



SVG

- [svg_playground](#)

SVG란

- 깨지지 않고, 용량이 적고, 출력이 빠르며, 수정과 애니메이션이 가능
- 2차원 벡터 그래픽을 서술하는 XML 기반의 마크업 언어
- 그림을 그리는 명령
- SVG태그 내부에 담을 수 있는 것은 원, 사각형, 다각형, 라인, path 등이 있음

```
<svg width="가로영역" height="세로영역">  
  SVG그래픽  
  ...  
</svg>
```

사용하는 이유

- 웹사이트 로딩 속도가 빠르다. SVG는 XML코드이기 때문에 파일의 크기가 작다.
- 어떤 크기에서든 재활용이 가능. SVG는 크기에 따라 이미지가 왜곡되거나 품질이 떨어져 보이지 않는다.
- CSS로 디자인 수정이 가능하다.

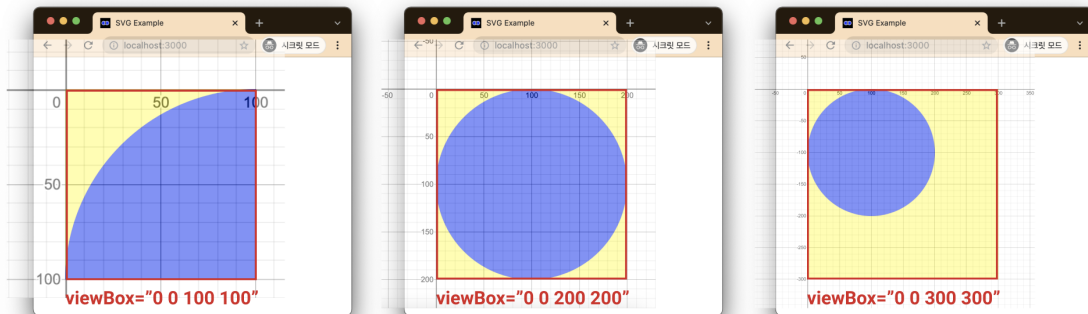
단점

- 너무 복잡한 SVG는 비효율적일 수 있다.
- 많은 모양, 색상, 그라디언트를 포함하여서 파일 크기가 JPG나 PNG 보다 커지는 경우가 종종 있기 때문에 단점이 될 수 있긴함

SVG의 기본속성

```
<svg height="100" width="100" viewBox="0 0 50 50">
  <circle cx="20" cy="20" r="20" fill="red" />
</svg>
```

- width, height : viewport 사이즈
- viewBox : **SVG 내부 요소들의 좌표 영역과 비율**, 그리고 뷰포트 내에서 도형을 보여줄 위치를 설정할 수 있다.
 - **viewBox** 속성은 **min-x**, **min-y**, **width**, **height** 를 순서대로 받는다.
 - **min-x**, **min-y** : 왼쪽 위를 기준으로 한 뷰박스의 좌표
 - **width**, **height** : 가로, 세로 뷰박스 사이즈



하위 태그

path

- <path /> 태그는 일러스트레이터처럼 패스(선과 면)을 이용한 태그
- 속성 중 **d** 라는 속성에 다양한 패스 데이터를 사용할 수 있습니다.

데이터 종류	설명
M(move to)	이동좌표

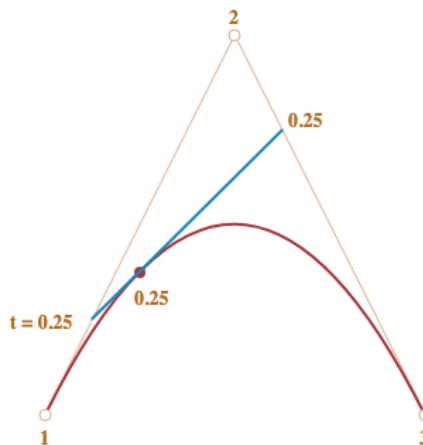
데이터 종류	설명
L	선
H	수평선
V	수직선
C	곡선
S	부드러운 곡선
Q	2차 베지어 곡선
T	부드러운 2차 베지어 곡선
A	타원형 호
Z	가까운 경로, 첫 점으로 연결

▼ 삼각형

```
<svg width="400" height="210" >
  <path d="M150 0 L75 200 L225 200 Z" />
</svg>
```

▼ 곡선

- 베지어 곡선은 무제한으로 크기를 조정할 수 있는 부드러운 곡선을 모델링하는데 사용됩니다.
- 일반적으로는 **두개의 끝점** 과 하나 또는 두개의 **제어점** 을 선택합니다.



- 하나의 제어 지점을 갖는 베지어 곡선을 이차 베지어 곡선이라고 합니다.
- 두개의 제어 지점을 갖는 베지어 곡선을 큐빅 베지어 곡선(입방향 3차원 곡선)이라고 합니다.

```
<svg width="400" height="400" >
  <path d="M 20 350 q 150 -300 300 0" stroke="blue" stroke=
</svg>
```

text

- 텍스트를 벡터 방식으로 작성

```
<text x="0" y="15" fill="red">I love OSSAM!</text>
  <text x="10" y="20" style="fill:red;">Several lines:
    <tspan x="10" y="45">First line.</tspan>
    <tspan x="10" y="70">Second line.</tspan>
  </text>
```

▼ textPath

<textPath>

To render text along the shape of a [<path>](#), enclose the text in a [<textPath>](#) element that has an [href](#) attribute with a reference to the [<path>](#) element.

- 참고

```
<svg height="1000" width="1000">
  <path id="curve" fill="red" d="M73.2,148.6c4-6.1,65.5-96.8,
  <text width="500">
    <textPath xlink:href="#curve">
      가나다라마바사아자차카타파하
    </textPath>
  </text>
```

</svg>

<path

```
id="curve"
```

```
fill="none"
```

```
stroke="red"
```

```
d="M10,90 Q90,90 90,45 Q90,10 50,10 Q10,10 10,40 Q10,70 10,90
```

```
body {
```

```
background-color: #333;
```

```
font-family: cursive;
```

```
font-size: 35px;
```

}

```
path {
```

```
fill: transparent;
```

}

```
text {
```

```
fill: #FF9800;
```

}

circle

- circle

```
<circle cx="100" cy="100" r="50" style="fill:lime;stroke:purple;stroke-width:2px;" />
```

- ellipse

```
<ellipse cx="200" cy="100" rx="100" ry="50" style="fill:#ddd"/>
```

SVG 그룹화

group

```
<svg viewBox="0 0 100 100" xmlns="http://www.w3.org/2000/svg">
  <!-- Using g to inherit presentation attributes -->
  <g fill="white" stroke="green" stroke-width="5">
    <circle cx="40" cy="40" r="25" />
    <circle cx="60" cy="60" r="25" />
  </g>
</svg>
```

use

```
<svg viewBox="0 0 30 10" xmlns="http://www.w3.org/2000/svg">
  <circle id="myCircle" cx="5" cy="5" r="4" stroke="blue" />
  <use href="#myCircle" x="10" fill="blue" />
  <use href="#myCircle" x="20" fill="white" stroke="red" />
</svg>
```

SVG sprite

- 💜 1타강사 블로그 💜

```
<svg xmlns="http://www.w3.org/2000/svg">
  <symbol id="cloud-svgrepo-com" viewBox="0 0 24 24">
    <path stroke="#000" stroke-linecap="round" stroke-linejoin="round" />
  </symbol>
  <symbol id="cloud-moon-svgrepo-com" viewBox="0 0 24 24">
    <path stroke="#000" stroke-linecap="round" stroke-linejoin="round" />
  </symbol>
  <symbol id="cloud-fog-svgrepo-com" viewBox="0 0 24 24">
    <path stroke="#000" stroke-linecap="round" stroke-linejoin="round" />
  </symbol>
```

```
    </symbol>
  </svg>
```

```
// <svg><use href="[스프라이트 이미지.svg]#facebook" /></svg>
```

```
import SpriteSVG from "../svg_sprite.svg"
```

```
const TestSVG = ({id, color, size=24}) => (
  <svg fill={color} width={size} height={size}>
    <use href={`_${SpriteSVG}#${id}`} />
  </svg>
)
```

```
<TestSVG id="cloud-moon-svgrepo-com" size=20/>
```

d3에서 사용방법

react playground

```
import { select } from "d3";
import { useEffect, useState } from "react";
import d3 from 'd3';

const TestChart = () => {
  const [svgRef, setSvgRef] = useState(null);

  useEffect(() => {
    if (!svgRef) return;
    const svg = select(svgRef).attr("width", 500).attr("height", 500);
    svg
      .append("path")
      .attr("fill", "red")
      .attr("stroke", "blue")
      .attr("stroke-width", 2)
  }, [svgRef]);
}
```

```

        .attr("stroke-linejoin", "round")
        .attr("d", "M 20 350 q 150 -300 300 0");
    return () => {
        setSvgRef(null);
        select(svgRef).selectAll("svg *").remove();
        select(svgRef).attr("width", 0).attr("height", 0).attr("v:
    };
    }, [svgRef]);

    return (
        <div style={{width:"500px", height:"500px", background:'green
            <svg ref={{(ref) => ref && setSvgRef(ref)}} width={0} height
        </div>
    );
};

export default TestChart;

```

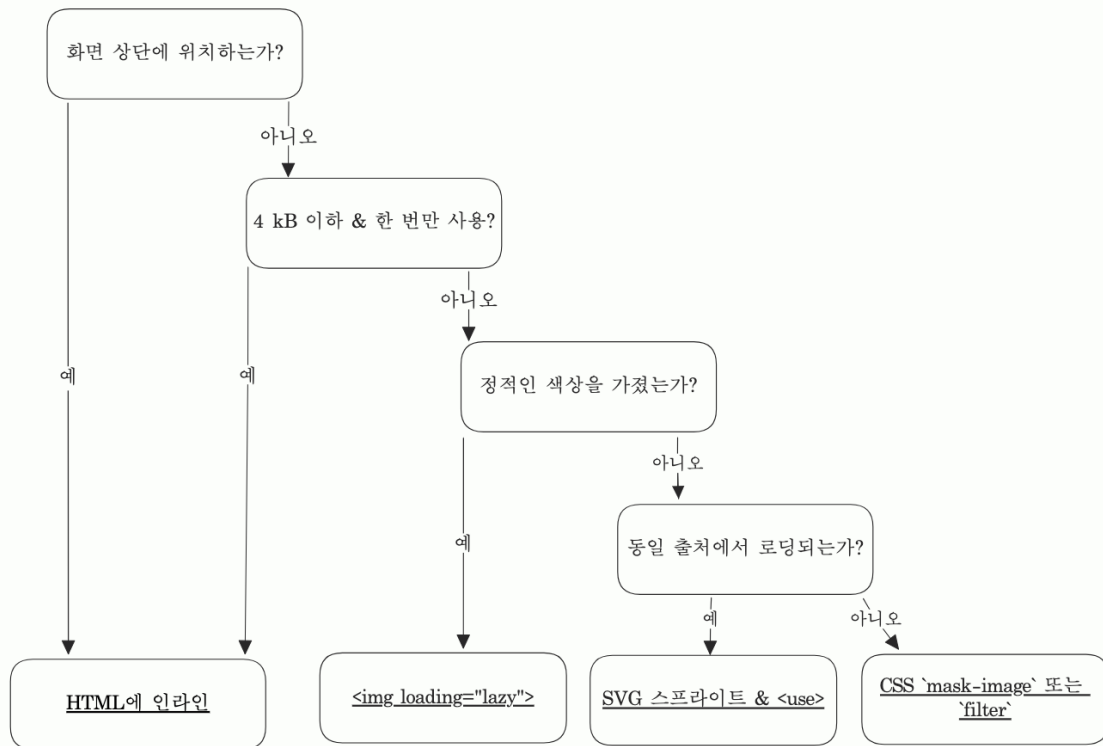
- d3-scale
- d3-array
 - ticks
 - sets - union, intersection

응용 사이트

- https://observablehq.com/@d3/gallery?utm_source=d3js-org&utm_medium=nav&utm_campaign=try-observable
- <https://d3-graph-gallery.com/index.html>

하지만...SVG in JS는 naver..

2023년 SVG-in-JS와 결별



질문

- SVG 관리 어떻게 하시는지
- 차트 라이브러리 뭐쓰시는지