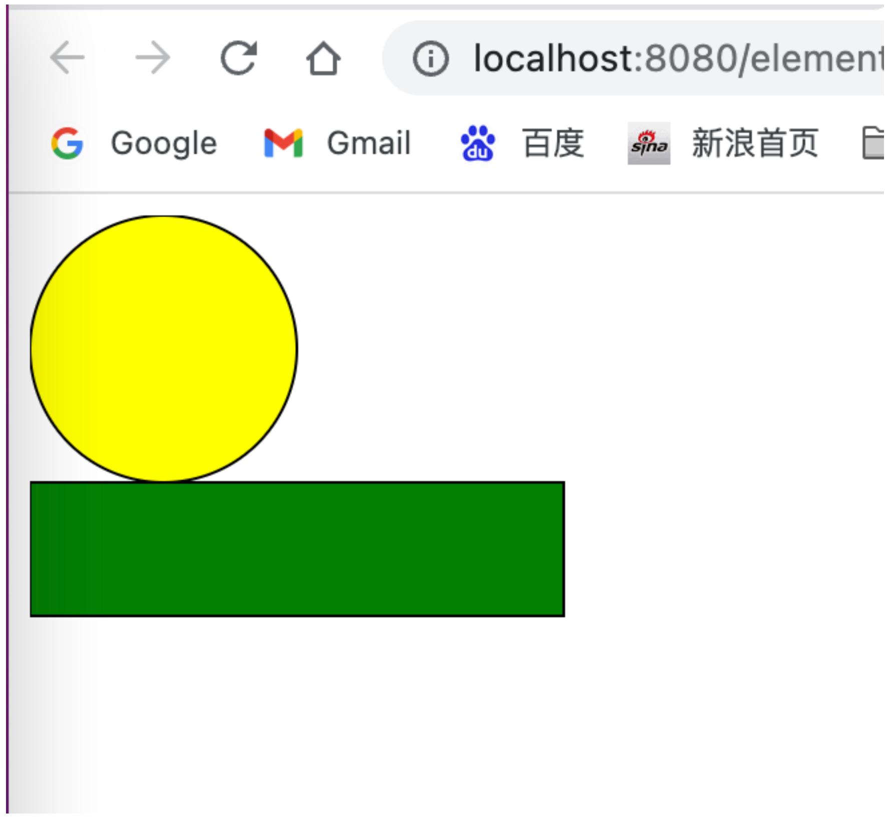
- Select and change SVG elements using D3:

```
<body>
	<svg width = "500" height = "500">
		<circle cx = "50" cy = "50" r = "50" stroke = "black" fill = "red"></circle>
		<rect x = "0" y = "100" width = "200" height = "50" stroke = "black" fill = "green"></rect>
	</svg>

	<script>
		d3.select("circle").attr("fill","yellow");
	</script>
</body>
```

D3.js

- Select and change SVG elements using D3:





D3.js

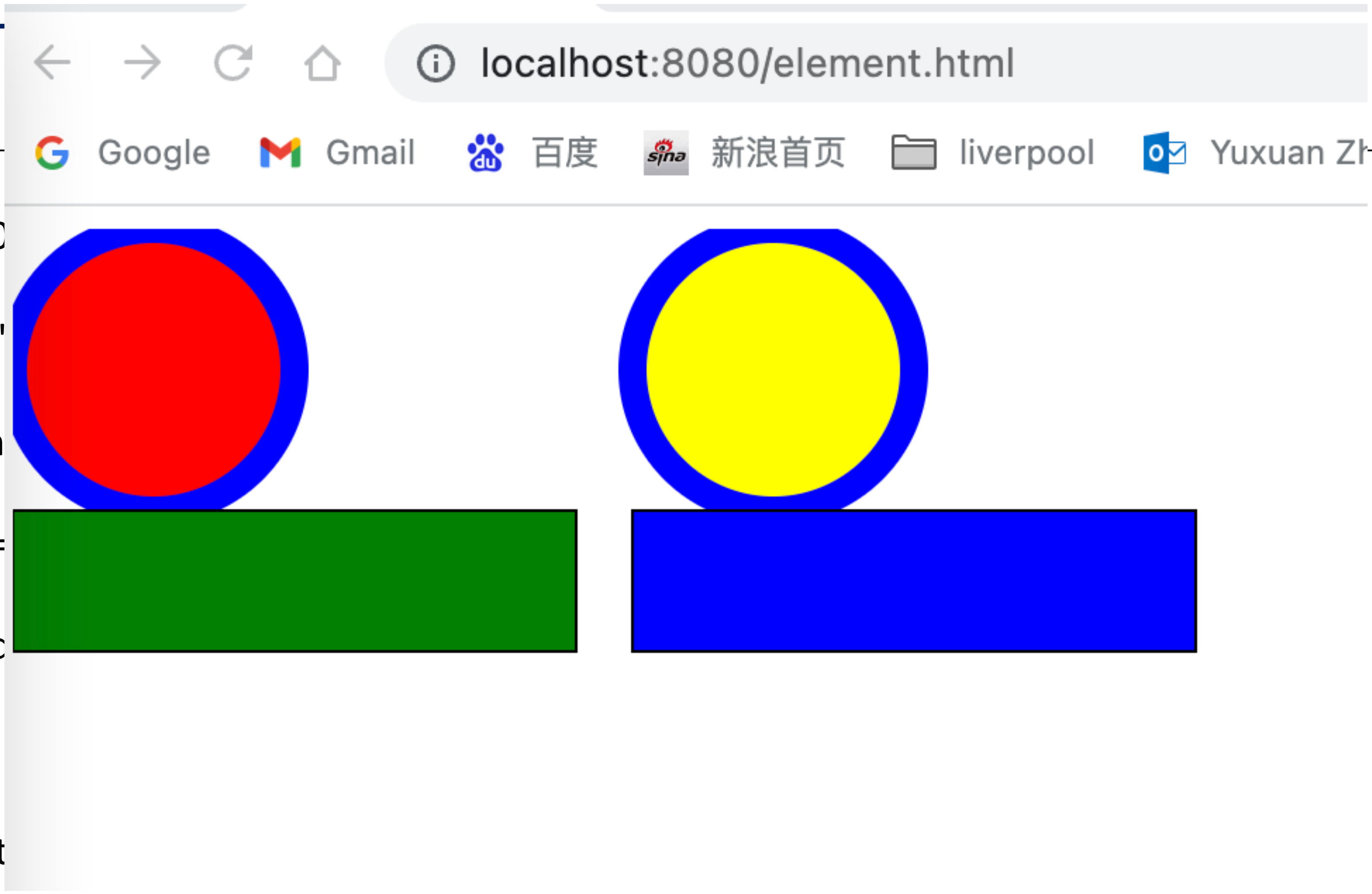
- Select All:

```
<body>
        <svg width = "500" height = "500">
                <circle id = "c1"
                cx = "50" cy = "50" r = "50" stroke = "black" fill = "red"></circle>
                <rect id = "r1"
                x = "0" y = "100" width = "200" height = "50" stroke = "black" fill = "green"></rect>
                <circle id = "c2"
                cx = "270" cy = "50" r = "50" stroke = "black" fill = "yellow"></circle>
                <rect id = "r2"
                x = "220" y = "100" width = "200" height = "50" stroke = "black" fill = "blue"></rect>
        </svg>

        <script>
                d3.selectAll("circle").attr("stroke", "blue").attr("stroke-width","10");
        </script>
</body>
```

D3.js

- Select All:

```
<body>
        <svg width = "500" height = "500
                <circle id = "c1"
                cx = "50" cy = "50" r = '
                <rect id = "r1"
                x = "0" y = "100" width
                <circle id = "c2"
                cx = "270" cy = "50" r =
                <rect id = "r2"
                x = "220" y = "100" wic
        </svg>

        <script>
                d3.selectAll("circle").at
        </script>
</body>
```



D3.js

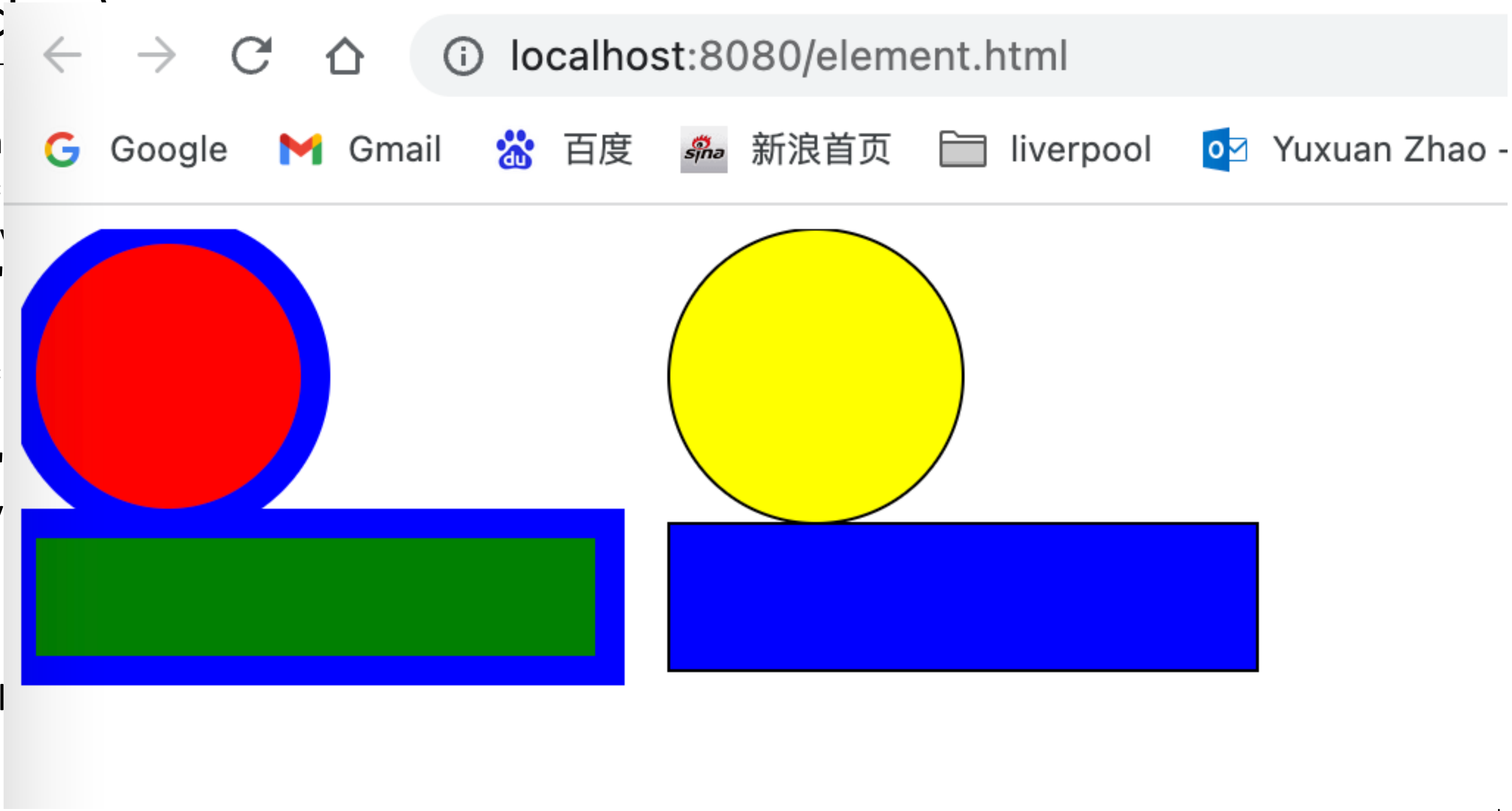
- Select All (by class):

```
<body>
        <svg width = "500" height = "500">
                <circle id = "c1" class = "class1"
                cx = "50" cy = "50" r = "50" stroke = "black" fill = "red"></circle>
                <rect id = "r1" class = "class1"
                x = "0" y = "100" width = "200" height = "50" stroke = "black" fill = "green"></rect>
                <circle id = "c2" class = "class2"
                cx = "270" cy = "50" r = "50" stroke = "black" fill = "yellow"></circle>
                <rect id = "r2" class = "class3"
                x = "220" y = "100" width = "200" height = "50" stroke = "black" fill = "blue"></rect>
        </svg>

        <script>
                d3.selectAll(".class1").attr("stroke", "blue").attr("stroke-width","10");
        </script>
</body>
```

D3.js

- Select All (by c

```
<body>
        <svg width = "500" h
                <circle id =
                cx = "50" c
                <rect id = "
                x = "0" y =
                <circle id =
                cx = "270"
                <rect id = "
                x = "220" y
        </svg>

        <script>
                d3.selectAl
        </script>
</body>
```



D3.js

- Select by from parent identifier: d3.selectAll(“#secondgroup rect”)

```
<svg width="1600" height="800" id="mainsvg" class="svgs">
        <g id="maingroup" transform="translate(100, 100)">
                <circle id="circle1"
                stroke="black" r="66" fill="#4B8E6F" cx="0"></circle>
                <rect id="rect1" class="class2"
                stroke="black" height="200" width="66" fill="#ff8603" x="100" y="-100"></rect>
                <rect id="rect2" class="class1"
                stroke="black" height="200" width="66" fill="#ffde1d" x="200" y="-100"></rect>
                <rect id="rect3" class="class1"
                stroke="black" height="200" width="66" fill="#7289AB" x="300" y="-100"></rect>
                <text id="myText" class="class2"
                stroke="yellow" font-size="2em" x="400" fill="#1e9d95">Hey D3! </text>
        </g>
        <g id="secondgroup" transform="translate(550, 100)">
                <rect id="rect4" class="class2"
                stroke="black" height="200" width="66" fill="#DD6B66" x="100" y="-100"></rect>
                <rect id="rect6" class="class1"
                stroke="black" height="200" width="66" fill="#E69D87" x="300" y="-100"></rect>
        </g>
</svg>
```

D3.js

- DOM Manipulation using D3:

Method	Description
text("content")	Gets or sets the text of the selected element
append("element name")	Adds an element inside the selected element but just before the end of the selected element.
insert("element name")	Inserts a new element in the selected element
remove()	Removes the specified element from the DOM
html("content")	Gets or sets the inner HTML of selected element
attr("name", "value")	Gets or sets an attribute on the selected element.
property("name", "value")	Gets or sets an attribute on the selected element.
style("name", "value")	Gets or sets the style of the selected element
classed("css class", bool)	Gets, adds or removes a css class from the selection

D3.js

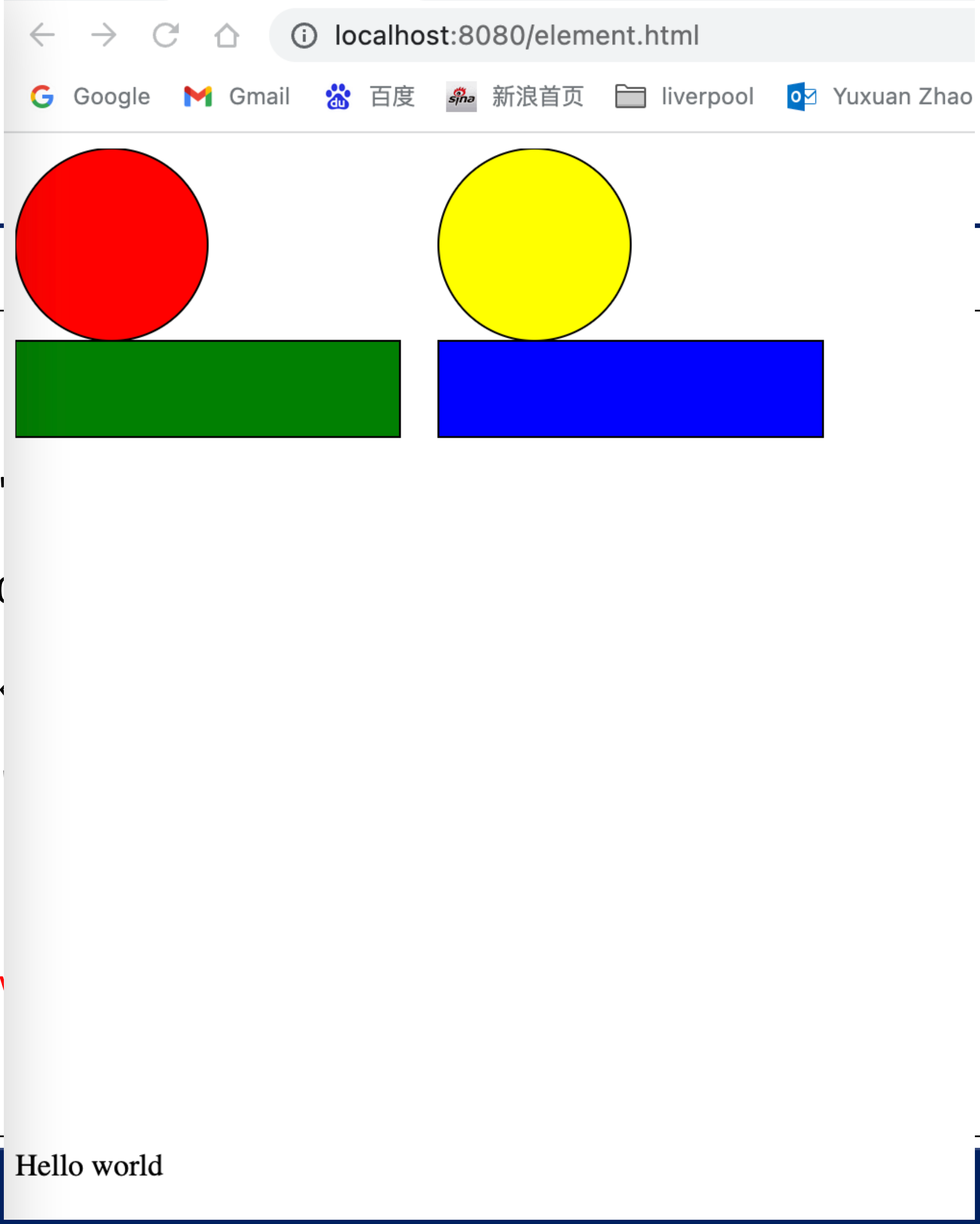
- Append:

```
<body>
        <svg width = "500" height = "500">
                <circle id = "c1" class = "class1"
                cx = "50" cy = "50" r = "50" stroke = "black" fill = "red"></circle>
                <rect id = "r1" class = "class1"
                x = "0" y = "100" width = "200" height = "50" stroke = "black" fill = "green"></rect>
                <circle id = "c2" class = "class2"
                cx = "270" cy = "50" r = "50" stroke = "black" fill = "yellow"></circle>
                <rect id = "r2" class = "class3"
                x = "220" y = "100" width = "200" height = "50" stroke = "black" fill = "blue"></rect>
        </svg>

        <script>
                d3.select("body").append("p").text("Hello world");
        </script>
</body>
```

D3.js

- Append:

```
<body>
        <svg width = "500" height = "500">
                <circle id = "c1" class = "class1"
                cx = "50" cy = "50" r = "50" stroke = "black"
                <rect id = "r1" class = "class1"
                x = "0" y = "100" width = "200" height = "5(
                <circle id = "c2" class = "class2"
                cx = "270" cy = "50" r = "50" stroke = "black
                <rect id = "r2" class = "class3"
                x = "220" y = "100" width = "200" height =
        </svg>

        <script>
                d3.select("body").append("p").text("Hello
        </script>
</body>
```



Hello world

D3.js

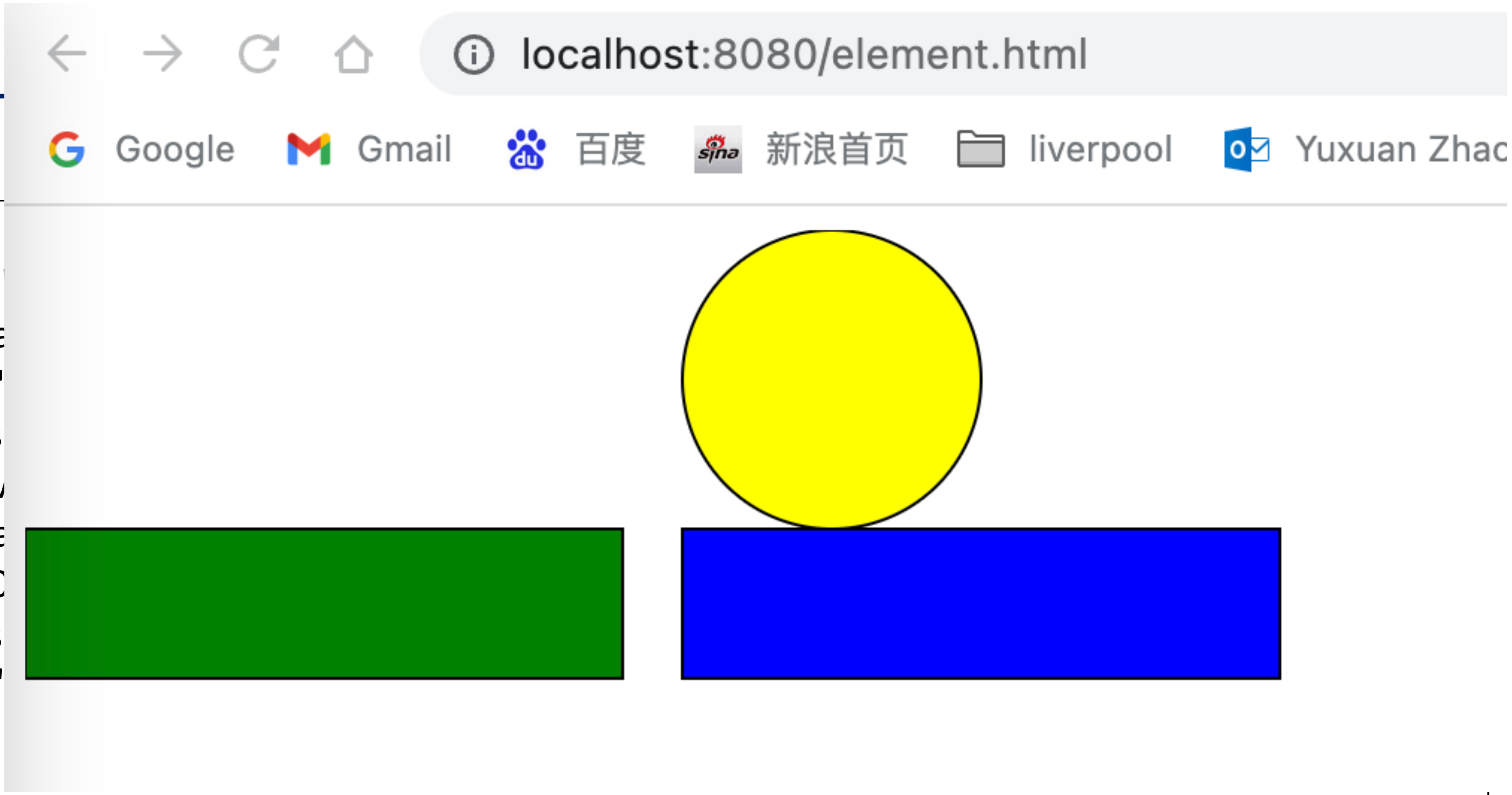
- Remove:

```
<body>
        <svg width = "500" height = "500">
                <circle id = "c1" class = "class1"
                cx = "50" cy = "50" r = "50" stroke = "black" fill = "red"></circle>
                <rect id = "r1" class = "class1"
                x = "0" y = "100" width = "200" height = "50" stroke = "black" fill = "green"></rect>
                <circle id = "c2" class = "class2"
                cx = "270" cy = "50" r = "50" stroke = "black" fill = "yellow"></circle>
                <rect id = "r2" class = "class3"
                x = "220" y = "100" width = "200" height = "50" stroke = "black" fill = "blue"></rect>
        </svg>

        <script>
                d3.select("#c1").remove();
        </script>
</body>
```

D3.js

- Remove:

```
<body>
        <svg width = "500" height = '
                <circle id = "c1" cla
                cx = "50" cy = "50"
                <rect id = "r1" clas
                x = "0" y = "100" w
                <circle id = "c2" cla
                cx = "270" cy = "50
                <rect id = "r2" clas
                x = "220" y = "100'
        </svg>

        <script>
                d3.select("#c1").remove();
        </script>
</body>
```



Summary

```
<!DOCTYPE html>
<html>
        <body>
                <svg >
                        SVG elements
                </svg>
                <script>
                        D3 Codes
                </script>
        </body>
</html>
```

Summary

- Visual Studio Code (VS Code)
- HTML (HyperText Markup Language)
- SVG (Scalable Vector Graphics)
- Java Script
- D3.js