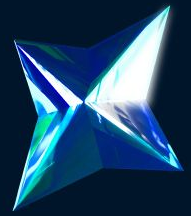
BRO Academy

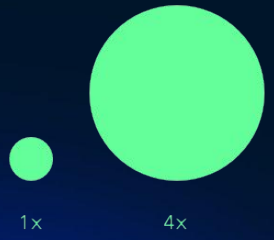


SVG inside

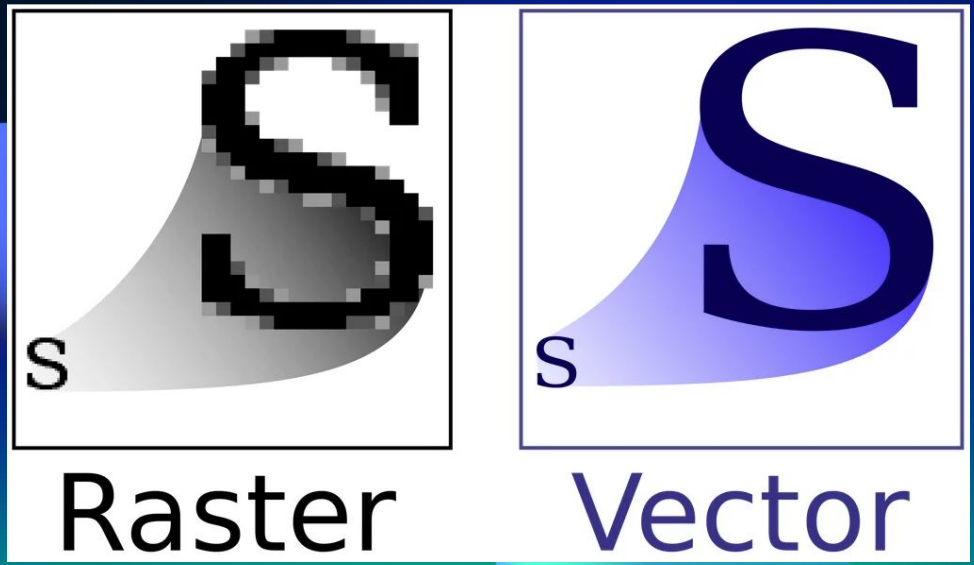
HTML course: Lesson 14

Repeat

Vector images



Created using **mathematical formulas** and geometric **shapes**, such as **lines and curves**



Repeat

Raster vs Vector

example →

	`.jpeg, .png, .avif, .webp`	`<svg>`
Scales without quality loss	❌	✅
Usually has smaller size	❌	✅
How can be changed?	In special programs for graphics	In text editor or IDE
Usage for	complex images	logos, icons, decor
Parts inside can be affected through css and js	❌	✅
Can be converted to	other raster formats only	vector and raster formats

Lesson Plan

1
What is it **inside**

2
How to insert on a page

3
SVG container

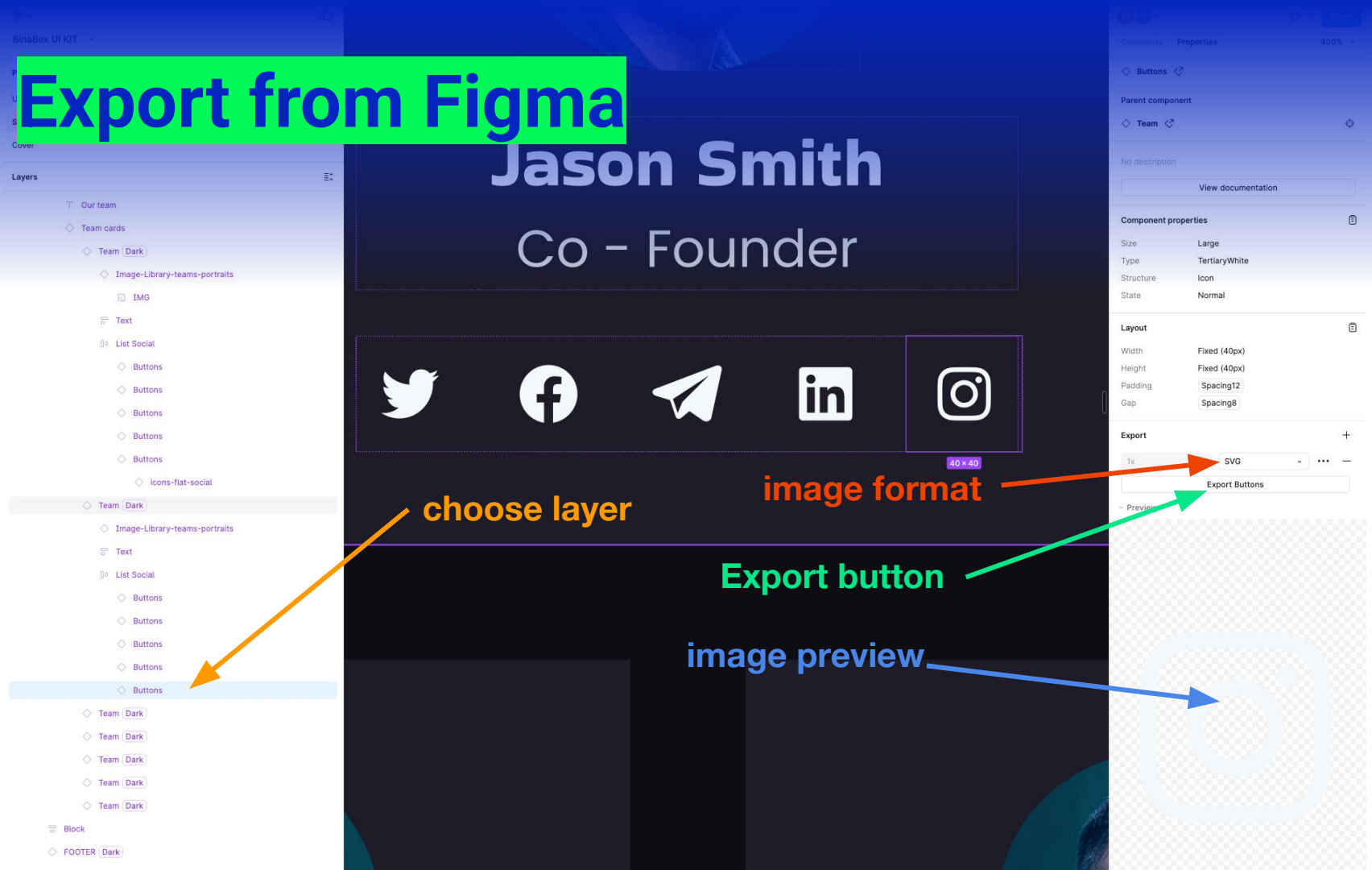
4
SVG tags

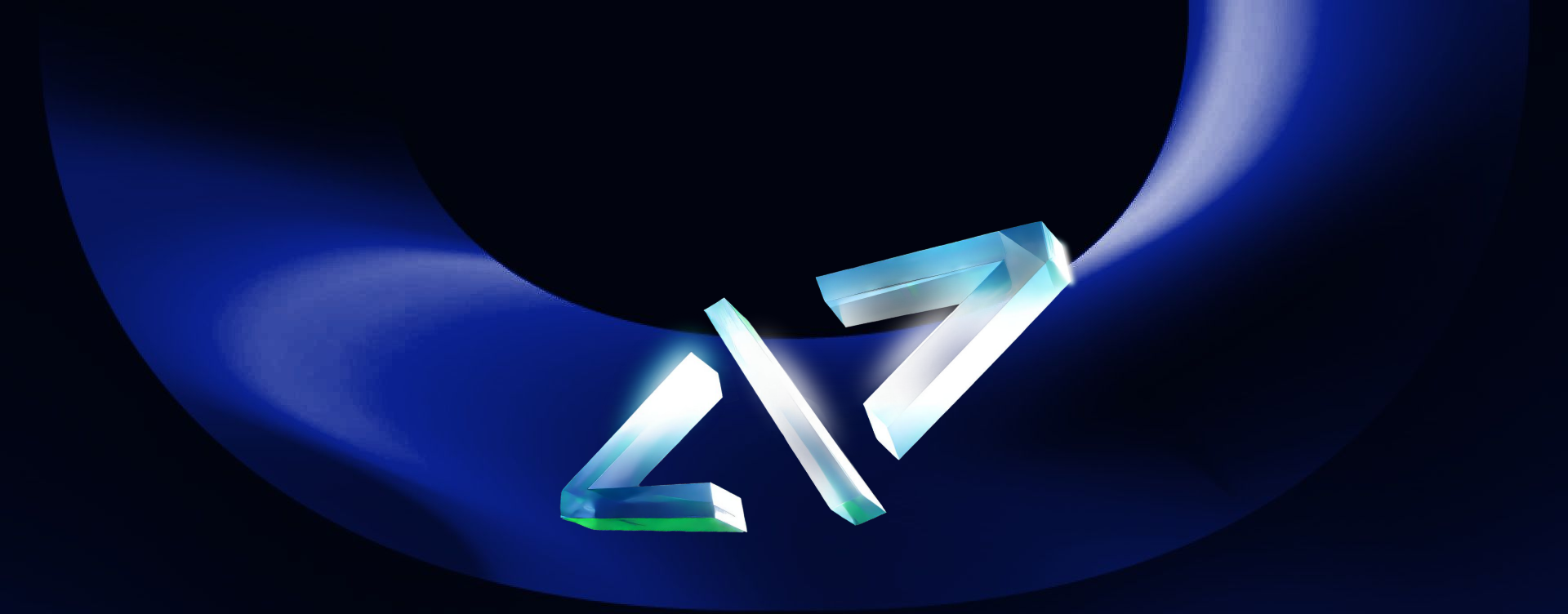
Scalable Vector Graphics

A markup language extended from XML for describing two-dimensional vector graphics

SVG files can be created and edited with any text editor and drawing programs

Export from Figma



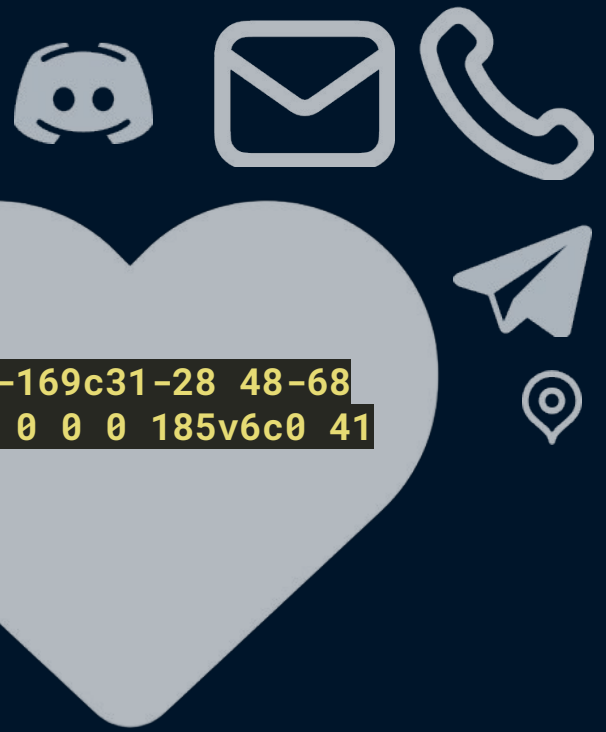


Ways to insert SVG on a page

You have an <SVG>

```
<svg viewBox="0 0 512 512"
xmlns="http://www.w3.org/2000/svg">
    <path d="m48 300 180 169a41 41
0 0 0 56 0l180-169c31-28 48-68
48-109v-6A143 143 0 0 0 268 84l-12
12-12-12A143 143 0 0 0 0 185v6c0 41
17 81 48 109z"/>
</svg>
```

Insert SVG content right into HTML



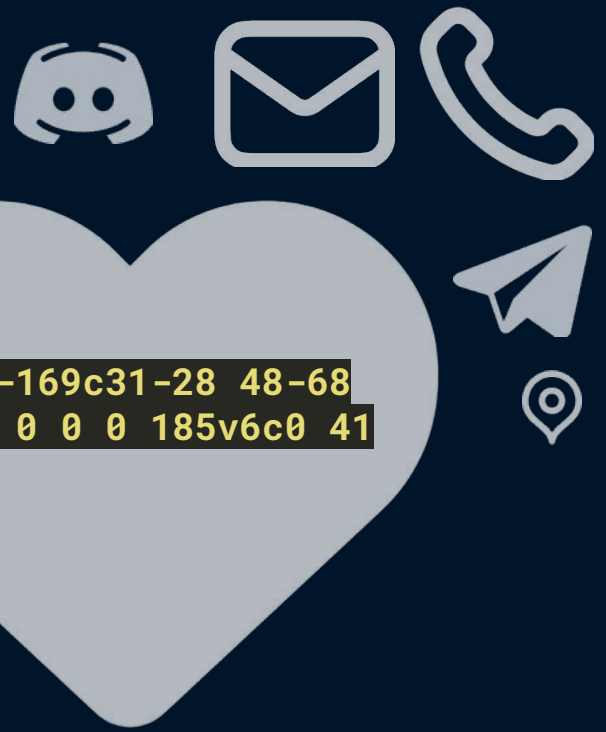
```
<a href="/" class="logo">
    <svg viewBox="0 0 512 512">
        <path d="m48 300 180 169a41 41 0 0 0 56 0l180-169c31-28 48-68
48-109v-6A143 143 0 0 0 268 84l-12 12-12-12A143 143 0 0 0 185v6c0 41
17 81 48 109z"/>
    </svg>
</a>
```

👆don’t use xmlns on svg tag

(`xmlns="http://www.w3.org/2000/svg"`)

[example →]

Insert SVG content right into HTML



```
<a href="/" class="logo">
    <svg viewBox="0 0 512 512" class="logo__icon">
        <path d="m48 300 180 169a41 41 0 0 0 56 0l180-169c31-28 48-68
48-109v-6A143 143 0 0 0 268 84l-12 12-12-12A143 143 0 0 0 185v6c0 41
17 81 48 109z"/>
    </svg>
</a>
```

```
.logo__icon {
    fill: red;
}
```

👆don’t use xmlns on svg tag

(`xmlns="http://www.w3.org/2000/svg"`)

[example →]

Repeat

<picture>

example →

```
<img src="/icons/lamborghini.svg"
alt="lamborghini" load="lazy">
```

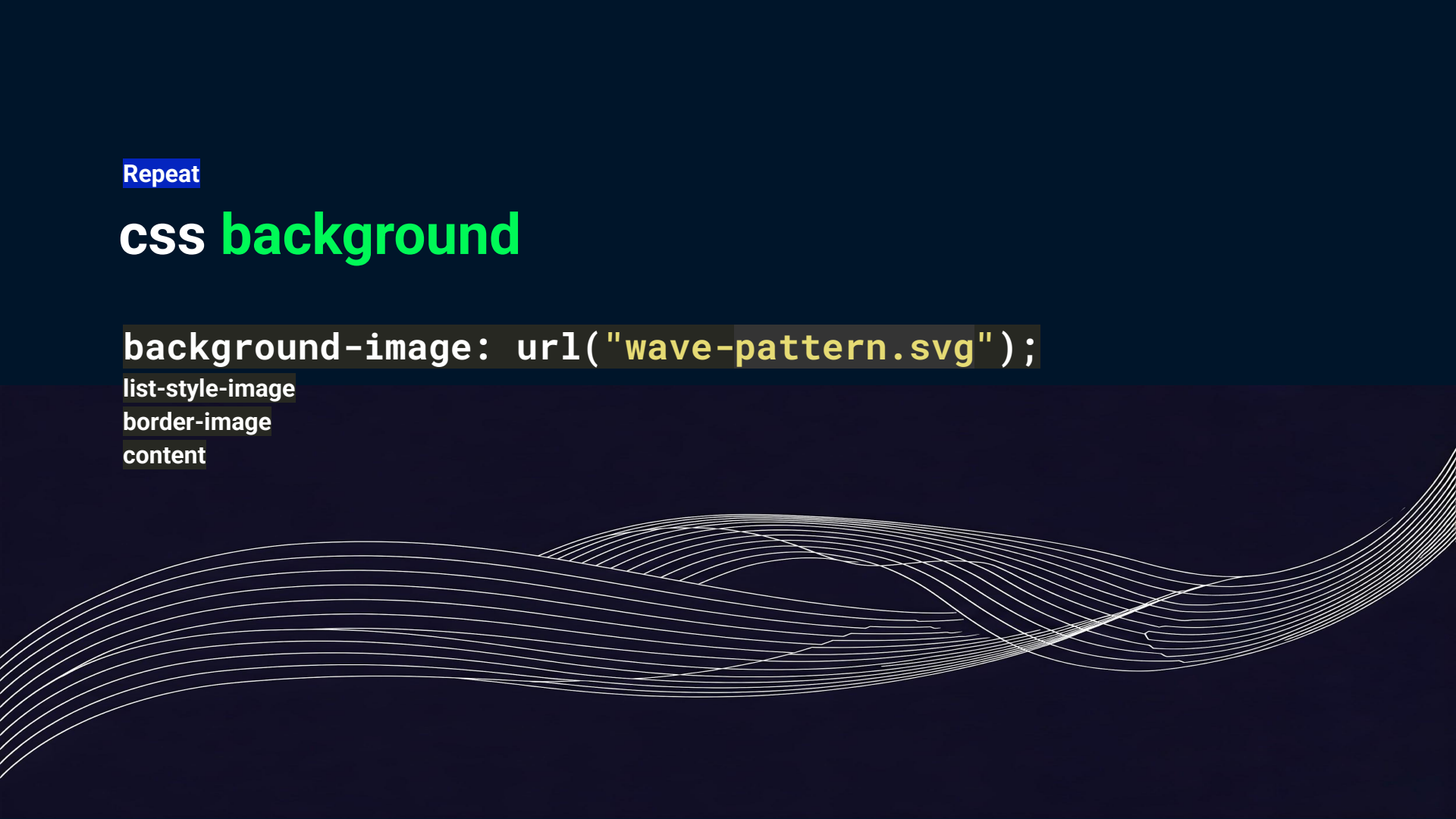
👆 don’t forget to use lazy load if image not in first viewport

Repeat

css background

```
background-image: url("wave-pattern.svg");
```

list-style-image
border-image
content



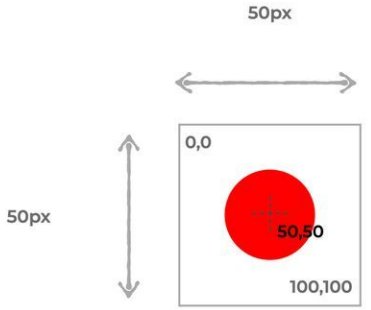
SVG container

viewBox and dimensions

– Defines the aspect ratio
– Responsible for scaling

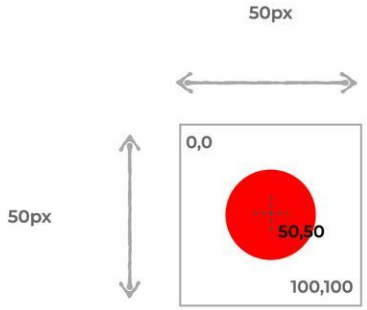
```
viewBox="0 0 512 512"
```

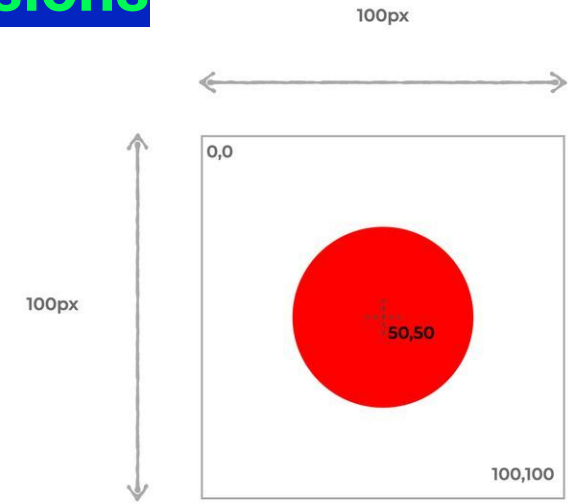
- `x` – Horizontal coordinate
- `y` – Vertical coordinate
- `width`
- `height`

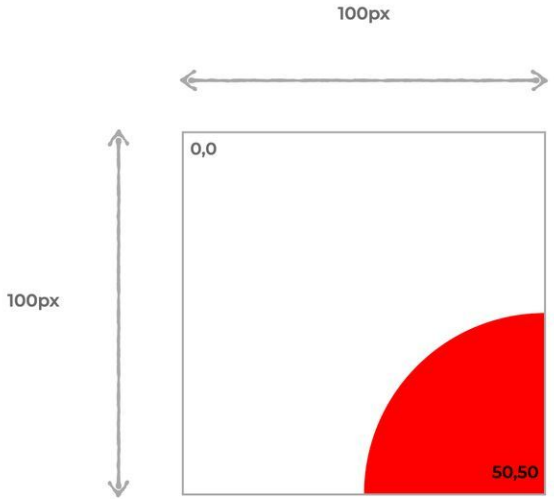



```
<svg
  width="50" height="50"
  viewBox="0 0 100 100"
>
  <circle
    cx="50" cy="50"
    r="25" fill="red"
  />
</svg>
```

viewBox + dimensions








```
<svg
  width="50" height="50"
  viewBox="0 0 100 100"
>
  <circle
    cx="50" cy="50"
    r="25" fill="red"
  />
</svg>
```

```
<svg
  width="100" height="100"
  viewBox="0 0 100 100"
>
  <circle
    cx="50" cy="50"
    r="25" fill="red"
  />
</svg>
```

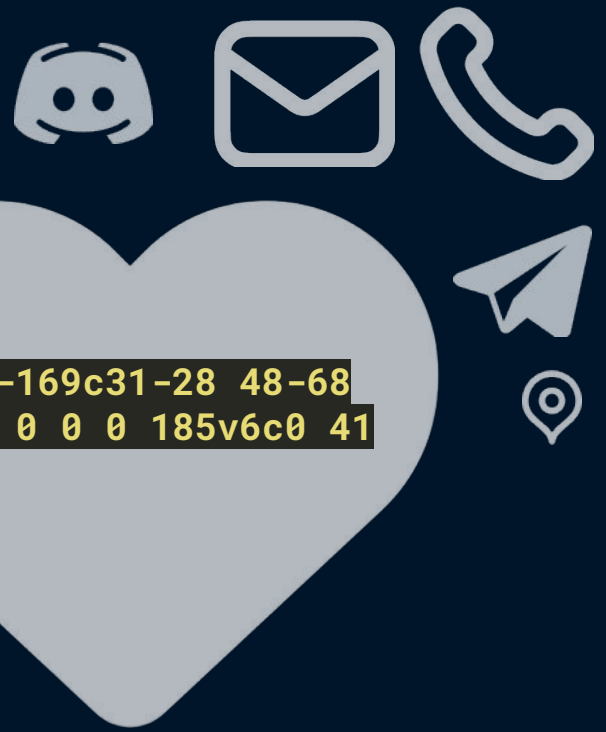
```
<svg
  width="100" height="100"
  viewBox="0 0 50 50"
>
  <circle
    cx="50" cy="50"
    r="25" fill="red"
  />
</svg>
```

Common attributes

- fill – background color
- stroke – border color
- stroke-width - border width

- Clipping, Masking and Compositing properties:
  - 'clip-path'
  - 'clip-rule'
  - 'mask'
  - 'opacity'
- Filter Effects properties:
  - 'enable-background'
  - 'filter'
  - 'flood-color'
  - 'flood-opacity'
  - 'lighting-color'
- Gradient properties:
  - 'stop-color'
  - 'stop-opacity'
- Interactivity properties:
  - 'pointer-events'
- Color and Painting properties:
  - 'color-interpolation'
  - 'color-rendering'
  - 'fill'
  - 'fill-opacity'
  - 'fill-rule'
  - 'image-rendering'
  - 'marker'
  - 'marker-end'
  - 'marker-mid'
  - 'marker-start'
  - 'shape-rendering'
  - 'stroke'
  - 'stroke-dasharray'
  - 'stroke-dashoffset'
  - 'stroke-linecap'
  - 'stroke-linejoin'
  - 'stroke-miterlimit'
  - 'stroke-opacity'
  - 'stroke-width'
  - 'text-rendering'
- Text properties:
  - 'alignment-baseline'
  - 'baseline-shift'
  - 'dominant-baseline'
  - 'glyph-orientation-horizontal'
  - 'glyph-orientation-vertical'
  - 'text-anchor'
  - 'writing-mode'

Insert SVG content right into HTML



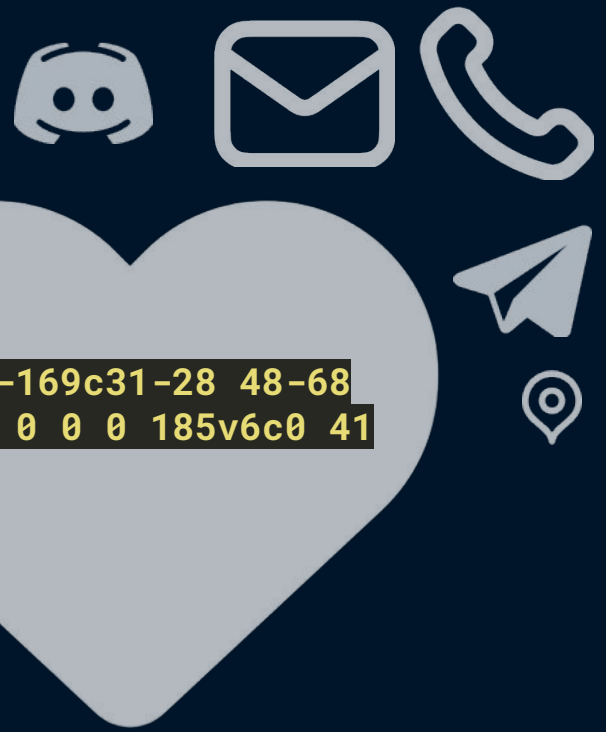
```
<a href="/" class="logo">
    <svg viewBox="0 0 512 512">
        <path d="m48 300 180 169a41 41 0 0 0 56 0l180-169c31-28 48-68
48-109v-6A143 143 0 0 0 268 84l-12 12-12-12A143 143 0 0 0 185v6c0 41
17 81 48 109z"/>
    </svg>
</a>
```

👆don’t use xmlns on svg tag

(`xmlns="http://www.w3.org/2000/svg"`)

[example →]

Insert SVG content right into HTML



```
<a href="/" class="logo">
    <svg viewBox="0 0 512 512" class="logo__icon">
        <path d="m48 300 180 169a41 41 0 0 0 56 0l180-169c31-28 48-68
48-109v-6A143 143 0 0 0 268 84l-12 12-12-12A143 143 0 0 0 185v6c0 41
17 81 48 109z"/>
    </svg>
</a>
```

```
.logo__icon {
    fill: red;
}
```

👆don’t use xmlns on svg tag

(`xmlns="http://www.w3.org/2000/svg"`)

[example →]

Check Binabox design

Break time

SVG tags

Anatomy of SVG

```
<div class="h-screen flex justify-center items-center">
  <svg
    xmlns="http://www.w3.org/2000/svg"
    fill="none"
    viewBox="0 0 24 24"
    stroke-width="1"
    stroke="rebeccapurple"
    class="h-72 w-72"
  >
    <path
      stroke-linecap="round"
      stroke-linejoin="round"
      d="M21 8.25c0-2.485-2.099-4.5-4.688-4.5-1.935 0-3.597
      1.126-4.312 2.733-.715-1.607-2.377-2.733-4.313-2.733C5.
      1 3.75 3 5.765 3 8.25c0 7.22 9 12 9 12s9-4.78 9-12z"
    />
  </svg>
</div>
```

<circle>
<rect>
<path>
<ellipse>
<polygon>
<text>

Grouping container – <g>
Hidden element – <defs>
Reuse an element – <use>

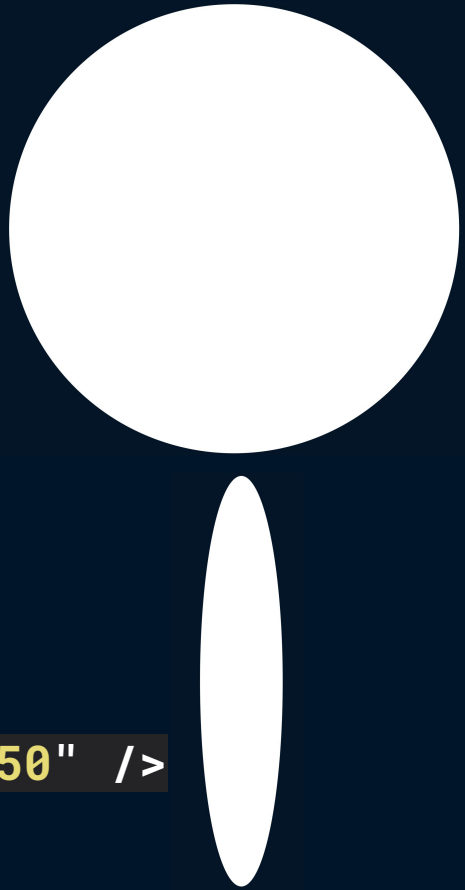
Layering of shapes goes from top to bottom in the code; the last element is most visible.

circle

```
<svg viewBox="0 0 512 512">
  <circle cx="250" cy="250" r="250" />
</svg>
```

ellipse

```
<svg viewBox="0 0 512 512">
  <ellipse cx="250" cy="250" rx="50" ry="250" />
</svg>
```

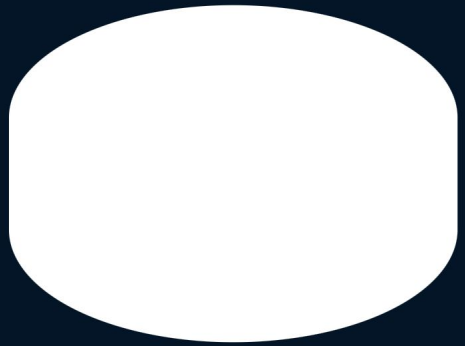


rect

```
<svg viewBox="0 0 512 512">
  <rect x="50" y="50" width="250" height="250" />
</svg>
```

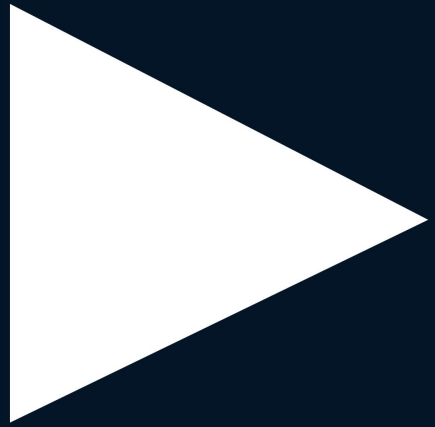
rect with rounded corners

```
<svg viewBox="0 0 512 512">
    <rect x="50" y="50" width="250"
height="150" rx="200" ry="50" />
</svg>
```



polygon for closed shapes

```
<svg viewBox="0 0 512 512">
  <polygon points="0,0 512,256, 0,512" />
</svg>
```



path for custom shapes

```
<svg viewBox="0 0 100 100">
  <path d="m 48 300 180 169
           a 41 41 0 0 0 56 0
           l 180-169c31-28 48-68 48-109
           v-6
           A 143 143 0 0 0 268 84
           l-12 12-12-12
           A 143 143 0 0 0 0 185
           v 6
           c 0 41 17 81 48 109 z" />
</svg>
```

👆 see MDN to get how d attribute works

MoveTo: M, m
LineTo: L, l, H, h, V, v
Cubic Bézier curve: C, c, S, s
Quadratic Bézier curve: Q, q, T, t
Elliptical arc curve: A, a
ClosePath: Z, z

g for grouping

```
<svg viewBox="0 0 100 100">
    <g fill="white" stroke="green" stroke-width="5">
        <circle cx="40" cy="40" r="25" />
        <circle cx="60" cy="60" r="25" />
    </g>
</svg>
```

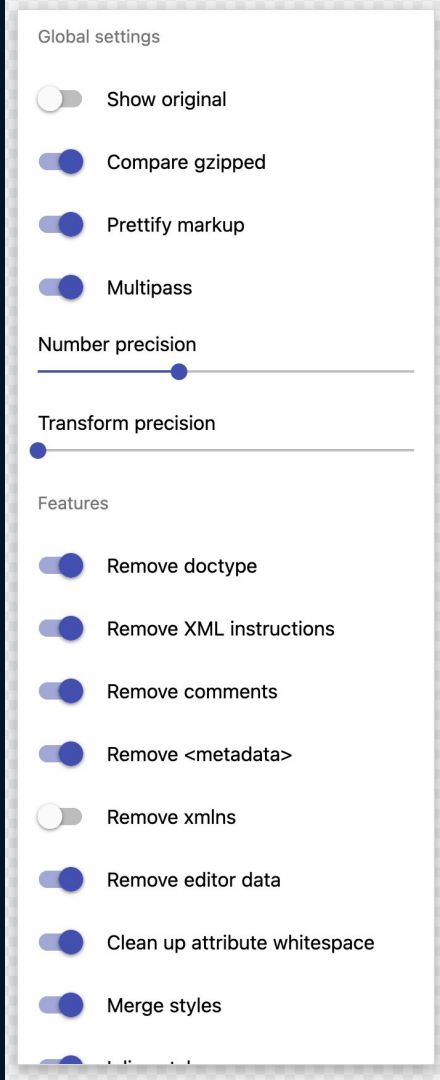
Practice: markup youtube logo SVG

Bonus tip

How to clean up the file

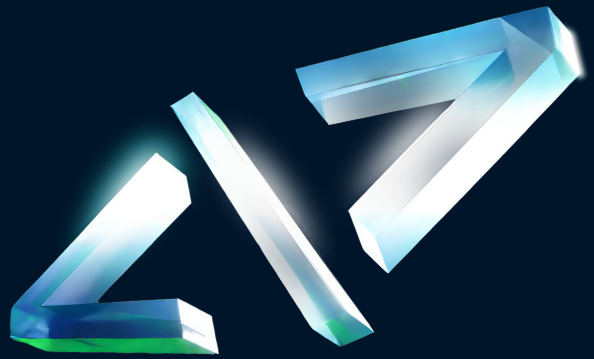
How to remove all waste extra markup after export

jakearchibald.github.io/SVGOMG

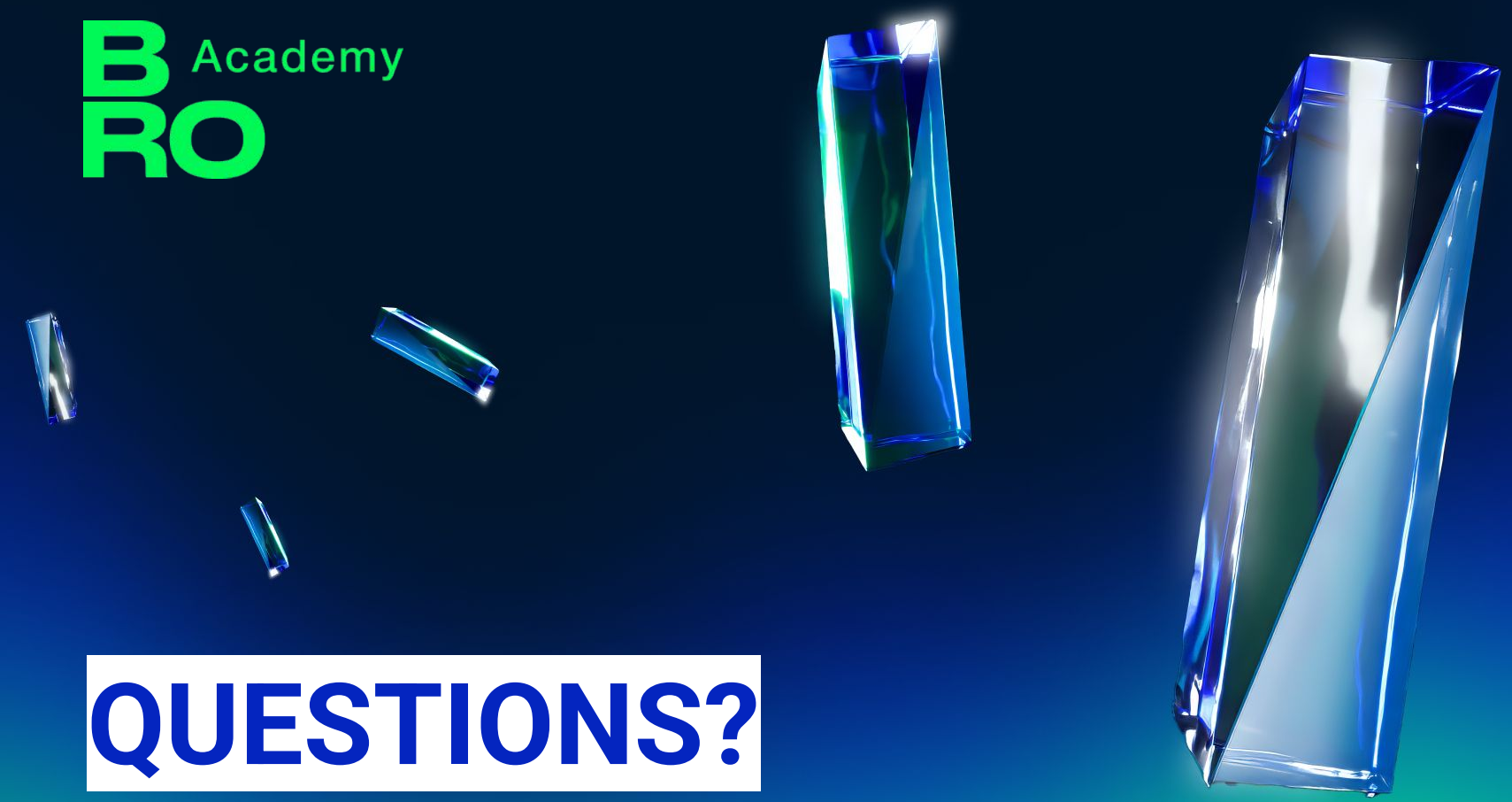


Homework

1. Put all sprite vector images into the src/images/sprite folder.
2. Use them on pages from sprite by #id.
3. Implement color changes on hover and focus using only CSS for icons that need interactive effects.



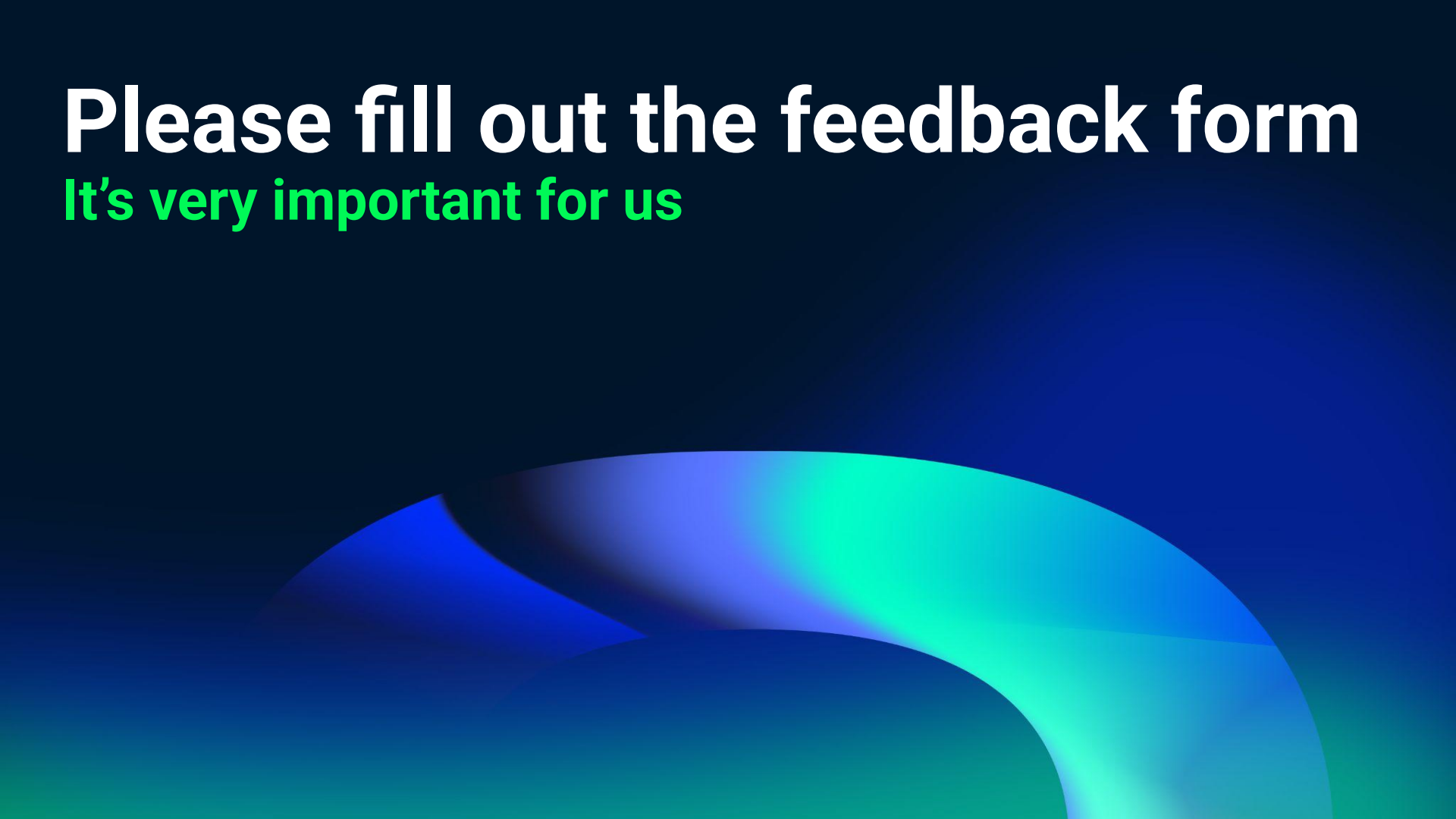
BRO Academy



QUESTIONS?

Please fill out the feedback form

It’s very important for us




THANK YOU!
Have a good evening!