SVG

Scalable Vector Graphics

Scalable Vector Graphics

SVG 란..???

- 2001년 표준으로 만들어진 벡터 그래픽을 표시하기 위한 기술
- HTML5와는 별도 표준안
- 사용 빈도수 높지 않았음...
- 하지만 HTML5가 주목받으면서 다시 주목
- HTML5에서는 태그 형식처럼 본문에 직접 사용할 수 있음
- XML 포맷 기반으로 정의

장점

- 사용자 환경의 해상도가 달라져도 품질 저하 없이 깨끗한 이미지를 보여줌
- 다양화되는 환경에 유연하게 대처할 수 있는 방안
- 텍스트 편집가능
- 공개 표준이면서 순수 XML 포맷

Scalable Vector Graphics

SVG W3C Recommendation

- SVG 1.0 : 2001년 9월 4일
- SVG 1.1 : 2003년 1월 14일
- SVG 1.1 (Second Edition) : 2011년 8월 16일

SVG를 HTML5 일반 태그처럼 HTML 본문에 직접 사용

HTML5에서는 추가된 <svg> 태그를 통해 직접 본문에 추가할 수 있음, 그렇기 때문에 HTML 요소처럼 컨트롤 가능.
이러한 방식을 Inline SVG라고 함, 대부분 브라우저가 모두 지원하고 있음

Embed SVG Directly Into HTML Pages

Here is an example of a simple SVG graphic:

and here is the HTML code:

Example

```
<!DOCTYPE html>
<html>
<body>

<h1>My first SVG</h1>

<svg width="100" height="100">
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />
</svg>

</body>
</html>
```

Try it Yourself »

Scalable Vector Graphics

사각형 그리기

- 사각형 그리기 위한 요소 : <rect>

```
<rect
   x="[사각형의 좌측 상단 X좌표]"
   y="[사각형의 좌측 상단 Y좌표]"
   width="[사각형 폭]"
   height="[사각형 높이]"
   rx="[가로 방향 외곽선 곡률]"
   ry="[세로 방향 외곽선 곡률]"
>
```

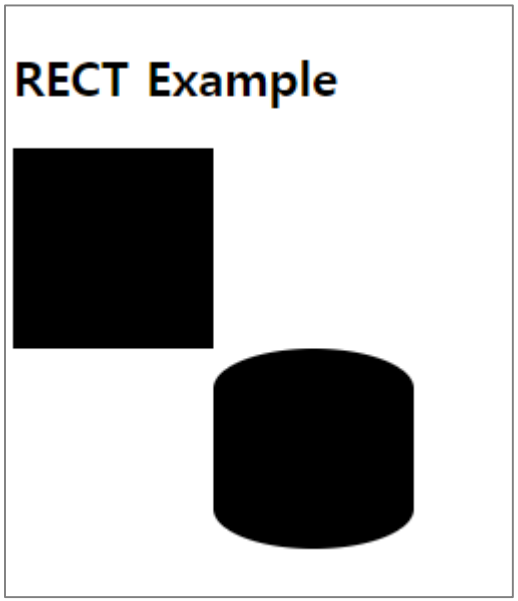
X는 우측 방향으로 증가
Y는 아랫방향으로 증가

벡터 그래픽이므로 크기 단위 무의미
그러므로 숫자만 입력

Scalable Vector Graphics

사각형 그리기

```
<!DOCTYPE HTML>
<html lang="en">
<head>
    <style>
    svg{
      width:400px;
      height:200px;
    }
    </style>
</head>
<body>
    <div id="example">
      <h2>RECT Example</h2>
      <div id="exampleSub">
        <svg>
          <rect width="100" height="100"/>
          <rect x="100" y="100" width="100" height="100" rx="50" ry="20"/>
        </svg>
      </div>
    </div>
</body>
</html>
```



Scalable Vector Graphics

SVG 요소 스타일 속성

- 기본적인 형태 외 선 색상이나 면 색상 등을 지정하기 위해 style 속성 사용
- CSS 와 비슷한 형식

fill : 채움 색상
stroke : 선 색상

- CSS 규칙과 동일하게 사용 투명도값 포함한 RGBA 값도 사용 가능

stroke-width : 선 두께

- 단위 없이 수치만 지정

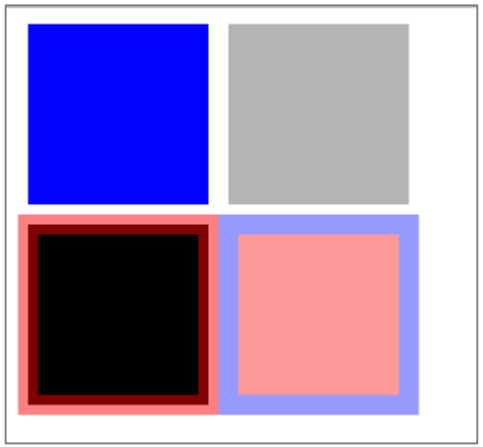
opacity : 투명도

- CSS 규칙과 동일하게 0부터 1사이 값 지정

Scalable Vector Graphics

SVG 요소 스타일 속성

```
<!DOCTYPE HTML>
<html lang="en">
<head>
    <style>
      svg{
        width:400px;
        height:200px;
      }
      #cssControl{
        fill:#f00;
        stroke:rgb(0,0,255);
        stroke-width:10;
        opacity:0.4;
      }
    </style>
</head>
<body>
    <svg>
      <rect x="10" width="90" height="90" style="fill:#00f" />
      <rect x="110" y="0" width="90" height="90" style="opacity:0.3"/>
      <rect x="10" y="100" width="90" height="90" style="stroke-width:10;stroke:rgba(255,0,0,0.5)"/>
      <rect id="cssControl" x="110" y="100" width="90" height="90" />
    </svg>
</body>
</html>
```

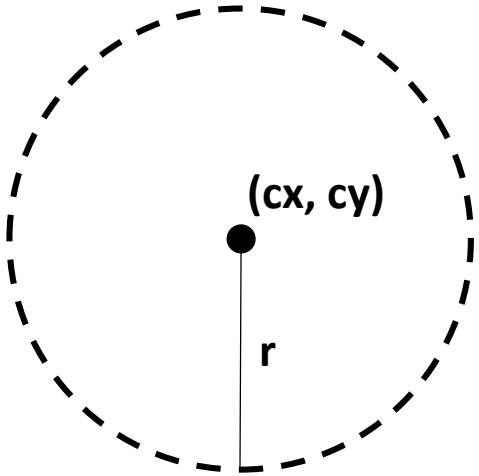


Scalable Vector Graphics

원 그리기

- 원 그리기 위한 요소 : <circle>

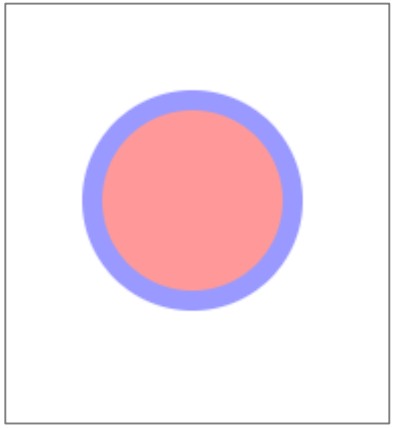
```
<circle
   cx="[원 중점의 X 좌표]"
   xy="[원 중점의 Y 좌표]"
   r="[원의 반지름]"
>
```



Scalable Vector Graphics

원 그리기

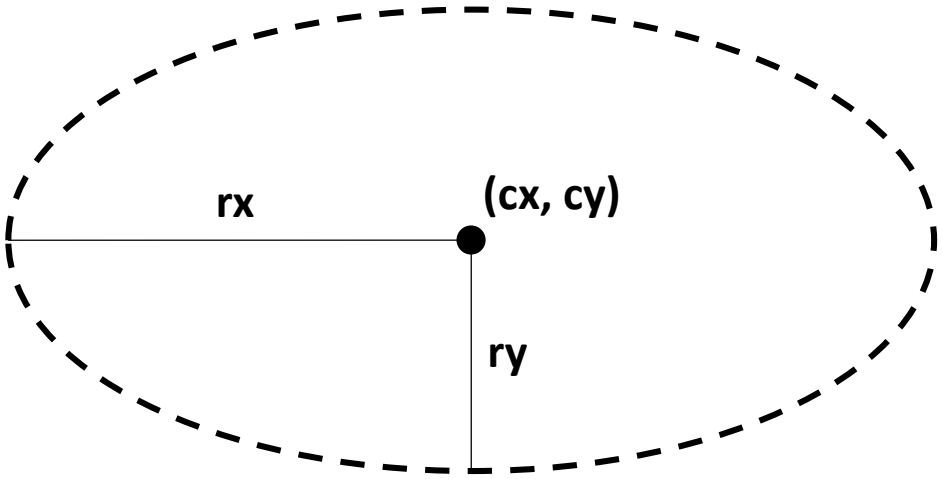
```
<!DOCTYPE HTML>
<html lang="en">
<head>
    <style>
      svg{
        width:400px;
        height:200px;
      }
      #cssControl{
        fill:#f00;
        stroke:rgb(0,0,255);
        stroke-width:10;
        opacity:0.4;
      }
    </style>
            </head>
            <body>
    <svg>
      <circle id="cssControl" cx="100" cy="100" r=50 />
    </svg>
</body>
</html>
```



타원 그리기

- 타원 그리기 위한 요소 : <ellipse>

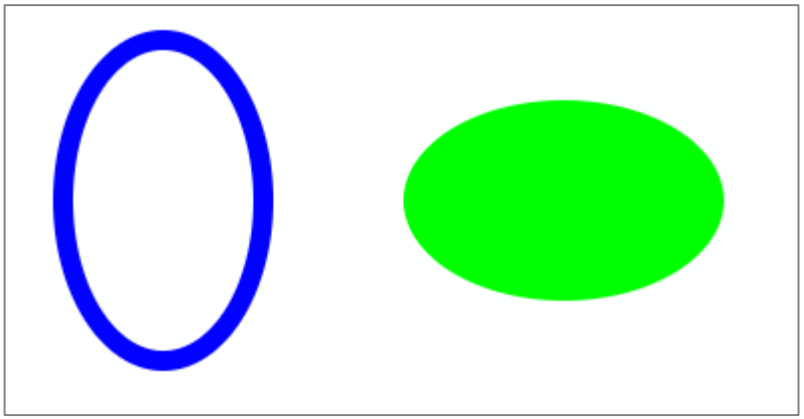
```
<ellipse
   cx="[타원 중점의 x 좌표]"
   cy="[타원 중점의 x 좌표]"
   rx="[타원 횡축 반지름]"
   ry="[타원 종축 반지름]"
>
```



Scalable Vector Graphics

타원 그리기

```
<!DOCTYPE HTML>
<html lang="en">
<head>
    <style>
        svg{
            width:400px;
            height:200px;
        }
        #ellipse01{
            fill:#fff;
            stroke:rgb(0,0,255);
            stroke-width:10;
        }
        #ellipse02{
            fill:#0f0;
        }
    </style>
</head>
<body>
    <svg>
        <ellipse id="ellipse01" cx="100" cy="100" rx="50" ry="80"/>
        <ellipse id="ellipse02" cx="300" cy="100" rx="80" ry="50"/>
    </svg>
</body>
</html>
```

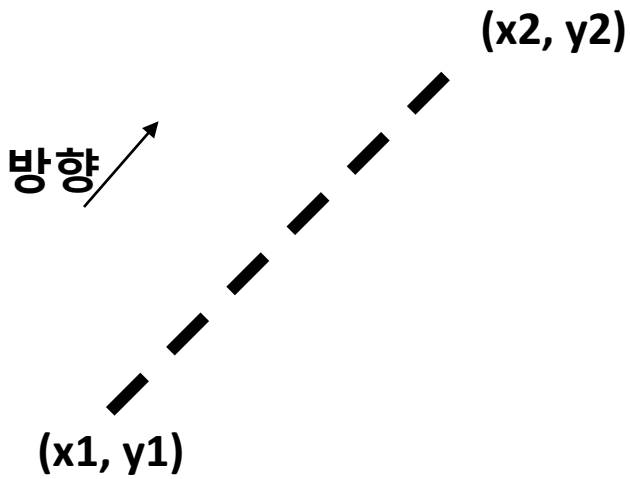


Scalable Vector Graphics

직선 그리기

- 직선 그리기 위한 요소 : <line>

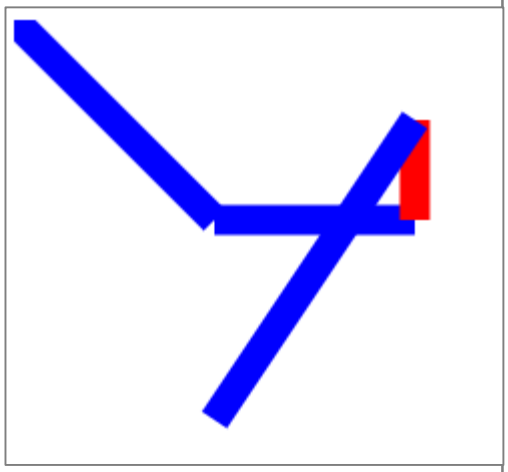
```
<line
   x1="[직선 시점의 X좌표]"
   y1="[직선 시점의 Y좌표]"
   x2="[직선 종점의 X좌표]"
   y2="[직선 종점의 Y좌표]"
>
```



Scalable Vector Graphics

직선 그리기

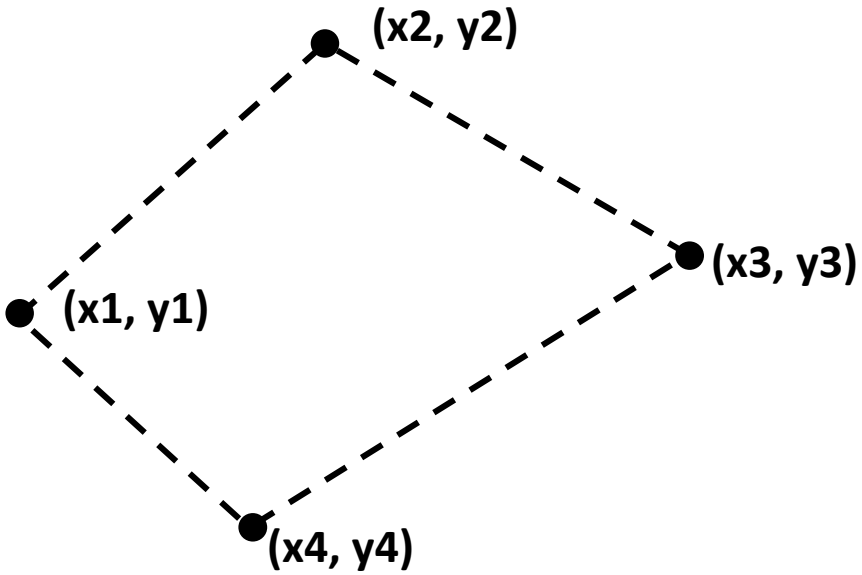
```
...
    <style>
        svg{
            width:600px;
            height:400px;
        }
        .lineClass{
            stroke:#00f;
            stroke-width:15;
        }
        #lineId{
            stroke:#f00;
            stroke-width:15;
        }
    </style>
</head>
<body>
    <svg>
        <line class="lineClass" x1="0" y1="0" x2="100" y2="100" />
        <line class="lineClass" x1="100" y1="100" x2="200" y2="100" />
        <line id="lineId" x1="200" y1="100" x2="200" y2="50" />
        <line class="lineClass" x1="200" y1="50" x2="100" y2="200" />
</svg>
</body>
</html>
```



다각형 그리기

- 다각형 그리기 위한 요소 : <polygon>

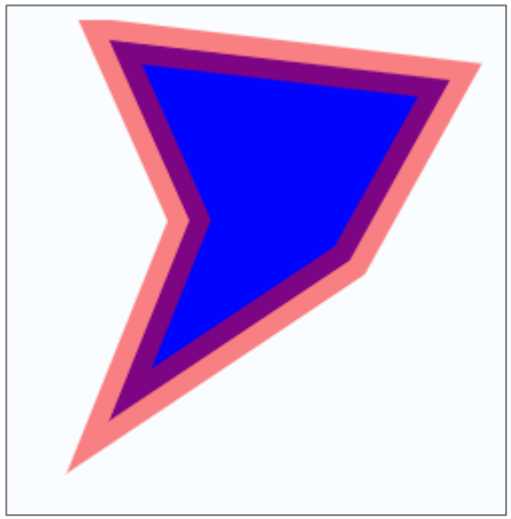
```
<polygon
   points="[다각형을 구성하는 x, y좌표] …"
>
```



Scalable Vector Graphics

다각형 그리기

```
<!DOCTYPE HTML>
<html lang="en">
<head>
    <style>
        svg{
            width:400px;
            height:400px;
        }
        #polygonSample{
            fill:#00f;
            stroke:rgba(255,0,0,0.5);
            stroke-width:20;
        }
    </style>
</head>
<body>
    <svg>
        <polygon id="polygonSample"
                points="80, 10, 250, 30, 200, 120, 80, 200, 120, 100"/>
    </svg>
</body>
</html>
```



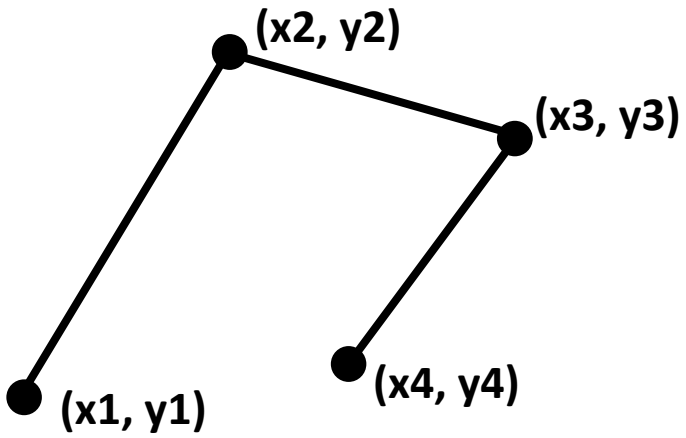
stroke??

Scalable Vector Graphics

다각선 그리기

- 다각형 그리기 위한 요소 : <polyline>

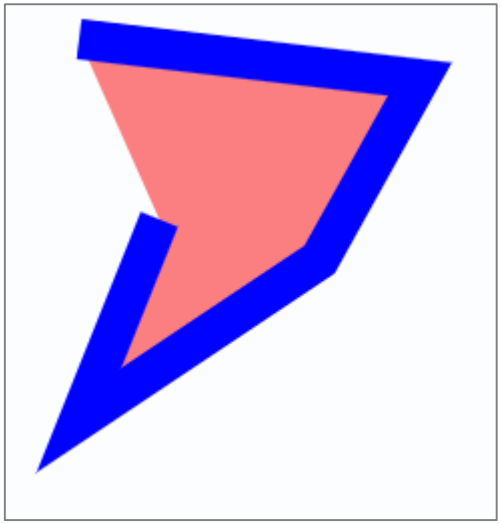
```
<polyline
   points="[다각선을 구성하는 x, y좌표] …"
>
```



Scalable Vector Graphics

다각선 그리기

```
<!DOCTYPE HTML>
<html lang="en">
<head>
    <style>
        svg{
            width:400px;
            height:400px;
        }
        #polylineSample{
            fill:rgba(255,0,0, 0.5);
            stroke:#00f;
            stroke-width:20;
        }
    </style>
</head>
<body>
    <svg>
        <polyline id="polylineSample"
                points="80, 10, 250, 30, 200, 120, 80, 200, 120, 100"/>
    </svg>
</body>
</html>
```



더 복잡한선 그리기

- 직선과 곡선으로 이루어진 복잡한 선을 그리기 위해 path 요소 사용

```
<path
   d="[Path Code]"
>
```

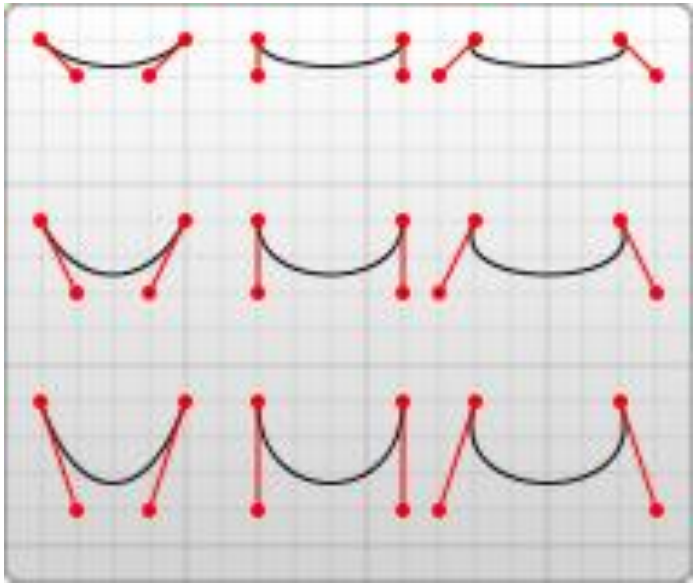
- Path Code?

기능	Path code 형식	Code 입력값
선의 시점 이동	M[X] [Y]	이동할 X, Y좌표
직선 그리기	L[X] [Y]	선 그릴 X, Y좌표
수평선 그리기	H[X]	수평선 그릴 X 좌표
수직선 그리기	V[Y]	수직선 그릴 Y좌표
3차 베지어 곡선	C[C1X] [C2Y] [C2X] [C2Y] [X] [Y]	조절점 {1,2}의 X, Y좌표, 커브 종점의 X, Y좌표
단순 커브 그리기	S[C1X] [C1Y] [X] [Y]	조절점 X, Y 커브 종점의 X, Y 좌표
2차 베지어 곡선	Q[C1X] [C1Y] [X] [Y]	조절점 X, Y 곡선 종점의 X, Y 좌표
베지어 곡선	T[X] [Y]	곡선 종점의 X, Y 좌표
타원 호	A[RX] [RY] [DEGX] [FUNC] [DIRECT] [X] [Y]	X 방향 반지름, Y 방향 반지름, X 축에 대한 경사도, 호 생성 방법, 회전 방향, 호 끝점의 X, Y 좌표
path 닫기	Z	

Scalable Vector Graphics

더 복잡한선 그리기

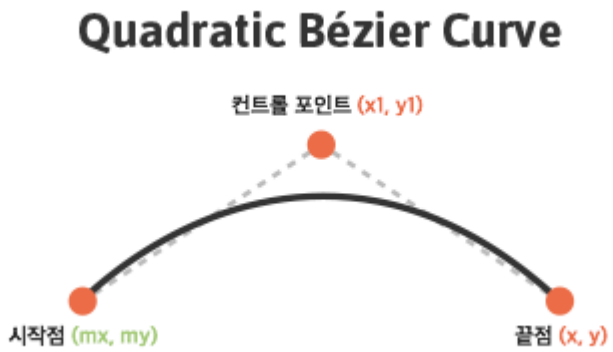
- 3차원 베지어 곡선 : c 와 s

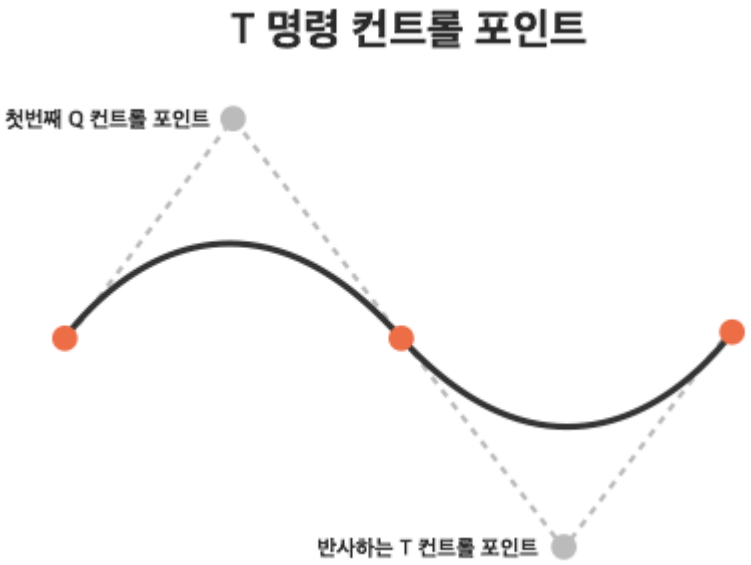


Scalable Vector Graphics

더 복잡한선 그리기

- 2차원 베지어 곡선 : Q 와 T



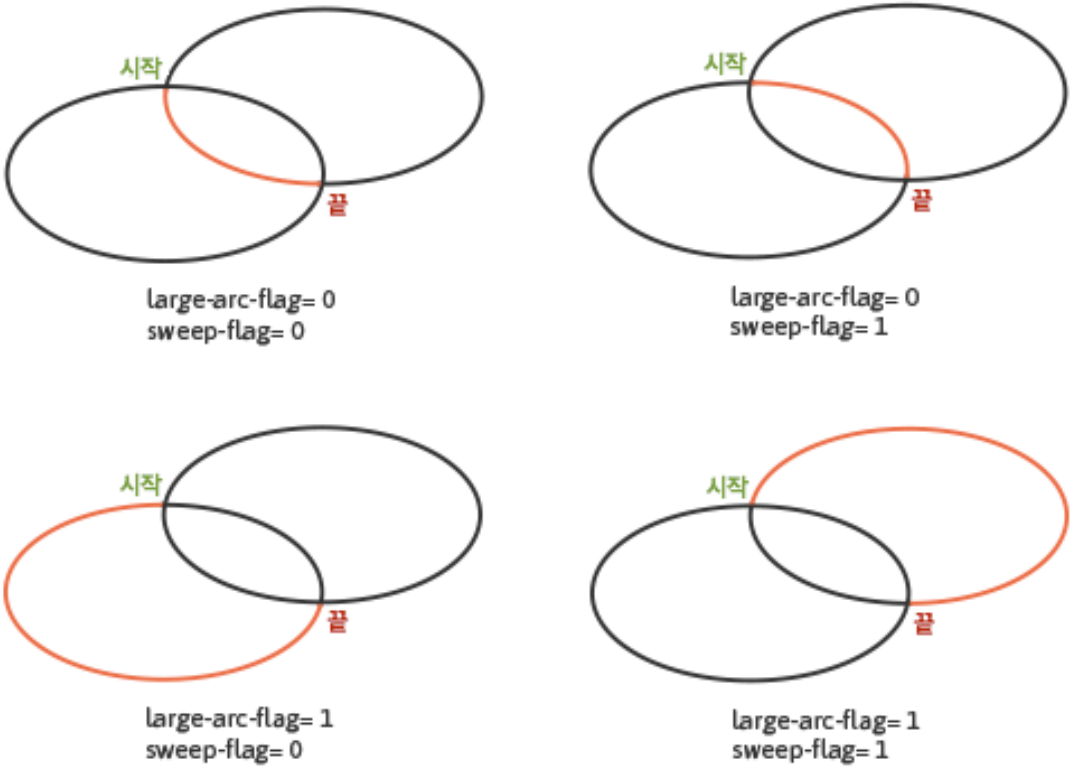


더 복잡한선 그리기

- 타원 호 그리기

타원 호	A[RX] [RY] [DEGX] [FUNC] [DIRECT] [X] [Y]	X 방향 반지름, Y 방향 반지름, X 축에 대한 경사도, 호 생성 방법, 회전 방향, 호 끝점의 X, Y 좌표

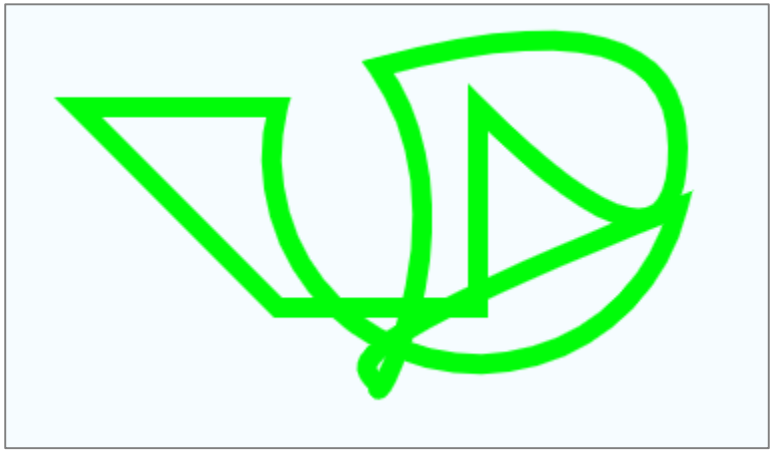
타원형 호 그리기



Scalable Vector Graphics

다각선 그리기

```
…
    <style>
       svg{
          width:400px;
          height:400px;
       }
       #pathSample{
          fill:transparent;
          stroke:#0f0;
          stroke-width:10;
       }
    </style>
</head>
<body>
    <svg>
       <path id="pathSample"
          d="M50 50          L150 150          H250          V50
            Q350 150 350 70          T200 30          C250 100 200 200 200 190
            S150 180 350 100          A10 10 0 1 1 150 50          Z"
          />
    </svg>
</body>
</html>
```



문자열 표시

- 문자열 표시를 위한 요소 : <text>

```
<text
   x="[문자열 위치의 X 좌표]"
   y="[문자열 위치의 Y 좌표]">
            [표시할 문자열]
</text>
```

- 문자열의 경우 그룹핑이 가능, 문자열 그룹을 하기 위해서는 tspan 요소 사용

```
<tspan
   dx="[원래 문자열 위치에서 변경할 X 좌표 거리]"
   dy="[원래 문자열 위치에서 변경할 Y좌표 거리]">
            [그룹화하는 문자열]
</text>
```

Scalable Vector Graphics

문자열 표시

```
<!DOCTYPE HTML>
<html lang="en">
<head>
    <style>
        svg{
            width:400px;
            height:400px;
        }
        #coForward{
            fill:#0099d8;
            font-size:40px;
            font-style:normal;
            font-weight:bold;
            font-family:Trebuchet MS, Arial,
Helvetica, sans-serif;
        }
        #coForward tspan{
            fill:#0850a0;
            font-variant:small-caps;
        }
        #nscreen{
            font-size:25px;
            font-family:Trebuchet MS, Arial, Helvetica,
sans-serif;
            fill:transparent;
            stroke:#096;
            stroke-width:2;
        }
    </style>
</head>
<body>
    <svg>
        <text id="textSample" x="60" y="100">
            <tspan id="coForward">
                co
                <tspan>Forward</tspan>
            </tspan>
            <tspan id="nscreen" dx="-150" dy="25">
                Web on The N Screen
            </tspan>
        </text>
    </svg>
</body>
</html>
```

co FORWARD
Web on The N Screen

Scalable Vector Graphics

문자열이 흐르는 패스

- 문자열 흐르는 패스 표시를 위한 요소 : <textPath>

```
<textPath
   xlink:href="[#패스 아이디]"
>
            [영향을 받는 문자열 또는 문자열 요소]
</textPath>
```

- textPath를 사용하기 위해서는 path 요소 필요
- Xlink:href 속성에 path로 작성된 요소 id 값 입력하면 path 요소의 형태로 텍스트가 흐름

Scalable Vector Graphics

문자열이 흐르는 패스

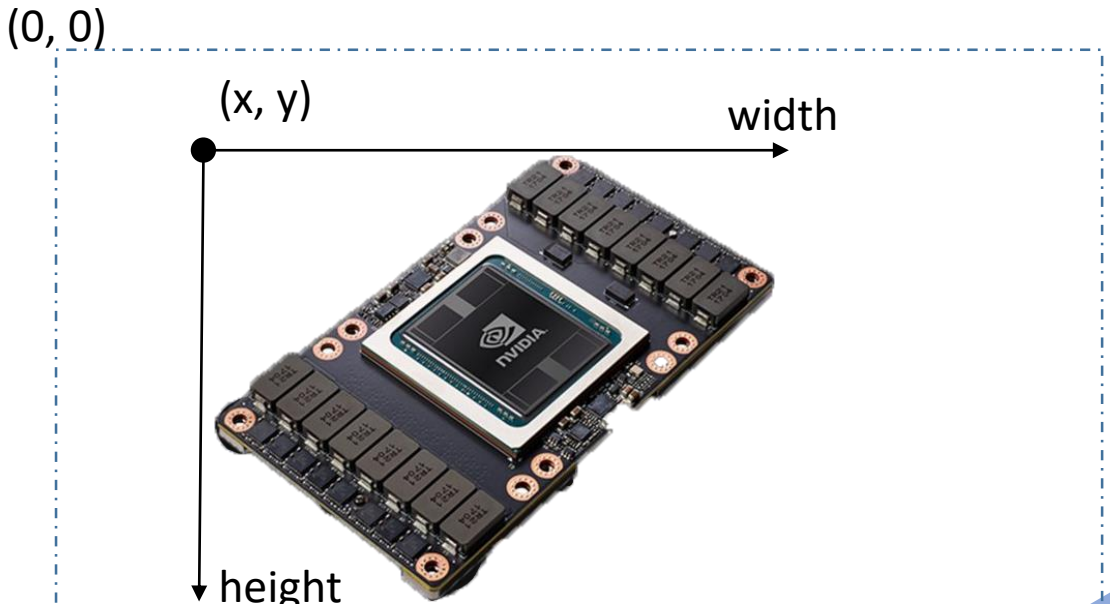
```
<!DOCTYPE HTML>
<html lang="en">
<head>
    <style>
        svg{
            width:400px;
            height:400px;
        }
        #nscreen{
            font-size: 20px;
            font-family: Trebuchet MS, Arial, Helvetica, sans-serif;
            font-weight: bold;
            fill: transparent;
            stroke: #096;
            stroke-width: 1;
            text-align: center;
        }
    </style>
</head>
<body>
  <svg>
  <defs>
    <path id="textPath"
        d="M80 150
          a10, 10 0 0,1 250, 150" />
  </defs>
  <text id="nscreen">
    <textPath xlink:href="#textPath" >
      coForward : Web on The N Screen
    </textPath>
  </text>
  </svg>
</body>
</html>
```

Scalable Vector Graphics

이미지 삽입

- **이미지를 위한 요소 : <image>**

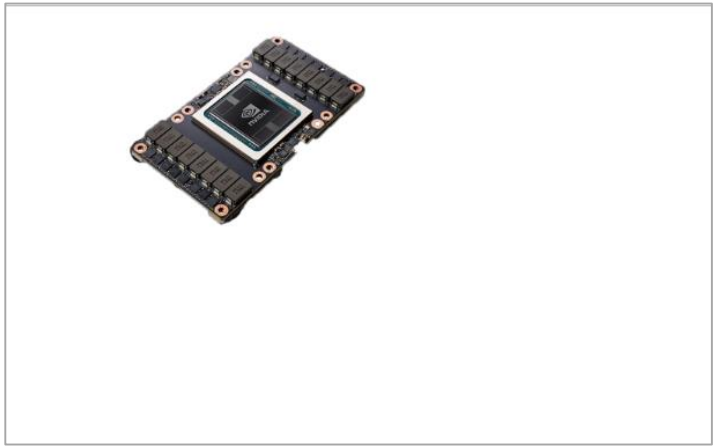
```
<image
   x="[이미지가 삽입되는 좌측 상단의 X좌표]"
   y="[이미지가 삽입되는 좌측 상단의 Y좌표]"
   width="[이미지가 삽입되는 폭]"
   height="[이미지가 삽입되는 높이]"
   xlink:href="[삽입되는 이미지 경로]"
/>
```



Scalable Vector Graphics

이미지 삽입

```
<!DOCTYPE HTML>
<html lang="en">
<head>
    <style>
        svg{
            width:400px;
            height:400px;
        }
    </style>
</head>
<body>
  <svg>
    <image x="100" y="5" width="204" height="189"
        xlink:href="img/text.png" />
  </svg>
</body>
</html>
```



형태 변경

- 형태 변경을 위한 속성

SVG 요소 형태를 변경하기 위해 transform 속성 지정

기능	속성값 형식	속성값에 따른 입력 수치
수평 이동	Translate(tx, ty)	이동시킬 x, y좌표
크기 변경	Scale(sx, sy)	스케일 변경할 x, y 축의 비율
X축 방향 기울임	**skewX(d)**	**X축에 대한 기울임 각도**
Y축 방향 기울임	**skewY(d)**	**Y축에 대한 기울임 각도**
회전	Rotate(d)	SVG 원점을 기준으로 회전할 각도
일괄 변환	Matrix(변환 매트릭스)	일괄 변환을 위한 변환 매트릭스

Scalable Vector Graphics

형태 변경

```
<!DOCTYPE HTML>
<html lang="en">
<head>
    <style>
        svg{
            width:400px;
            height:400px;
        }
        #orgImg{
            opacity:0.4;
        }
    </style>
</head>
<body>
  <svg>
    <image x="100" y="5" width="204" height="189" id="orgImg"
        xlink:href="img/text.png" />
    <image x="150" y="5" width="204" height="189"
         xlink:href="img/text.png"
        transform="skewX(-30) rotate(15)"/>
  </svg>
…
```

그레이디언트 (선형)

- **선형 그레이디언트를 위한 요소 : linearGradient**

```
<linearGradient
   x1="[그레이디언트 적용 시점의 X축 비율]"
   y1="[그레이디언트 적용 시점의 Y축 비율]"
   x2="[그레이디언트 적용 종점의 X축 비율]"
   y2="[그레이디언트 적용 종점의 Y축 비율]"
   spreadMethod="[그레이디언트의 채움 방식 keyword(pad, repeat, reflect)]"
   gradientTransform="[transform 속성과 동일한 방법으로 그레이디언트를 변형]"
/>
```

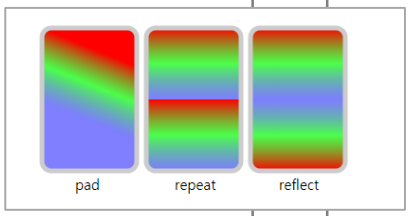
- **색상 변환점을 지정하기 위해서는 자식 요소로 stop 요소를 작성해야함**

```
<linearGradient>
   <stop
            offset="[그레이디언트 내에서 색상 변화점 위치의 거리 비율]"
            stop-color="[색상 변환점의 변환 색상]"
            stop-opacity="[변환점 영역의 투명도]"
   />
</linearGradient>
```

Scalable Vector Graphics

그레이디언트 (선형)

```
<!DOCTYPE HTML>
<html lang="en">
<head>
    <style>
       svg{
          width:600px;
          height:400px;
       }
       rect{
          stroke:#ccc;
          stroke-width:5;
       }
       #rect01{
          fill:url(#lGradientSample01);
       }
       #rect02{
          fill:url(#lGradientSample02);
       }
       #rect03{
          fill:url(#lGradientSample03);
       }
       text{
          font-size:15px;
          font-weight:blod;
       }
    </style>
</head>
<body>
    <svg>
      <defs>
       <linearGradient id="lGradientSample01"
                x1="0%" y1="0%"
                x2="0%" y2="50%"
                spreadMethod="pad"
                gradientTransform="rotate(15)">
```




```
      <stop offset="0%" stop-color="#f00" stop-opacity="1"/>
      <stop offset="50%" stop-color="#0f0" stop-opacity="0.7"/>
      <stop offset="100%" stop-color="#00f" stop-opacity="0.5"/>
    </linearGradient>
    <linearGradient id="lGradientSample02"
                x1="0%" y1="0%"
                x2="0%" y2="50%"
                spreadMethod="repeat">
      <stop offset="0%" stop-color="#f00" stop-opacity="1"/>
      <stop offset="50%" stop-color="#0f0" stop-opacity="0.7"/>
      <stop offset="100%" stop-color="#00f" stop-opacity="0.5"/>
    </linearGradient>
    <linearGradient id="lGradientSample03"
                x1="0%" y1="0%"
                x2="0%" y2="50%"
                spreadMethod="reflect">
      <stop offset="0%" stop-color="#f00" stop-opacity="1"/>
      <stop offset="50%" stop-color="#0f0" stop-opacity="0.7"/>
      <stop offset="100%" stop-color="#00f" stop-opacity="0.5"/>
    </linearGradient>
    </defs>
    <rect id="rect01" x="40" y="20" width="100" height="150"
        rx="10" ry="10" />
    <text x="75" y="190">pad</text>
    <rect id="rect02" x="150" y="20" width="100" height="150"
        rx="10" ry="10" />
    <text x="180" y="190">repeat</text>
    <rect id="rect03" x="260" y="20" width="100" height="150"
        rx="10" ry="10" />
    <text x="290" y="190">reflect</text>

  </svg>
</body>
</html>
```

그레이디언트 (원형)

- **원형 그레이디언트를 위한 요소 : radialGradient**

```
<radialGradient
   cx="[원형 그레이디언트 중점 X축 비율]"
   cy="[원형 그레이디언트 중점 Y축 비율]"
   fx="[원형 그레이디언트 초점 X축 비율]"
   fy="[원형 그레이디언트 초점 Y축 비율]"
   r="[원형 그레이디언트 반지름 비율]"
   spreadMethod="[그레이디언트의 채움 방식 keyword(pad, repeat, reflect)]"
   gradientTransform="[transform 속성과 동일한 방법으로 그레이디언트를 변형]"
/>
```

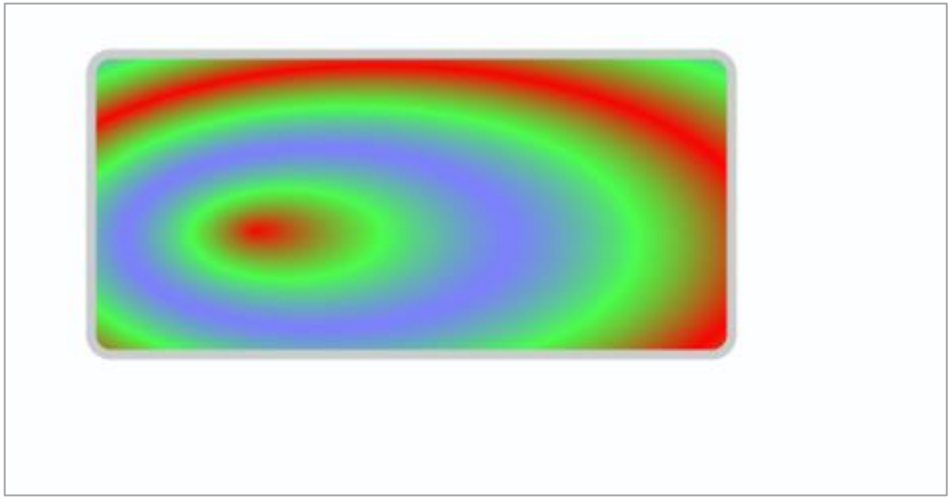
- **색상 변환점을 지정하기 위해서는 자식 요소로 stop 요소를 작성해야함**

```
<radialGradient>
   <stop
            offset="[그레이디언트 내에서 색상 변화점 위치의 거리 비율]"
            stop-color="[색상 변환점의 변환 색상]"
            stop-opacity="[변환점 영역의 투명도]"
   />
</radialGradient>
```

Scalable Vector Graphics

그레이디언트 (원형)

```
<!DOCTYPE HTML>
<html lang="en">
<head>
    <style>
        svg{
            width:600px;
            height:400px;
        }
        rect{
            stroke:#ccc;
            stroke-width:5;
        }
        #rect01{
            stroke:#CCC;
            stroke-width:5;
            fill:url(#rGradientSample01);
        }
    </style>
</head>
<body>
    <svg>
        <defs>
        <radialGradient id="rGradientSample01"
                    cx="50%" cy="50%"
                    fx="40%" fy="50%" r="30%"
                    spreadMethod="reflect"
                    gradientTransform="rotate(15)">
            <stop offset="0%" stop-color="#f00" stop-
opacity="1"/>
            <stop offset="50%" stop-color="#0f0" stop-
opacity="0.7"/>
            <stop offset="100%" stop-color="#00f" stop-
opacity="0.5"/>
        </radialGradient>
        </defs>
```



```
        <rect id="rect01" x="40" y="20" width="320" height="150"
            rx="10" ry="10" />

    </svg>
</body>
</html>
```

Scalable Vector Graphics

필터 효과

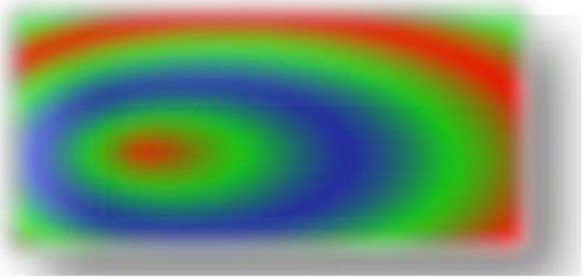
- **포토샵 이미지 특수 효과 적용과 비슷**
- **필터 적용을 위한 요소 : filter**
- **Filter 종류**
  <feBlend> <feColorMatrix> <feComponentTransfer> <feComposite>
  <feConvolveMatrix> <feDiffuseLighting> <feDisplacementMap> <feFlood>
  <feGaussianBlur> <feImage> <feMerge> <feMorphology> **<feOffset>**
  <feSpecularLighting> <feTile> <feTurbulence> <feDistantLight>
  <fePointLight> <feSpotLight>
- 블러 효과와 offset을 이용하여 그림자 효과 생성 예)

```
<filter>
  <feGaussianBlur
    stdDeviation="[흐려짐 정도값]"
   />
  <feOffset
    dx="가로 방향 이동 거리"
    dy="세로 방향 이동 거리"
  />
</filter>
```

Scalable Vector Graphics

필터 효과

```
<!DOCTYPE HTML>
<html lang="en">
<head>
    <style>
        svg{
            width:600px;
            height:400px;
        }
        rect{
            stroke:#ccc;
            stroke-width:5;
        }
        #rect01{
            stroke:#CCC;
            stroke-width:5;
            fill:url(#rGradientSample01);
            filter:url(#blur);
        }
        #shadowRect{
            fill:rgba(0, 0, 0, 0.7);
            filter:url(#shadow);
        }
    </style>
</head>
<body>
    <svg>
        <defs>
            <filter id="blur">
                <feGaussianBlur stdDeviation="5" />
            </filter>
            <filter id="shadow">
                <feGaussianBlur stdDeviation="15" />
                <feOffset dx="20" dy="20" />
            </filter>
```



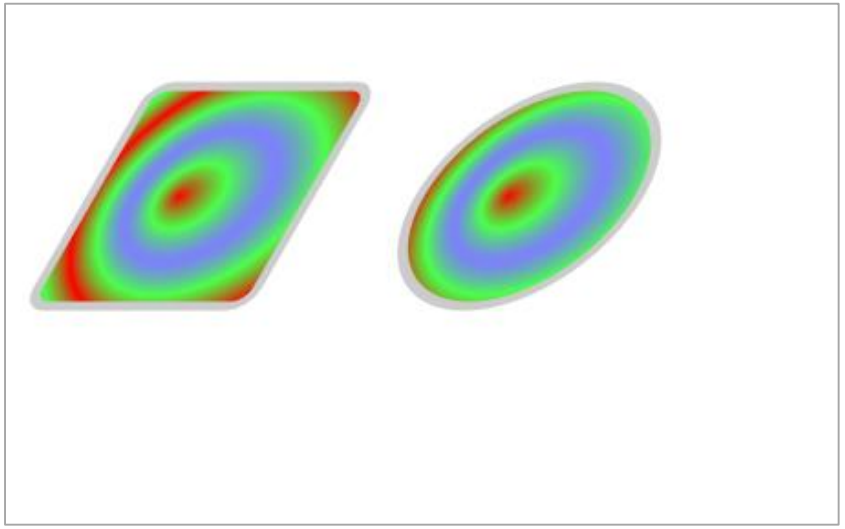
```
            <radialGradient id="rGradientSample01"
                    cx="50%" cy="50%"
                    fx="40%" fy="50%" r="30%"
                    spreadMethod="reflect"
                    gradientTransform="rotate(15)">
                <stop offset="0%" stop-color="#f00" stop-opacity="1"/>
                <stop offset="50%" stop-color="#0f0" stop-opacity="0.7"/>
                <stop offset="100%" stop-color="#00f" stop-opacity="0.5"/>
            </radialGradient>
        </defs>
        <rect id="shadowRect" x="40" y="20" width="320" height="150"
            rx="10" ry="10" />
        <rect id="rect01" x="40" y="20" width="320" height="150"
            rx="10" ry="10" />

    </svg>
</body>
</html>
```

SVG 요소 그룹핑

- 그룹핑을 위한 요소 : g
- g 요소로 묶인 SVG 요소는 하나의 요소처럼 취급할 수 있음

```
<!DOCTYPE HTML>
<html lang="en">
<head>
    <style>
        svg{
            width:600px;
            height:400px;
        }
        #groupTest{
            stroke:#ccc;
            stroke-width:5;
            fill:url(#rGradientSample01);
        }
    </style>
</head>
<body>
    <svg>
        <defs>
            <radialGradient id="rGradientSample01"
                    cx="50%" cy="50%"
                    fx="40%" fy="50%" r="30%"
                    spreadMethod="reflect">
                <stop offset="0%" stop-color="#f00" stop-opacity="1"/>
                <stop offset="50%" stop-color="#0f0" stop-opacity="0.7"/>
                <stop offset="100%" stop-color="#00f" stop-opacity="0.5"/>
            </radialGradient>
        </defs>
```



```
        <g id="groupTest" transform="skewX(-30)">
            <rect x="110" y="40" width="120" height="120"
                rx="10" ry="10" />
            <circle cx="350" cy="100" r="60" />
        </g>

    </svg>
</body>
</html>
```

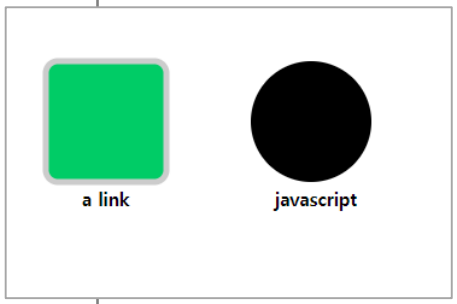
Scalable Vector Graphics

인터렉션

- DOM 구조가 유지되므로 각 요소를 편집하고 인터랙션을 부여하거나 SVG 가진 정보를 추출 가능

```
<!DOCTYPE HTML>
<html lang="en">
<head>
    <style>
        svg{
            width:600px;
            height:400px;
        }
        circle, rect{
            cursor:pointer;
        }
        rect{
            fill:#0c6;
            stroke:#ccc;
            stroke-width:5;
        }
        text{
            font-size:15px;
            font-weight:bold;
        }
        #interaction{
            stroke:#ccc;
            stroke-width:5;
            fill:url(#rGradientSample01);
        }
    </style>
```

```
    <script>
        function svgInteraction(svg){
            alert(svg.firstChild.nodeValue);
            svg.id="interaction";
        }
    </script>
</head>
<body>
```




```
    <svg>
        <defs>
        <radialGradient id="rGradientSample01"
                    cx="50%" cy="50%"
                    fx="40%" fy="50%" r="30%"
                    spreadMethod="reflect"
                    gradientTransform="rotate(15)">
            <stop offset="0%" stop-color="#f00" stop-opacity="1"/>
            <stop offset="50%" stop-color="#0f0" stop-opacity="0.7"/>
            <stop offset="100%" stop-color="#00f" stop-opacity="0.5"/>
        </radialGradient>
        </defs>
        <a xlink:href="http://lecture.engintruder.com" target="_blank">
            <rect x="60" y="50" width="100" height="100" rx="10" ry="10" />
        </a>
        <text x="90" y="170">a link</text>
        <circle cx="280" cy="100" r="50" onclick="svgInteraction(this)">
            "svg interaction Test"
        </circle>
        <text x="250" y="170">javascript</text>
    </svg>
</body>
</html>
```