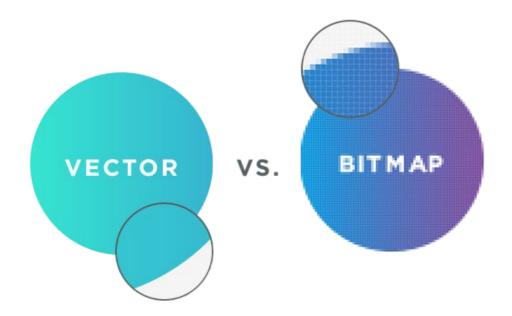
Deep Dive into **SVG**



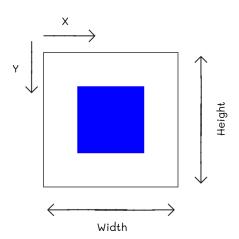
Mohsen Madani – Feb 2024

Why?



Rectangle

```
1 <svg
2  width="200"
3  height="200"
4  style="margin: 100px;border: 1px solid;"
5  viewBox="0 0 100 100"
6 >
7  <rect width="50" height="50" y="25" x="25" fill="blue" />
8 </svg>
```



MoveTo (M or m): Moves the "pen" to a new location without drawing a line.

M 10 10: Moves the pen to the coordinates (10, 10)

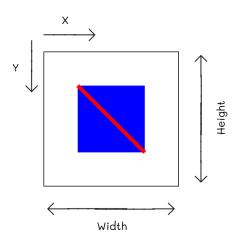
m 20 20: Moves the pen 20 units right and 20 units down from its current position

LineTo (L or I): Draws a line from the current position to a new position.

L 20 20: Draws a line from the current position to the coordinates (20, 20)

1 30 30: Draws a line 30 units right and 30 units down from the current position

```
1 <svg
2  width="200"
3  height="200"
4  style="margin: 100px;border: 1px solid;"
5  viewBox="0 0 100 100"
6 >
7  <rect width="50" height="50" y="25" x="25" fill="blue" />
8  <path stroke-width="3" d="M 25 25 L 75 75" stroke="red" />
9 </svg>
```



Horizontal LineTo (H or h): Draws a horizontal line from the current position to a new x-coordinate.

50: Draws a horizontal line from the current position to the x-coordinate 50

h 10: Draws a horizontal line 10 units to the right from the current position

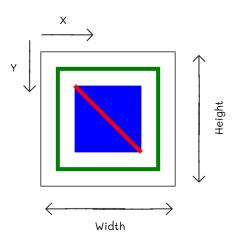
Vertical LineTo (V or v): Draws a vertical line from the current position to a new y-coordinate.

V 60: Draws a vertical line from the current position to the y-coordinate 60

v 20: Draws a vertical line 20 units down from the current position

ClosePath (Z or z): Closes the current subpath by drawing a straight line from the current position back to the starting point of the subpath.

```
1 <svg
 2 width="200"
 3 height="200"
 4 style="margin: 100px;border: 1px solid;"
 5 viewBox="0 0 100 100"
 6 >
     <rect width="50" height="50" y="25" x="25" fill="blue" />
     <path stroke-width="3" d="M 25 25 L 75 75" stroke="red" />
    <path
      stroke-width="3"
10
11
      stroke="green"
      d="M 12.5 12.5 H 87.5 V 87.5 L 12.5 87.5 Z"
      fill="transparent"
14
    />
15 </svg>
```

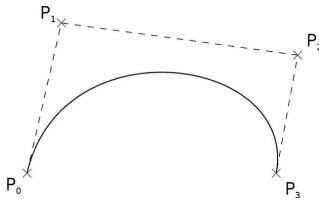


Path (Curves)

- Bézier Curves
 - Cubic
 - Quadratic
- Arcs

Bézier Curves

Bézier curves are parametric curves (with the parameter t varying from 0 to 1) that are defined by a set of **control points**.



A bezier curve with 4 control points (cubic curve)

Bézier Curves

The curve goes from the "Start" point to the "End" point while the "Control" points define its curvature.

Control 1 Control 2

.

Start

End

Bernstein Polynomials

A Bernstein polynomial of degree n is defined as a sum of **Bernstein basis** polynomials, each multiplied by a **Bernstein coefficient**.

Bernstein coefficient

Bernstein Polynomial of degree
$$n$$

$$B_n(x) = \sum_{v=0}^n \frac{\beta_v}{b_{v,n}(x)} \longrightarrow b_{v,n}(x) = \binom{n}{v} x^v (1-x)^{n-v}$$

Bernstein Coefficient

It's a Bernstein polynomial where the Bernstein coefficients are the **control points**. So, when building a bézier curve, we're essentially approximating a **real function**!

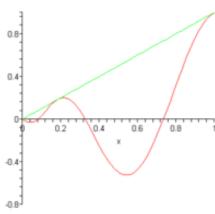
Let ${m f}$ be a continuous function on the interval [0, 1]. Consider the Bernstein polynomial

$$B_n(f)(x) = \sum_{
u=0}^n f\left(rac{
u}{n}
ight) b_{
u,n}(x).$$

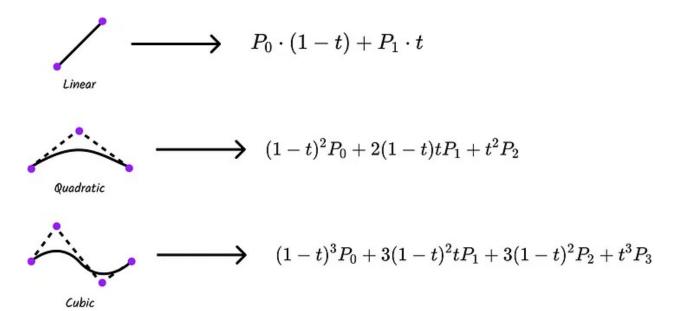
It can be shown that

$$\lim_{n o\infty}B_n(f)=f$$

uniformly on the interval [0, 1]



Bézier Curves



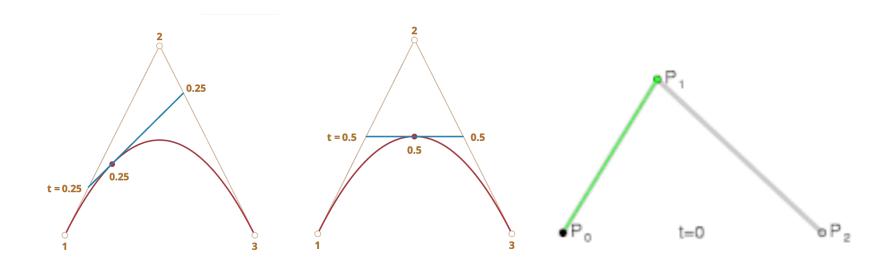
Recursive Bézier Curves

$$(1-t)^2P_0+2(1-t)tP_1+t^2P_2$$
 $\equiv (1-t)\cdot \left[(1-t)P_0+tP_1
ight]+t\cdot \left[(1-t)P_1+tP_2
ight]$ Line that connects PO and PI Line that connects PI and P2

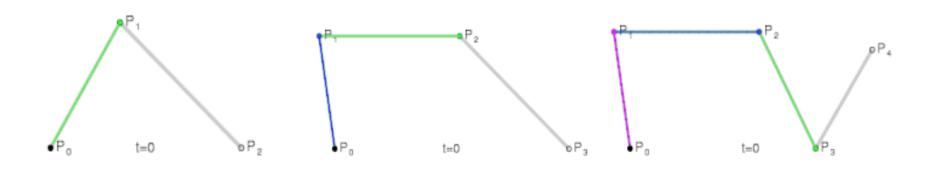
Line that connects lines PO_PI and PI_P2

$$B_{P_0,P_1,...,P_n}(t) = (1-t)B_{P_0,...,P_{n-1}}(t) + tB_{P_1,...,P_n}(t)$$

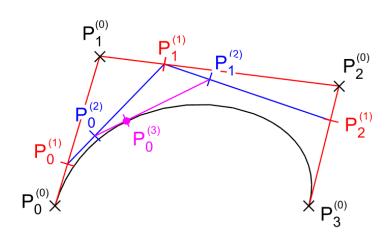
De Casteljau's algorithm

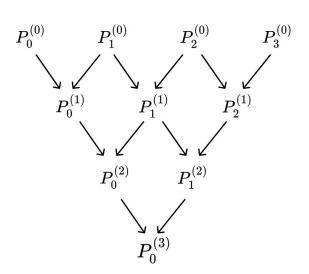


De Casteljau's algorithm



De Casteljau's algorithm



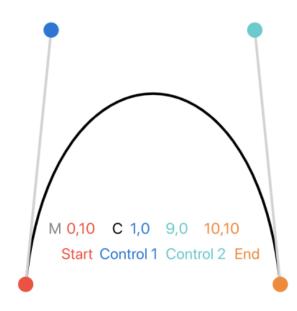


$$P_i^{(j)} = (1-t)P_i^{j-1} + tP_{i+1}^{j-1}$$

Path (Curves)

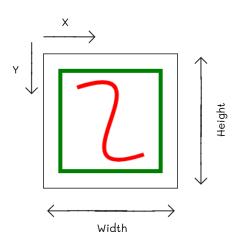
- Bézier Curves
 - Cubic
 - Quadratic
- Arcs

Cubic Bézier Curves

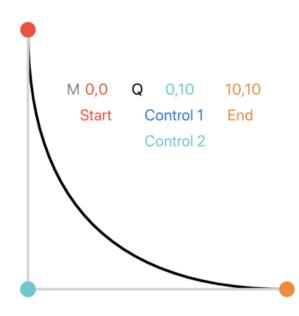


Path (Cubic Bézier Curves)

```
1 < svq
 2 width="200"
 3 height="200"
    style="margin: 100px;border: 1px solid;"
    viewBox="0 0 100 100"
 6 >
     <path</pre>
       stroke-width="3"
       stroke="red"
       d="M 25 25 C 100 0, 0 100, 75 75"
10
       fill="transparent"
11
12
     />
     <path
       stroke-width="3"
14
       stroke="green"
15
       d="M 12.5 12.5 H 87.5 V 87.5 L 12.5 87.5 Z"
16
       fill="transparent"
17
18
     />
19 </svg>
```

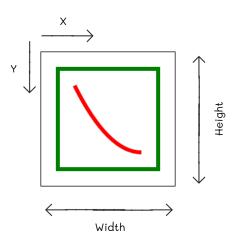


Quadratic Bézier Curves



Path (Quadratic Bézier Curves)

```
1 <svq
 2 width="200"
 3 height="200"
    style="margin: 100px;border: 1px solid;"
     viewBox="0 0 100 100"
 6 >
     <path
       stroke-width="3"
       stroke="red"
      fill="transparent"
10
       d="M 25 25 Q 50 75, 75 75"
     />
12
     <path
      stroke-width="3"
14
      stroke="green"
15
16
      d="M 12.5 12.5 H 87.5 V 87.5 L 12.5 87.5 Z"
       fill="transparent"
18
    />
19 </svq>
```



Path (Curves)

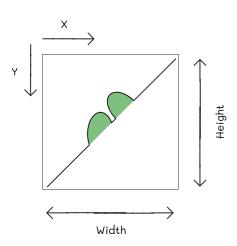
- Bézier Curves
 - Cubic V
 - Quadratic 🗸
- Arcs

The SVG path command for drawing arcs is $\bf A$ or $\bf a$. This command allows you to draw a section of an ellipse. The syntax for the command is as follows:

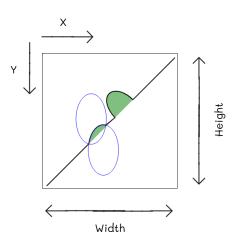
```
A rx ry x-axis-rotation large-arc-flag sweep-flag x y
a rx ry x-axis-rotation large-arc-flag sweep-flag dx dy
```

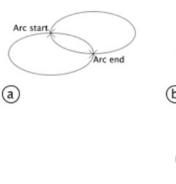
- rx and ry: These are the x and y radii of the ellipse. They define the size of the ellipse.
- x-axis-rotation: This parameter specifies the rotation angle of the ellipse's x-axis.
- large-arc-flag: This flag determines which of the four possible arcs to draw between the current position and the end point (x, y). If the value is 0, the smaller arc is chosen. If it's 1, the larger arc is chosen.
- sweep-flag: This flag determines the direction in which the arc is drawn. If the value is 0, the arc is drawn in a negative-angle direction. If it's 1, the arc is drawn in a positive-angle direction.
- \mathbf{x} and \mathbf{y} or \mathbf{dx} and \mathbf{dy} : These are the \mathbf{x} and \mathbf{y} coordinates where the arc should end. If the command is capitalized (\mathbf{A}), the coordinates are absolute. If it's lowercase (\mathbf{a}), the coordinates are relative to the current position.

```
1 < svq
 2 width="320"
    height="320"
     style="margin: 100px;border: 1px solid;"
 5 >
     <path
       d="M 10 315 L 110 215 A 30 50 0 0 1 162.55 162.45 L 172.55 152.45
          A 30 50 -45 0 1 215.1 109.9 L 315 10"
      stroke="black"
      fill="green"
      stroke-width="2"
      fill-opacity="0.5"
12
    />
13 </svg>
```



```
1 < svq
 2 width="320"
    height="320"
     style="margin: 100px;border: 1px solid;"
 5 >
     <path
       d="M 10 315 L 110 215 A 36 60 0 0 1 150.71 170.29 L 172.55 152.45
          A 30 50 -45 0 1 215.1 109.9 L 315 10"
      stroke="black"
      fill="green"
      stroke-width="2"
      fill-opacity="0.5"
12
    />
13 </svg>
```



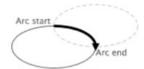




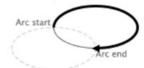
b large-arc-flag=0 sweep-flag=0



d large-arc-flag=1 sweep-flag=0



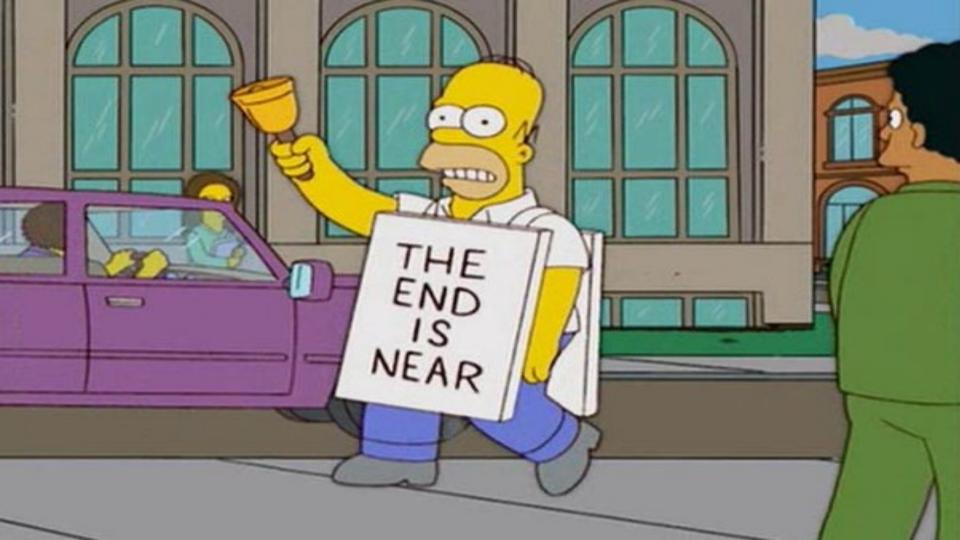
© large-arc-flag=0 sweep-flag=1



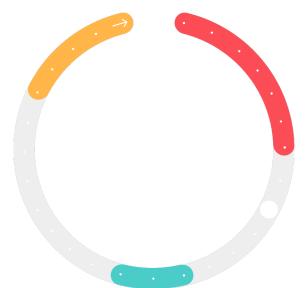
e large-arc-flag=1 sweep-flag=1



x-axis-rotation=30

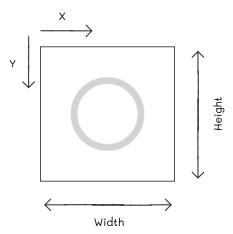


Ready to recreate the **HerLife** cycle?

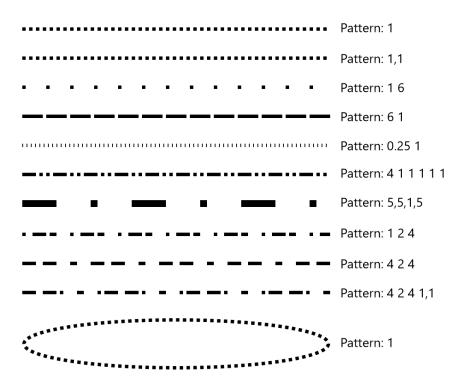


Circle

```
1 <svg
 2 width="200"
 3 height="200"
 4 style="margin: 100px;border: 1px solid;"
 5 viewBox="0 0 100 100"
 6 >
     <circle
      r="25"
      cx="50"
 9
      cy="50"
10
11
      stroke="lightgray"
      fill="transparent"
12
      stroke-width="5"
13
14
    />
15 </svg>
```

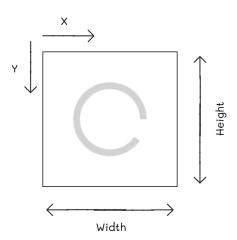


Stroke Dash Array



Stroke Dash Array

```
1 < svq
 2 width="200"
 3 height="200"
 4 style="margin: 100px;border: 1px solid;"
     viewBox="0 0 100 100"
 6 >
     <circle
       r="25"
       cx="50"
 9
10
       cv="50"
       stroke="lightgray"
11
12
       fill="transparent"
13
       stroke-width="5"
       stroke-dasharray="0 15.7 141.3"
14
15
    />
16 </svq>
```



Stroke Dash Array

```
1 < svq
 2 width="200"
 3 height="200"
 4 style="margin: 100px;border: 1px solid;"
 5 viewBox="0 0 100 100"
 6 >
 7 <circle
   r="25"
   cx="50"
     cv="50"
     stroke="lightgray"
     fill="transparent"
     stroke-width="5"
      stroke-dasharray="0 15.7 141.3"
15 />
16 </svq>
```

```
r=25

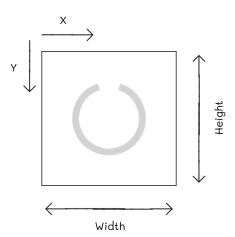
Circumference = 2\pi r=2*3.14*25=157

Circumference * 10\%=15.7

Circumference * 90\%=141.3
```

Rotate

```
1 <svg
2 width="200"
    height="200"
4 style="margin: 100px;border: 1px solid;"
    viewBox="0 0 100 100"
6 >
    <circle
      r="25"
      cx="50"
      cv="50"
10
11
      stroke="lightgray"
12
      fill="transparent"
13
      stroke-width="5"
14
      stroke-dasharray="0 15.7 141.3"
      transform="rotate(-108, 50, 50)"
15
16
    />
17 </svg>
```



Rotate

```
1 <svg
 2 width="200"
 3 height="200"
 4 style="margin: 100px;border: 1px solid;"
 5 viewBox="0 0 100 100"
 6 >
    <circle
     r="25"
     cx="50"
      cv="50"
      stroke="lightgray"
      fill="transparent"
13
      stroke-width="5"
      stroke-dasharray="0 15.7 141.3"
14
      transform="rotate(-108, 50, 50)"
16 />
17 </svq>
```

```
Circle = 360 degree

5% Circle = 18

90 + 18 = 108
```

Stroke Line Cap







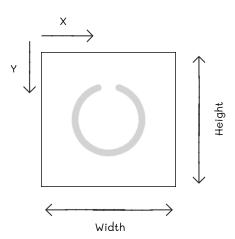
round cap



square cap

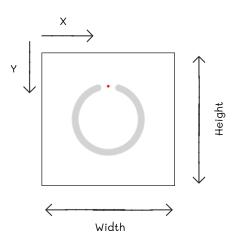
Stroke Line Cap

```
1 <svg
 2 width="200"
 3 height="200"
 4 style="margin: 100px;border: 1px solid;"
 5 viewBox="0 0 100 100"
 6 >
     <circle
       r="25"
       cx="50"
       cv="50"
10
       stroke="lightgray"
11
      fill="transparent"
13
       stroke-width="5"
       stroke-dasharray="0 15.7 125.6"
14
       transform="rotate(-108, 50, 50)"
15
16
       stroke-linecap="round"
17
    />
18 </svq>
```



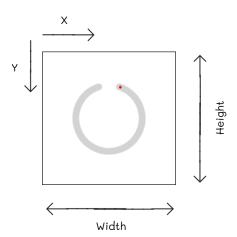
Adding Dots

```
1 <svq
 2 width="200"
 3 height="200"
    style="margin: 100px;border: 1px solid;"
    viewBox="0 0 100 100"
    <circle
      r="25"
 9
      cx="50"
10
      cv="50"
      stroke="lightgray"
11
12
      fill="transparent"
     stroke-width="5"
13
14
      stroke-dasharray="0 15.7 141.3"
      transform="rotate(-108, 50, 50)"
15
16
      stroke-linecap="round"
17
18
    <circle
19
      r="1"
20
     cx="50"
      cy="25"
21
22
      fill="red"
23 />
24 </sva>
```

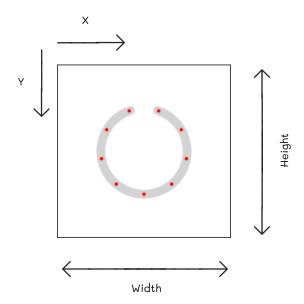


Rotating Dots

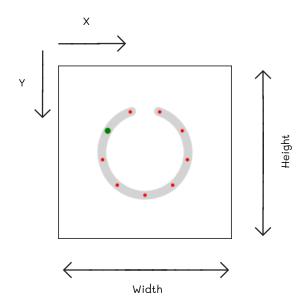
```
1 <sva
 2 width="200"
    height="200"
    style="margin: 100px;border: 1px solid;"
    viewBox="0 0 100 100"
6 >
7 <circle
      r="25"
9
      cx="50"
      cy="50"
      stroke="lightgray"
11
      fill="transparent"
12
      stroke-width="5"
13
14
      stroke-dasharray="0 15.7 141.3"
      transform="rotate(-108, 50, 50)"
15
      stroke-linecap="round"
16
17
    />
    <circle
      r="1"
      cx="50"
20
      cv="25"
      fill="red"
      transform="rotate(20, 50, 50)"
24 />
25 </svg>
```



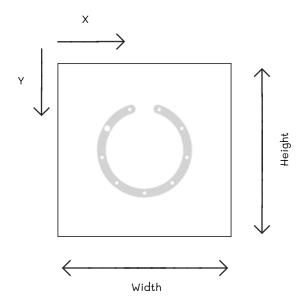
Add More Dots



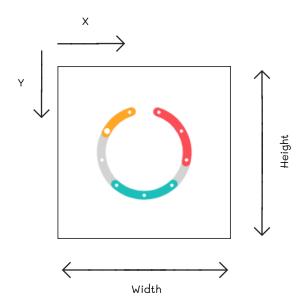
Add Current Dot



Colors



Add Ranges



Mission Complete



References

MDN web docs (link)

Javascript Info (<u>link</u>)

DevTo (link)

Wikimedia Commons (link)

Wikipedia (link)

Q&A

Ask me question if you have any

The END

Mohsen Madani





