Select

Data Visualization

Select

- D3 Select: grab a hold of elements
  - elements: , <g>, <rect>, <circle>…..

- d3.select()
- d3.selectAll()
- We can select elements by “tag”, “ID”, or “class”
  - d3.select(“tag”)
  - d3.select(“#id”)
  - d3.select(“.class”)

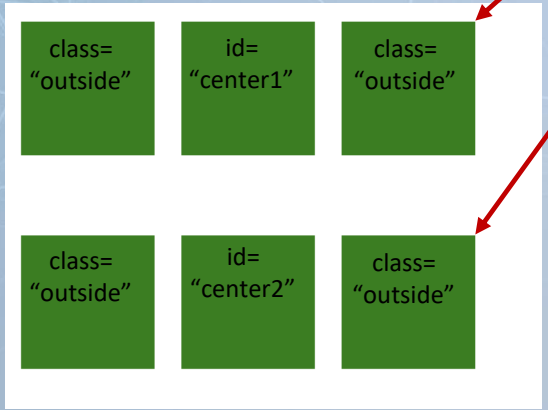
Ex02-1 (Select)

- Demonstration of d3.select() and d3.selectAll()

- Files
  - index.html
  - main.js

Ex02-1 (Select)

- index.html
  - Without selection, all rectangles are green
  - The first row is group1 and the second row is group2



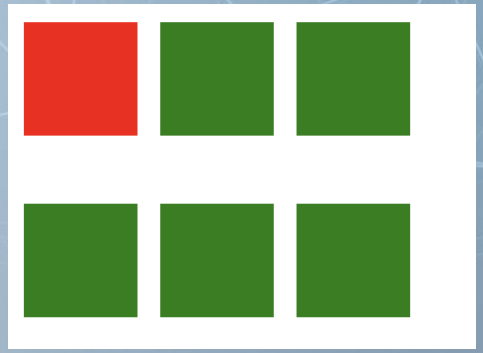

```
<!doctype html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="description" content="">
    <title>The First D3 Example</title>
</head>
<body>
    <svg width="400" height="400">
        <g id="group1">
            <rect class="outside" x="0" y="0" width="50" height="50" fill="green"></rect>
            <rect id="center1" x="50" y="0" width="50" height="50" fill="green"></rect>
            <rect class="outside" x="120" y="0" width="50" height="50" fill="green"></rect>
        </g>
        <g id="group2">
            <rect class="outside" x="0" y="80" width="50" height="50" fill="green"></rect>
            <rect id="center2" x="50" y="80" width="50" height="50" fill="green"></rect>
            <rect class="outside" x="120" y="80" width="50" height="50" fill="green"></rect>
        </g>
    </svg>

    <script src="https://d3js.org/d3.v5.min.js"></script>
    <script src="main.js"></script>
</body>
</html>
```

Ex02-1 (Select)

- main.js

- .attr(“fill”, “red”)
  - fill what we select by red
- d3.select(“rect”)
  - select the first element with tag “rect”

```
d3.select("rect").attr("fill", "red");
// d3.select("#center1").attr("fill", "red");
// d3.select("#center2").attr("fill", "blue");
// d3.select(".outside").attr("fill", "red");
// d3.selectAll("rect").attr("fill", "red");
// d3.selectAll(".outside").attr("fill", "red");
// var select1 = d3.selectAll("g");
// select1.select("rect").attr("fill", "red");
// var select2 = d3.select("#group1");
// select2.selectAll("rect").attr("fill", "red");
```

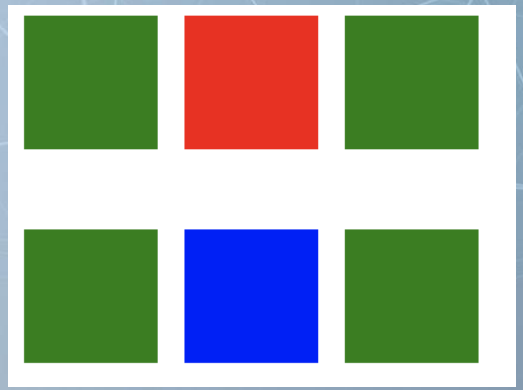


Ex02-1 (Select)

- main.js

- d3.select(“#center1”)
  - select the element with ID “center1”
  - # indicates that the string is an ID
  - Element ID should be unique

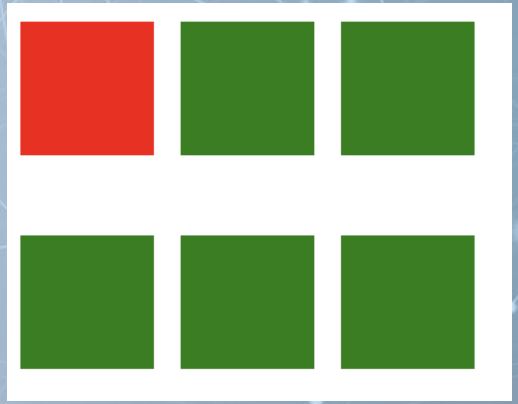
```
// d3.select("rect").attr("fill", "red");
d3.select("#center1").attr("fill", "red");
d3.select("#center2").attr("fill", "blue");
// d3.select(".outside").attr("fill", "red");
// d3.selectAll("rect").attr("fill", "red");
// d3.selectAll(".outside").attr("fill", "red");
// var select1 = d3.selectAll("g");
// select1.select("rect").attr("fill", "red");
// var select2 = d3.select("#group1");
// select2.selectAll("rect").attr("fill", "red");
```



Ex02-1 (Select)

```
// d3.select("rect").attr("fill", "red");
// d3.select("#center1").attr("fill", "red");
// d3.select("#center2").attr("fill", "blue");
d3.select(".outside").attr("fill", "red");
// d3.selectAll("rect").attr("fill", "red");
// d3.selectAll(".outside").attr("fill", "red");
// var select1 = d3.selectAll("g");
// select1.select("rect").attr("fill", "red");
// var select2 = d3.select("#group1");
// select2.selectAll("rect").attr("fill", "red");
```

- main.js
- d3.select(“.outside”)
  - select the first element with class “outside”
  - . indicates that the string is a class
  - Multiple elements could have same class name
- If multiple elements meet the d3.select() condition, it only select the first one

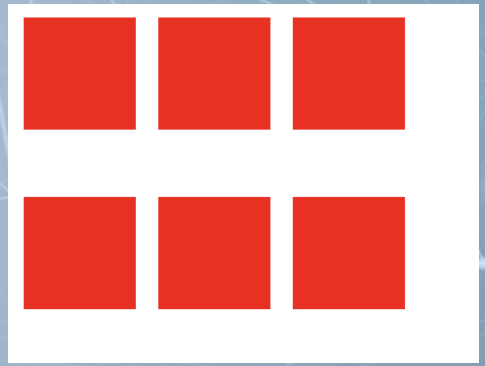


Ex02-1 (Select)

- main.js

- d3.selectAll(“rect”)
  - select all elements with tag rect

```
// d3.select("rect").attr("fill", "red");
// d3.select("#center1").attr("fill", "red");
// d3.select("#center2").attr("fill", "blue");
d3.selectAll("rect").attr("fill", "red");
// d3.selectAll(".outside").attr("fill", "red");
// var select1 = d3.selectAll("g");
// select1.select("rect").attr("fill", "red");
// var select2 = d3.select("#group1");
// select2.selectAll("rect").attr("fill", "red");
```

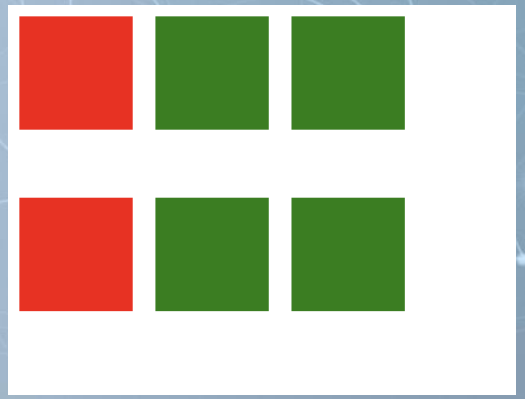


Ex02-1 (Select)

- main.js

```
// d3.select("rect").attr("fill", "red");
// d3.select("#center1").attr("fill", "red");
// d3.select("#center2").attr("fill", "blue");
// d3.select(".outside").attr("fill", "red");
// d3.selectAll("rect").attr("fill", "red");
// d3.selectAll(".outside").attr("fill", "red");
var select1 = d3.selectAll("g");
select1.select("rect").attr("fill", "red");
// var select2 = d3.select("#group1");
// select2.selectAll("rect").attr("fill", "red");
```

- d3.selectAll(“g”)
  - It selects all elements with tag “g”
  - So, “select1” stores the two <g>
- Select1.select(“rect”)
  - Select the first element with tag “rect” from each element in “select1”

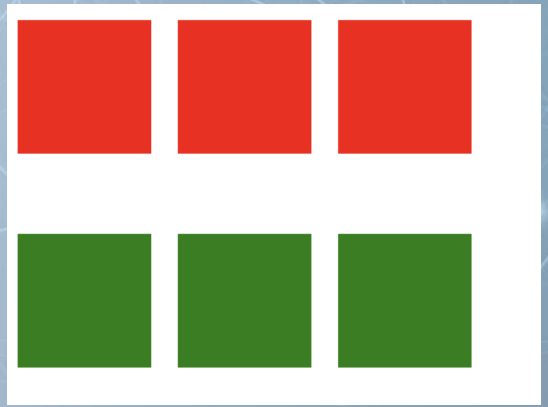


Ex02-1 (Select)

- main.js

- d3.select(“#group1”)
  - Select the element with ID group1
- Select all elements with tag rect from “select2”

```
// d3.select("rect").attr("fill", "red");
// d3.select("#center1").attr("fill", "red");
// d3.select("#center2").attr("fill", "blue");
// d3.select(".outside").attr("fill", "red");
// d3.selectAll("rect").attr("fill", "red");
// d3.selectAll(".outside").attr("fill", "red");
// var select1 = d3.selectAll("g");
// select1.select("rect").attr("fill", "red");
var select2 = d3.select("#group1");
select2.selectAll("rect").attr("fill", "red");
```



Try it

- Try to run Ex02-1
- Make sure you fully understand it

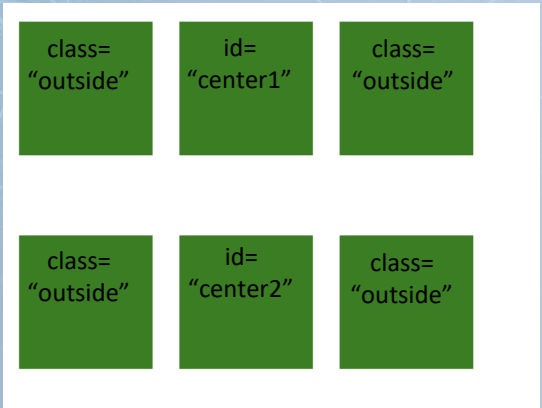
.attr(attrName, value)

- Set the attribute to the specified value on the selected elements
- Attributes

```
<!doctype html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="description" content="">
    <title>The First D3 Example</title>
</head>
<body>
    <svg width="400" height="400">
        <g id="group1">
            <rect class="outsi e" x="0" y="0" width="50" height="50" fill="green"></ ect>
            <rect id="center1" x="60" y="0" width="50" height="50" fill="green"></re t>
            <rect class="outsi e" x="120" y="0" width="50" height="50" fill="green"></rect>
        </g>
        <g id="group2">
            <rect class="outsi e" x="0" y="80" width="50" height="50" fill="green"></rect>
            <rect id="center2" x="60" y="80" width="50" height="50" fill="green"></r ct>
            <rect class="outsi e" x="120" y="80" width="50" height="50" fill="green" ></rect>
        </g>
    </svg>

    <script src="https://d3js.org/d3.v5.min.js"></script>
    <script src="main.js"></script>
</body>
</html>
```

Ex02-2 (.attr())

- Demonstration of .attr()
- Files
  - index.html
  - main.js



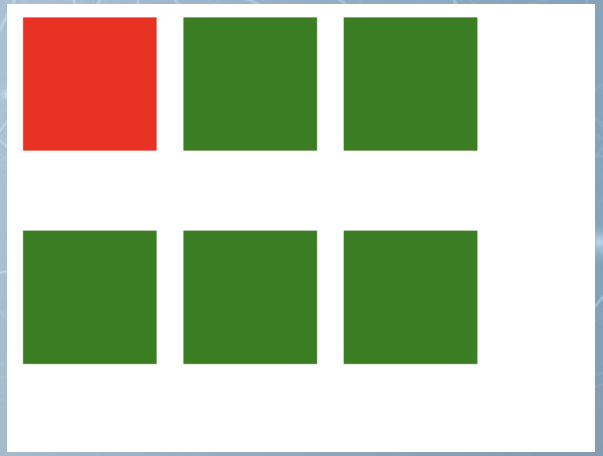

```
<!doctype html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="description" content="">
    <title>The First D3 Example</title>
</head>
<body>
    <svg width="400" height="400">
        <g id="group1">
            <rect class="outside" x="0" y="0" width="50" height="50" fill="green"></rect>
            <rect id="center1" x="60" y="0" width="50" height="50" fill="green"></rect>
            <rect class="outside" x="120" y="0" width="50" height="50" fill="green"></rect>
        </g>
        <g id="group2">
            <rect class="outside" x="0" y="80" width="50" height="50" fill="green"></rect>
            <rect id="center2" x="60" y="80" width="50" height="50" fill="green"></rect>
            <rect class="outside" x="120" y="80" width="50" height="50" fill="green"></rect>
        </g>
    </svg>

    <script src="https://d3js.org/d3.v5.min.js"></script>
    <script src="main.js"></script>
</body>
</html>
```

Ex02-2 (.attr())

- main.js

```
d3.select("rect").attr("fill", "red");
// d3.selectAll(".outside").attr("fill", "red");
// d3.select("rect").attr("y", "150");
// d3.selectAll("rect").attr("width", "25");
```

- d3.select(“rect”) only select one rect

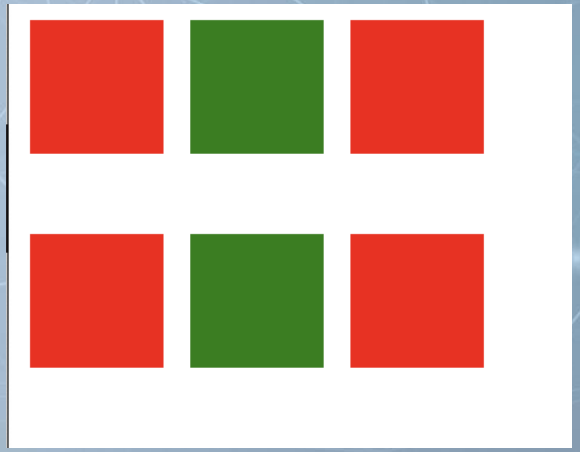


Ex02-2 (.attr())

- main.js

```
// d3.select("rect").attr("fill", "red");
d3.selectAll(".outside").attr("fill", "red");
// d3.select("rect").attr("y", "150");
// d3.selectAll("rect").attr("width", "25");
```

- All elements selected by d3.selectAll(“.outside”) are filled by red

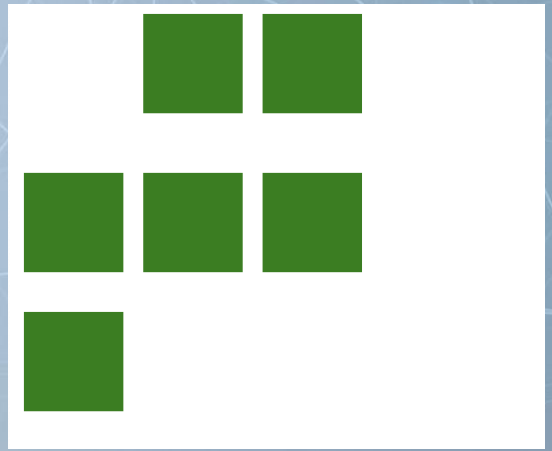


Ex02-2 (.attr())

- main.js

- We can also modify other attributes, such as “y”

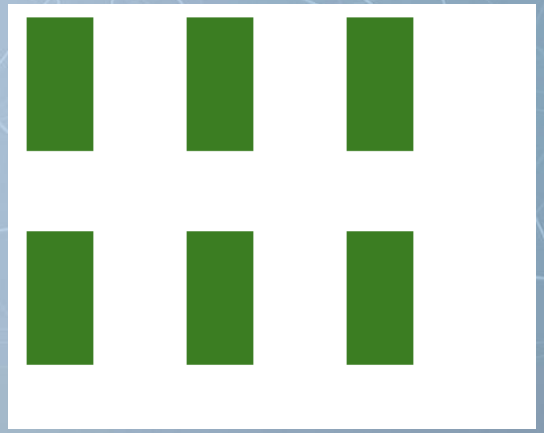
```
// d3.select("rect").attr("fill", "red");
// d3.selectAll(".outside").attr("fill", "red");
d3.select("rect").attr("y", "150");
// d3.selectAll("rect").attr("width", "25");
```



Ex02-2 (.attr())

- main.js
- Or “width”

```
// d3.select("rect").attr("fill", "red");
// d3.selectAll(".outside").attr("fill", "red");
// d3.select("rect").attr("y", "150");
d3.selectAll("rect").attr("width", "25");
```



SVG Elements Reference

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

SVG elements A to Z

A

- <a>
- <animate>
- <animateMotion>
- <animateTransform>

C

- <circle>
- <clipPath>
- <color-profile>

D

- <defs>
- <desc>
- <discard>

E

- <ellipse>

F

- <feBlend>
- <feColorMatrix>
- <feComponentTransfer>
- <feComposite>
- <feConvolveMatrix>
- <feDiffuseLighting>
- <feDisplacementMap>
- <feDistantLight>
- <feDropShadow>
- <feFlood>
- <feFuncA>
- <feFuncB>
- <feFuncG>
- <feFuncR>
- <feGaussianBlur>
- <feImage>
- <feMerge>
- <feMergeNode>
- <feMorphology>
- <feOffset>
- <fePointLight>
- <feSpecularLighting>
- <feSpotLight>
- <feTile>
- <feTurbulence>
- <filter>
- <foreignObject>

G

- <g>

H

- <hatch>
- <hatchpath>

I

- <image>

L

- <line>
- <linearGradient>

M

- <marker>
- <mask>
- <mesh>
- <meshgradient>
- <meshpatch>
- <meshrow>
- <metadata>
- <mpath>

P

- <path>
- <pattern>
- <polygon>
- <polyline>

R

- <radialGradient>
- <rect>

S

- <script>
- <set>
- <solidcolor>
- <stop>
- <style>
- <svg>
- <switch>
- <symbol>

T

- <text>
- <textPath>
- <title>
- <tspan>

U

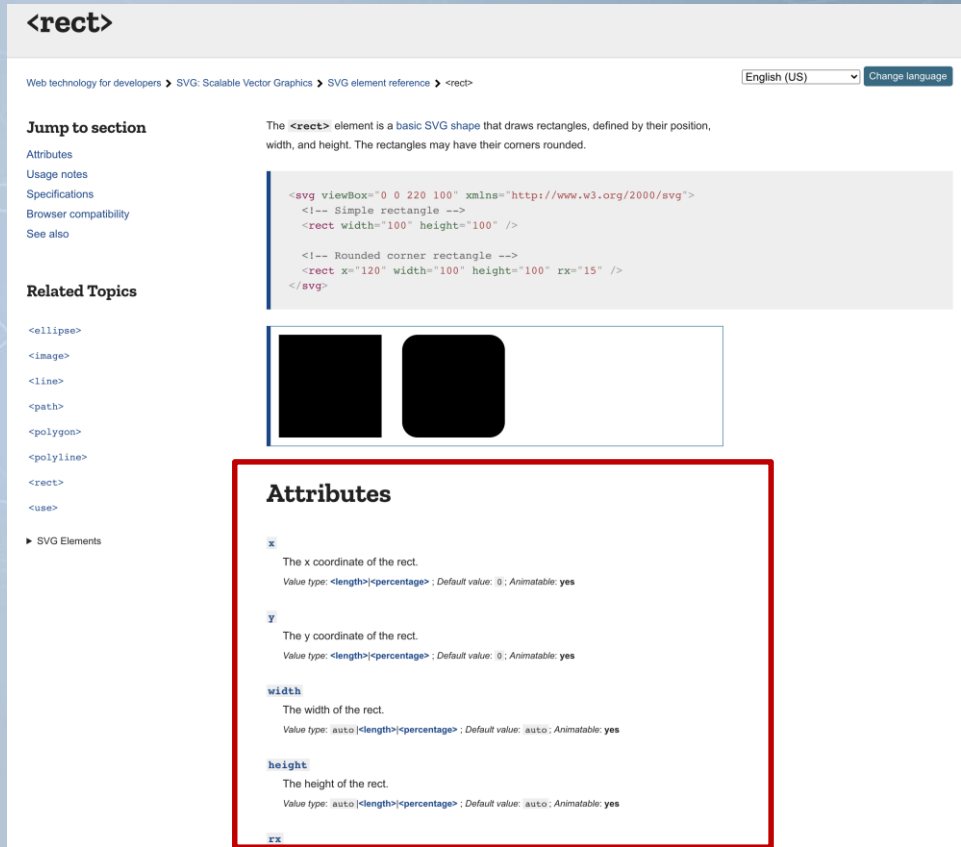
- <unknown>
- <use>

V

- <view>

SVG Element Attributes Reference

- Example <rect>
  - Check what attributes you can use for different elements



<rect>

Web technology for developers › SVG: Scalable Vector Graphics › SVG element reference › <rect>

English (US) Change language

Jump to section

Attributes
Usage notes
Specifications
Browser compatibility
See also

Related Topics

<ellipse>
<image>
<line>
<path>
<polygon>
<polyline>
<rect>
<use>
▸ SVG Elements

The **<rect>** element is a basic SVG shape that draws rectangles, defined by their position, width, and height. The rectangles may have their corners rounded.

```
<svg viewBox="0 0 220 100" xmlns="http://www.w3.org/2000/svg">
  <!-- Simple rectangle -->
  <rect width="100" height="100" />

  <!-- Rounded corner rectangle -->
  <rect x="120" width="100" height="100" rx="15" />
</svg>
```

Attributes

x
The x coordinate of the rect.
Value type: **<length>|<percentage>** ; *Default value:* `0` ; *Animatable:* **yes**

y
The y coordinate of the rect.
Value type: **<length>|<percentage>** ; *Default value:* `0` ; *Animatable:* **yes**

width
The width of the rect.
Value type: `auto` **|<length>|<percentage>** ; *Default value:* `auto` ; *Animatable:* **yes**

height
The height of the rect.
Value type: `auto` **|<length>|<percentage>** ; *Default value:* `auto` ; *Animatable:* **yes**

rx

Try it

- Try to run Ex02-2
- Check the attribute webpage
- Make sure you fully understand it

.classed(className, value)

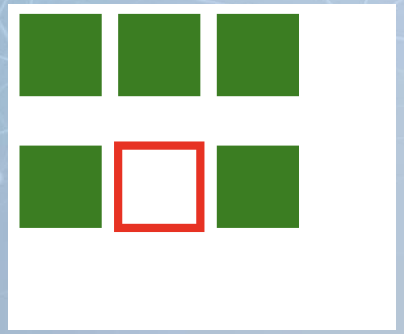
- If we define a CSS style
  - Ex:

```
<head>
    <meta charset="utf-8">
    <meta name="description" content="">
    <title>The First D3 Example</title>
    <style>
        .hollow_rect{
            fill: white;
            stroke: red;
            stroke-width: 5px;
        }
    </style>
</head>
```

- We can apply it to or remove it from an element by selection.classed(className, value)
  - ”value” is either true or false
  - true: applied it
  - false: remove it

Ex02-3 (.classed)

- Files
  - index.html
  - main.js
- Without main.js, you will see this

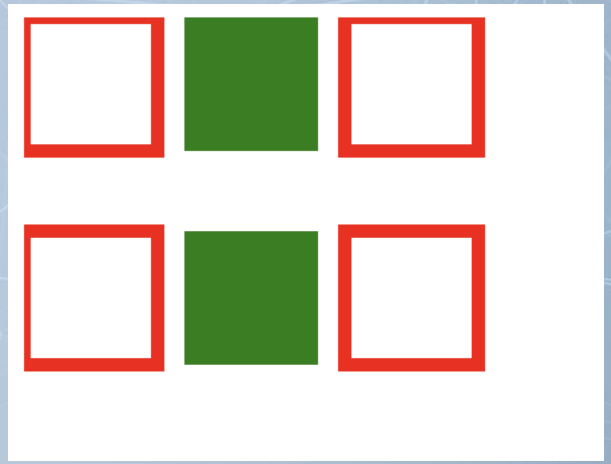


```
<!doctype html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="description" content="">
    <title>The First D3 Example</title>
    <style>
        .hollow_rect{
            fill: white;
            stroke: red;
            stroke-width: 5px;
        }
    </style>
</head>
<body>
    <svg width="400" height="400">
        <g id="group1">
            <rect class="outside" x="0" y="0" width="50" height="50" fill="green"></rect>
            <rect id="center1" x="60" y="0" width="50" height="50" fill="green"></rect>
            <rect class="outside" x="120" y="0" width="50" height="50" fill="green"></rect>
        </g>
        <g id="group2">
            <rect class="outside" x="0" y="80" width="50" height="50" fill="green"></rect>
            <rect class="hollow_rect" id="center2" x="60" y="80" width="50" height="50" fill="green"></rect>
            <rect class="outside" x="120" y="80" width="50" height="50" fill="green"></rect>
        </g>
    </svg>

    <script src="https://d3js.org/d3.v5.min.js"></script>
    <script src="main.js"></script>
</body>
</html>
```

Ex02-3 (.classed)

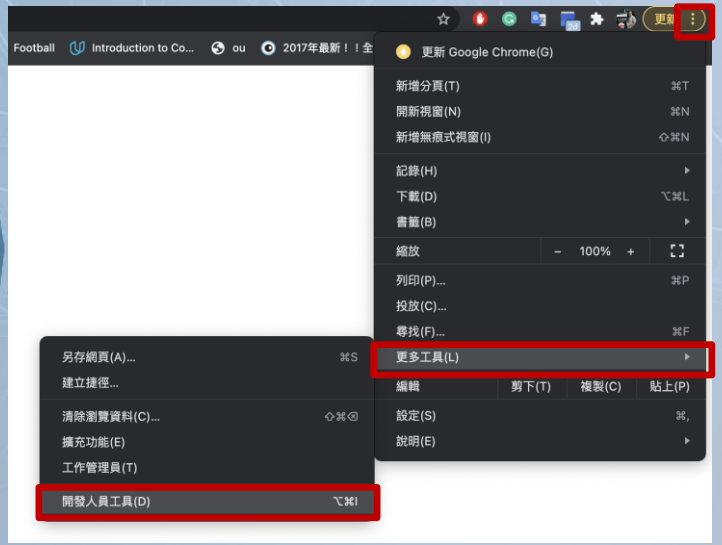
- main.js

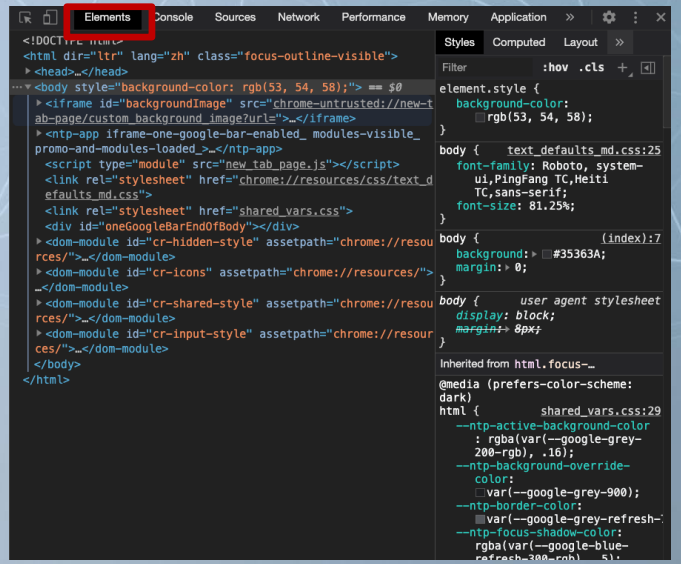
```
d3.selectAll(".outside").classed("hollow_rect", true);
d3.select("#center2").classed("hollow_rect", false);
```



Check DOM

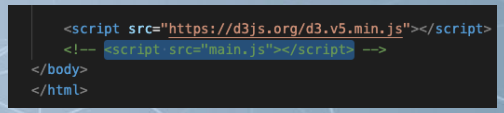
- Now, everything is short and easy.
- If your program become complicated, how to check your code works well?
  - Check DEM in “Elements” tab in developer tool





Try it

- Use Ex02-3
  - comment this line in index.html
  - Load the page check the Element tab in the developer tool



  - Uncomment the same line in index.html
  - Reload the page check the Element tab in the developer tool again

- Check the differences?

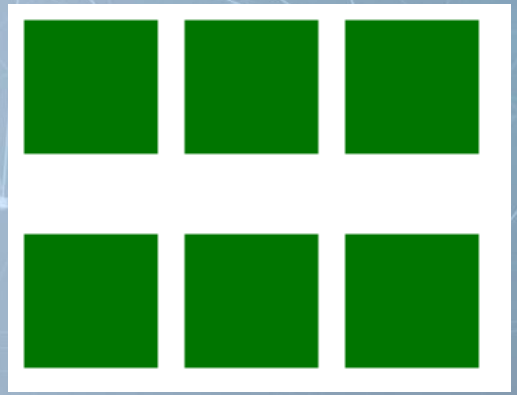
append(tagName)

- Append a new element as the last child of each selected element

Ex02-4 (.append)

- index.html

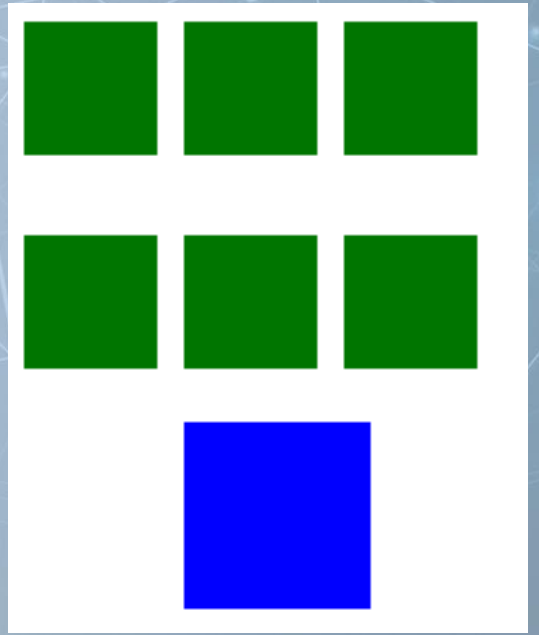
```
<!doctype html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="description" content="">
    <title>The First D3 Example</title>
</head>
<body>
    <svg width="400" height="400">
        <g id="group1">
            <rect class="outside" x="0" y="0" width="50" height="50" fill="green"></rect>
            <rect id="center1" x="60" y="0" width="50" height="50" fill="green"></rect>
            <rect class="outside" x="120" y="0" width="50" height="50" fill="green"></rect>
        </g>
        <g id="group2">
            <rect class="outside" x="0" y="80" width="50" height="50" fill="green"></rect>
            <rect id="center2" x="60" y="80" width="50" height="50" fill="green"></rect>
            <rect class="outside" x="120" y="80" width="50" height="50" fill="green"></rect>
        </g>
    </svg>

    <script src="https://d3js.org/d3.v5.min.js"></script>
    <script src="main.js"></script>
</body>
</html>
```



Ex02-4 (.append)

- main.js

```
d3.selectAll("#group2")
  .append("rect")
  .attr("x", "60")
  .attr("y", "150")
  .attr("width", "70")
  .attr("height", "70")
  .attr("fill", "blue");
```



Try it

- Run Ex02-4 and check what happen in DOM

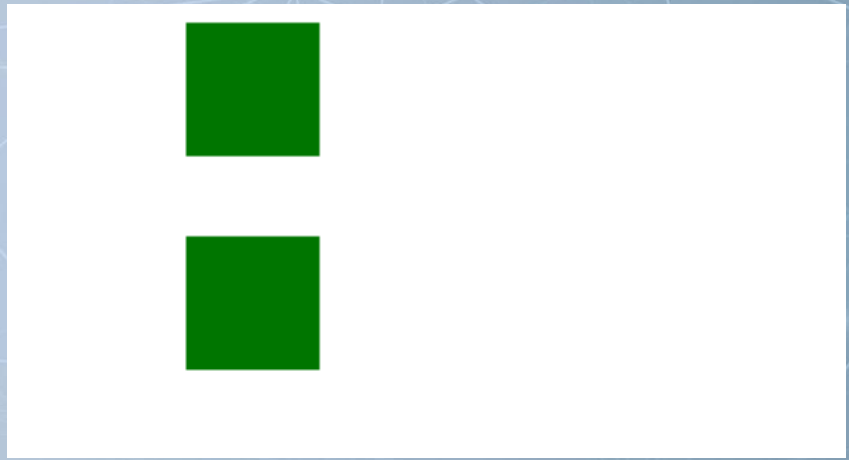
.remove()

- Removes the selected elements from the document

Ex02-5 (.append)

- main.js

```
d3.selectAll(".outside").remove();
```



Try it

- Run Ex02-5 and check what happens in DOM

.text(value)

- Sets the text content to the specified value on all selected elements
  - Replacing any existing child elements

Method Chaining

- Ex02-5: main.js
  - The same
  - .append(“rect”) returns the rect obj
  - rect.attr() also return the rect obj

```
d3.selectAll("#group2")
  .append("rect")
  .attr("x", "60")
  .attr("y", "150")
  .attr("width", "70")
  .attr("height", "70")
  .attr("fill", "blue");
```

```
var rect = d3.selectAll("#group2").append("rect");
rect.attr("x", "60");
rect.attr("y", "150")
rect.attr("width", "70")
rect.attr("height", "70")
rect.attr("fill", "blue");
```

Method Chaining

- To know how to chain methods, we have to know what a method returns
- Check the document and d3 API
- In practice, we usually learn what methods can be chained together by experiences and from examples (at least, that is how I learn)