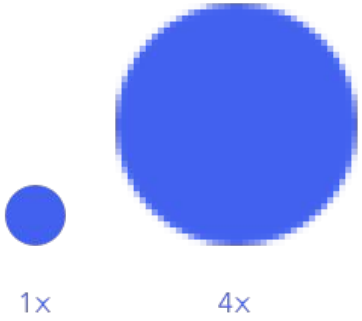
BRO Academy

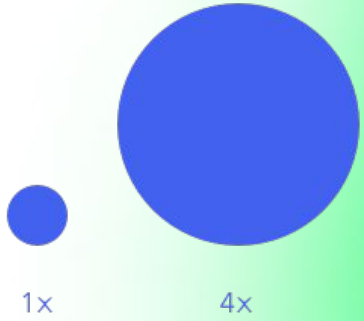
Manual

SVG

VS

Raster	Vector
Consist of pixels	Consist of mathematical formulas, shapes, and lines
Large file size	Very small file size
Scales with quality loss	Looks the same at any size
Can only be converted to other raster formats	Can be converted to raster images
Cannot be modified easily	Can be easily modified (e.g., splitting into components, animating)
Ideal for photographs	Ideal for icons
Uses RGB color space	





Vector

images

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

```
.block {
      background-image: url(image.svg);
      list-style-image: url(image.svg);
      content: url(image.svg);
}
<img src="image.svg" alt="">
```

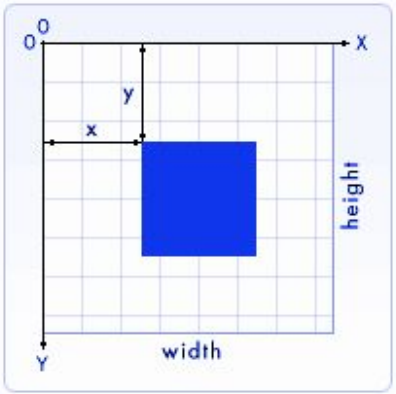
```
<svg viewBox="0 0 30 10" width="300"
height="200">
  <rect width="100%" height="100%" fill="red"
/>
</svg>

<svg viewBox="0 0 30 10">
  <use xlink:href="image.svg"/>
</svg>
```

SVG

Common Attributes

width
height
x – Horizontal coordinate
y – Vertical coordinate

fill – Fill color
stroke – Stroke color
stroke-width



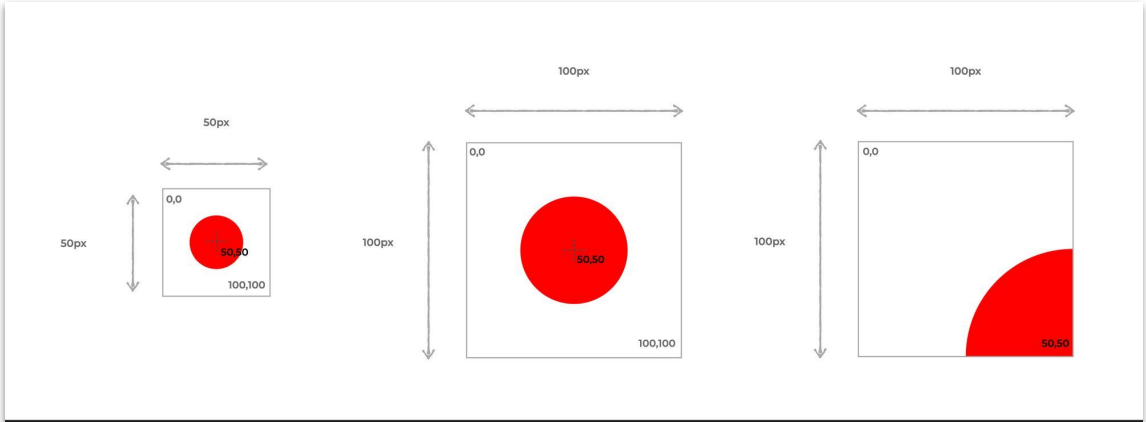
- Clipping, Masking and Compositing properties:
  - 'clip-path'
  - 'clip-rule'
  - 'mask'
  - 'opacity'
- Filter Effects properties:
  - 'enable-background'
  - 'filter'
  - 'flood-color'
  - 'flood-opacity'
  - 'lighting-color'
- Gradient properties:
  - 'stop-color'
  - 'stop-opacity'
- Interactivity 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'

viewBox

– Defines the aspect ratio
– Responsible for scaling

viewBox="0 0 24 24"

- coordinate x
- coordinate y
- width
- height




```
<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>
```

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>

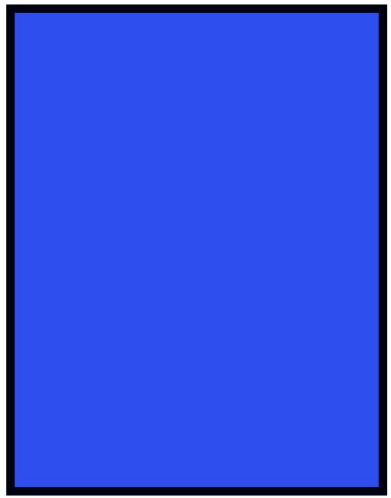
Grouping container – <g>
Hidden element – <defs>
Reuse an element – <use>

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

SVG

rect

- width
- height
- fill
- stroke
- stroke-width

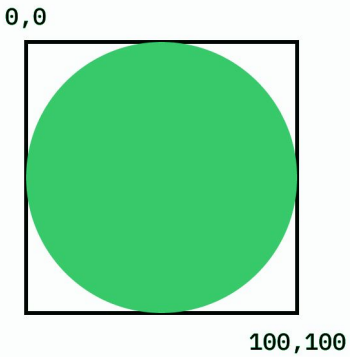


```
...
<rect width="180"
height="220"
style="fill:pink;
stroke-width:4;
stroke:rgb(0,0,0)">
...
```

SVG

circle

fill
stroke
stroke-width
cx – central horizontal coordinate
cy – central vertical coordinate
r – radius

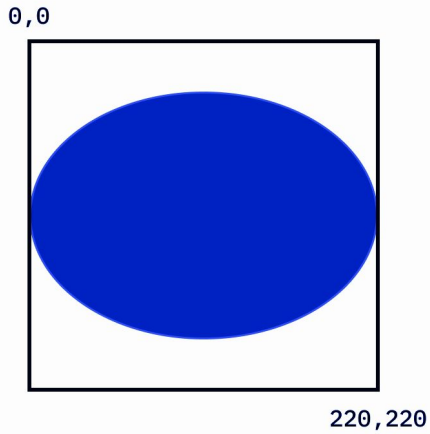



```
<svg width="100"
height="100">
<circle cx="50"
cy="50" r="50"
fill="yellow" />
</svg>
```

SVG

ellipse

cx – central horizontal coordinate
cy – central vertical coordinate
rx – horizontal radius
ry – vertical radius




```
...
<ellipse
cx="110" cy="110"
rx="100"ry="70"
style="fill:purple;
stroke:pink">
...
```

SVG

text

x

y

fill

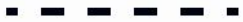
SVG

line

x1/x2 – first/last horizontal coordinate

y1/y2 – first/last vertical coordinate

```
...<line x1="0" y1="0" x2="120"
y2="0" style="stroke:pink" />...
```



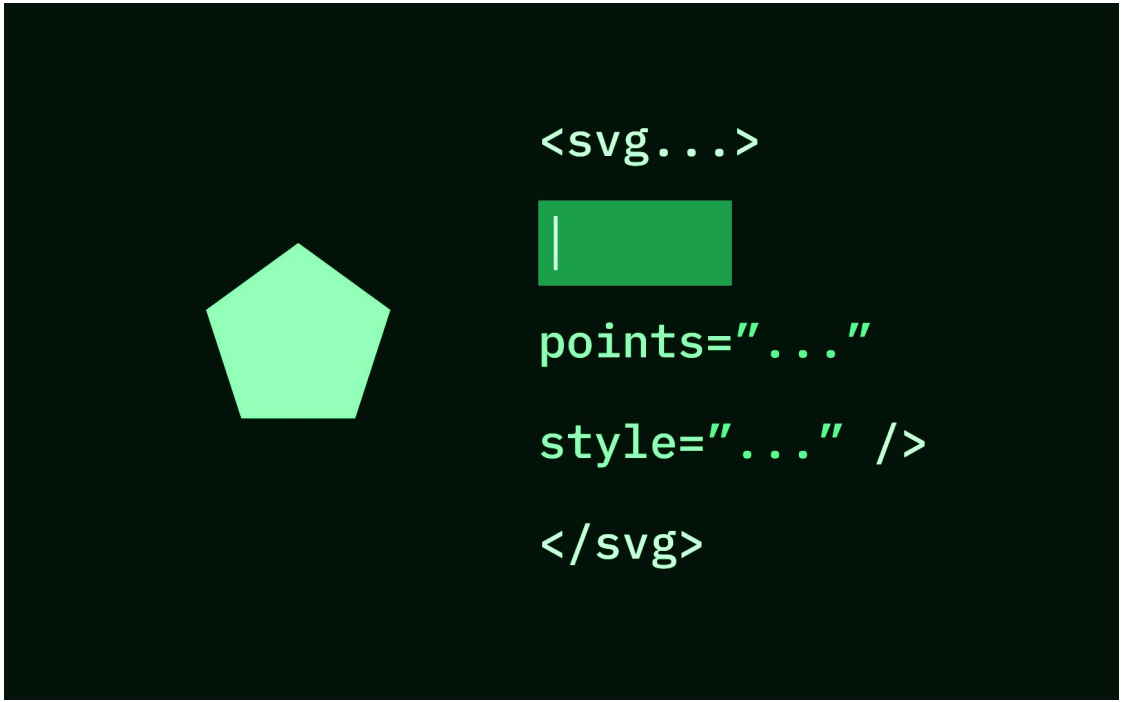
```
...<line x1="0" y1="0" x2="120" y2="0"
style="stroke-dasharray:10,10" />...
```

```
...<line x1="0" y1="0" x2="120"
y2="0" style="stroke-width:6" />...
```

SVG

polygon

points – vertex coordinates
fill
stroke
stroke-width



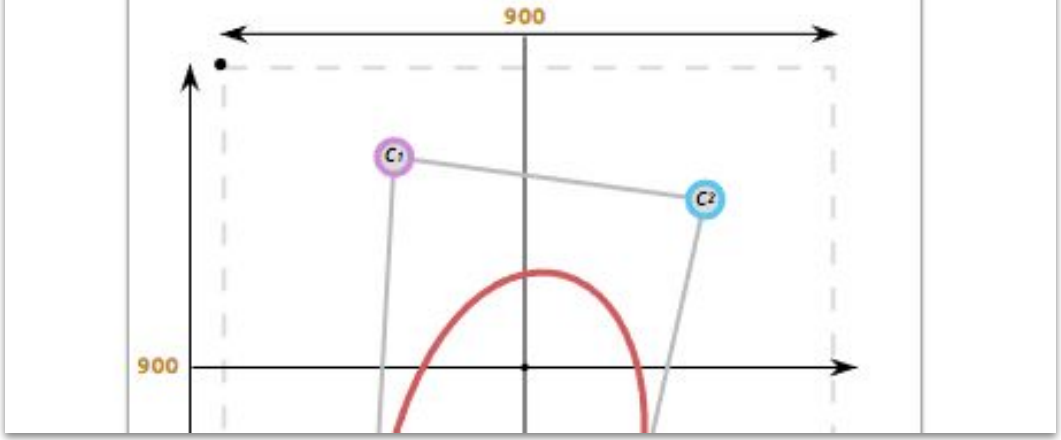
SVG

path

line for drawing figures

d – listing of coordinates

```
<svg viewBox='-450 -450 900 900'>
  <path d='M-229,333
           C-193,-312 267,-249 161,222'/>
</svg>
```

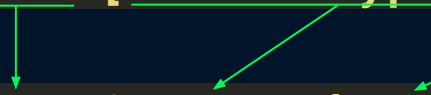


base64

❌ don't use

base64 at all —
with raster images —

```
<img src="data:[<media-type>][;base64],<data>" alt="logo">
```



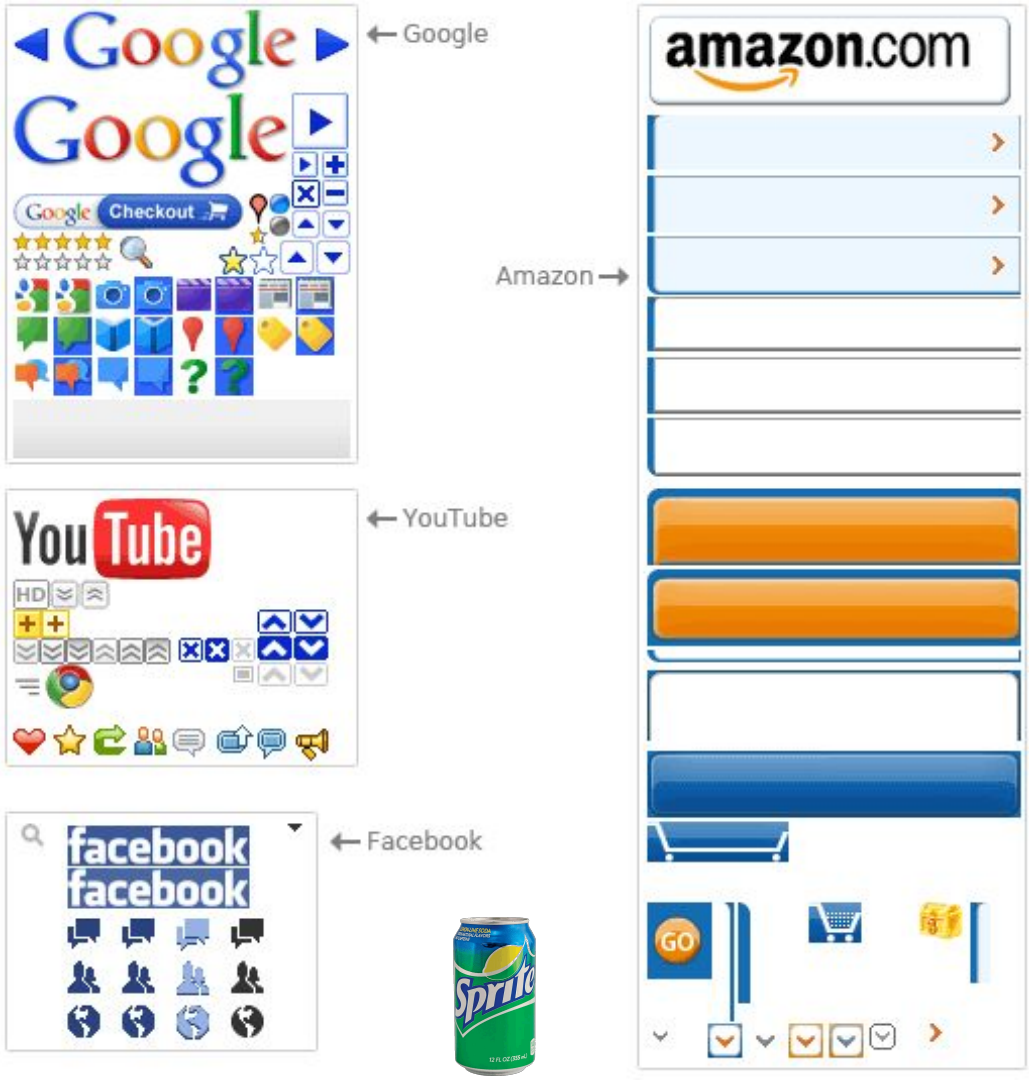
```
<img src="data:img/png;base64,iVBORw0KGgoAAAANSUhEUgAAACAAA
AAgCAMAAABEpIrGAAABO1BMVEU/NFy5BOguxz1eFmMqJnD7ctI1mK3/gf6e
wKFQpwQIHWi1BUsiVTYrHr/j88JngEGdlzVo33v+JvN0xbq1+b9nQ10ofsI
dwN89u6keUO8FD8M65/ZVS0hhpqwKse7Exw0AAABW0lEQVQ4jX2TC1OCQBS
Ft5qyUppqnF72JHN6OVaMxwiLAkuNMs3U1B5WZv3/X9AuAhKLHGbg7pxvzt
0LLCFU0bW6Vte8UseIrQx8FXOAI00XAEPX49ZF3XK8NukApJK6AqZSqbylO
oxm/pW4NQJcOoswcPvfNoGGXStbwLzHJyVXglwDdgOARA5QZS9wCCStMloG
Nr0+A6yE+xmgWOGAczshewyIYc53WkS69CV1eZ8lPLBni/pqZBigTBeof8p
NwPQIXDdm2Sc62=" alt="logo">
```

Sprite

use sprites to join all images network requests

can be done with generators like spritecow.com

to not do 1x and 2x sprites can be used SVG



Icon fonts

Huge font file
with all the icons
suitable for all projects

❌ never use

```
Filter :hov .cls
element.style {
  line-height: 1;
  color: rgb(30, 48, 80);
}
:is(i[class*="fa-"], .svg-inline--fa) {  main.css:1671
  --spacing-inline: var(--spacing-6xs);
}
.fa-solid, .fas {  all.css:9
  font-weight: 900;
}
.fa-home, .fa-home-alt, .fa-home-lg-alt, .fa-house {  all.css:8
  --fa: "\f015";
  --fa--fa: "\f015\f015";
}
.fa-classic, .fa-light, .fa-regular, .fa-solid, .fa-thin, .fal, .far, .fas, .fat {  all.css:6
  font-family: "Font Awesome 6 Pro";
}
.fa, .fa-brands, .fa-classic, .fa-duotone, .fa-light, .fa-regular, .fa-sharp, .fa-sharp-duotone, .fa-solid, .fa-thin, .fab, .fad, .fadl, .fadr, .fadt, .fal, .far, .fas, .fasdl, .fasdr, .fasds, .fasdt, .fasl, .fasr, .fass, .fast, .fat {  all.css:6
  -moz-osx-font-smoothing: grayscale;
  -webkit-font-smoothing: antialiased;
  display: var(--fa-display, inline-block);
  font-style: normal;
  font-variant: ▸ normal;
  line-height: 1;
  text-rendering: auto;
}
i, .i, em, .em {  main.css:2533
  --font-style: italic;
  font-style: italic;
  font-style: var( font-style);
}
*, *::before, *::after {  main.css:139
  box-sizing: border-box;
}
i {  user agent stylesheet
```

Introducing Font Awesome 7!

house

f015

HTML REACT VUE SVG

```
<i class="fa-solid fa-house"></i>
```

Accessibility + Icons