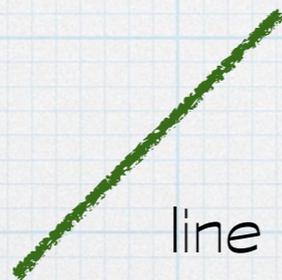
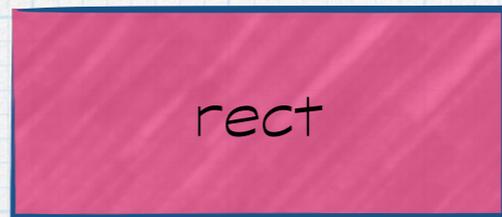
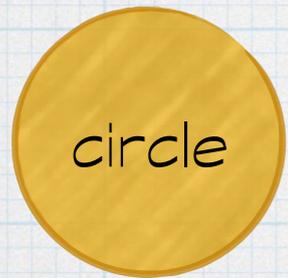


SVG

Scalable Vector Graphics

SVG SHAPES



text

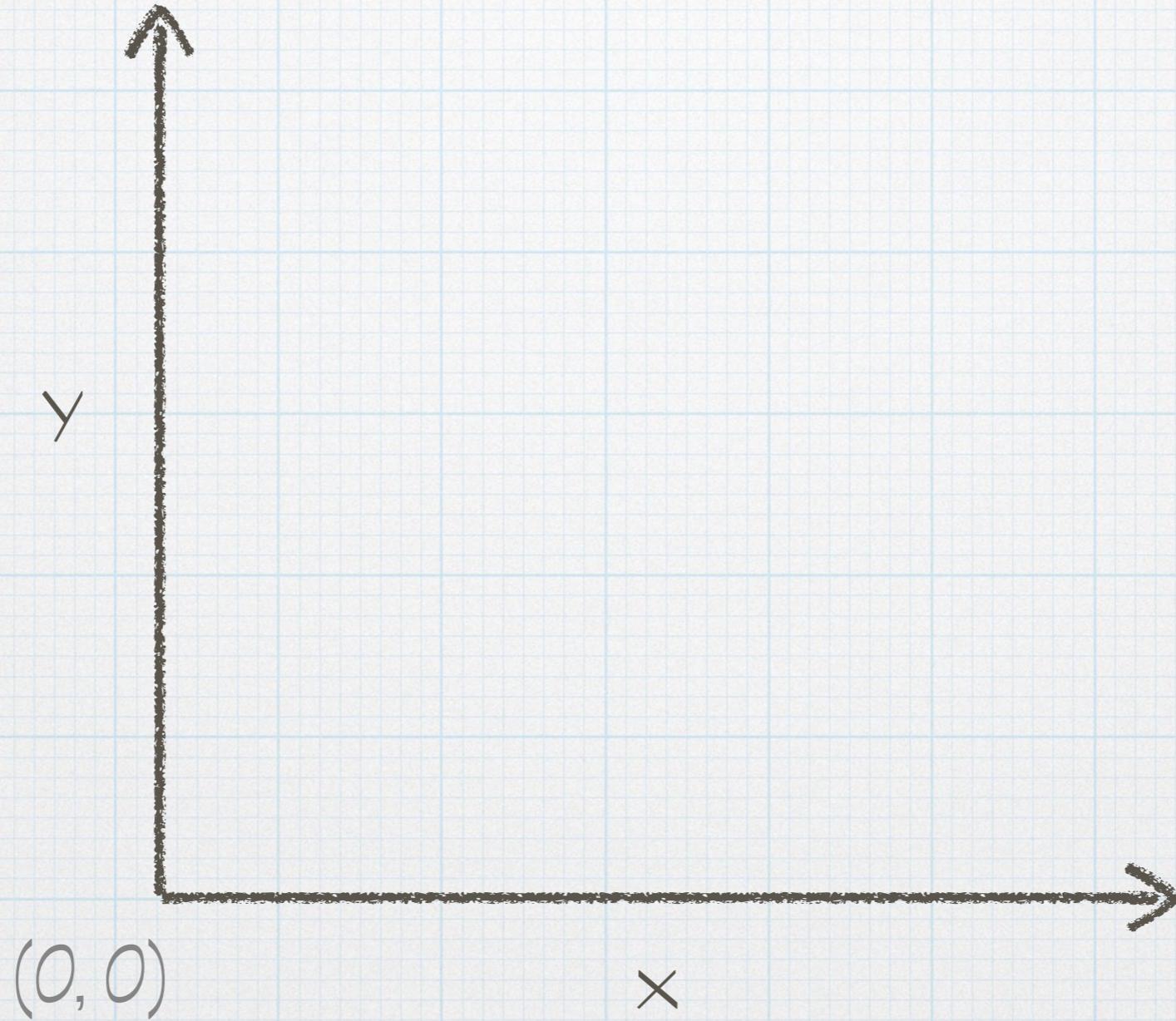
Scalable vector graphics are shapes that are composed not of dots (or “pixels”) but rather of *lines* and *fills*. The “scalable” part of SVG means that the graphics can be resized without losing resolution — that is, they don’t ever get pixelated because they were never made out of pixels to begin with.

We create SVG graphics by writing XML. That XML needs to be inside an `<svg>` container in our HTML document. The SVG validates just like HTML does.

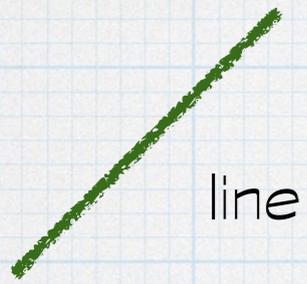
D3 draws all its graphics with SVG, so we need to learn how to add SVG shapes to our webpage.

To repeat: SVG is XML!

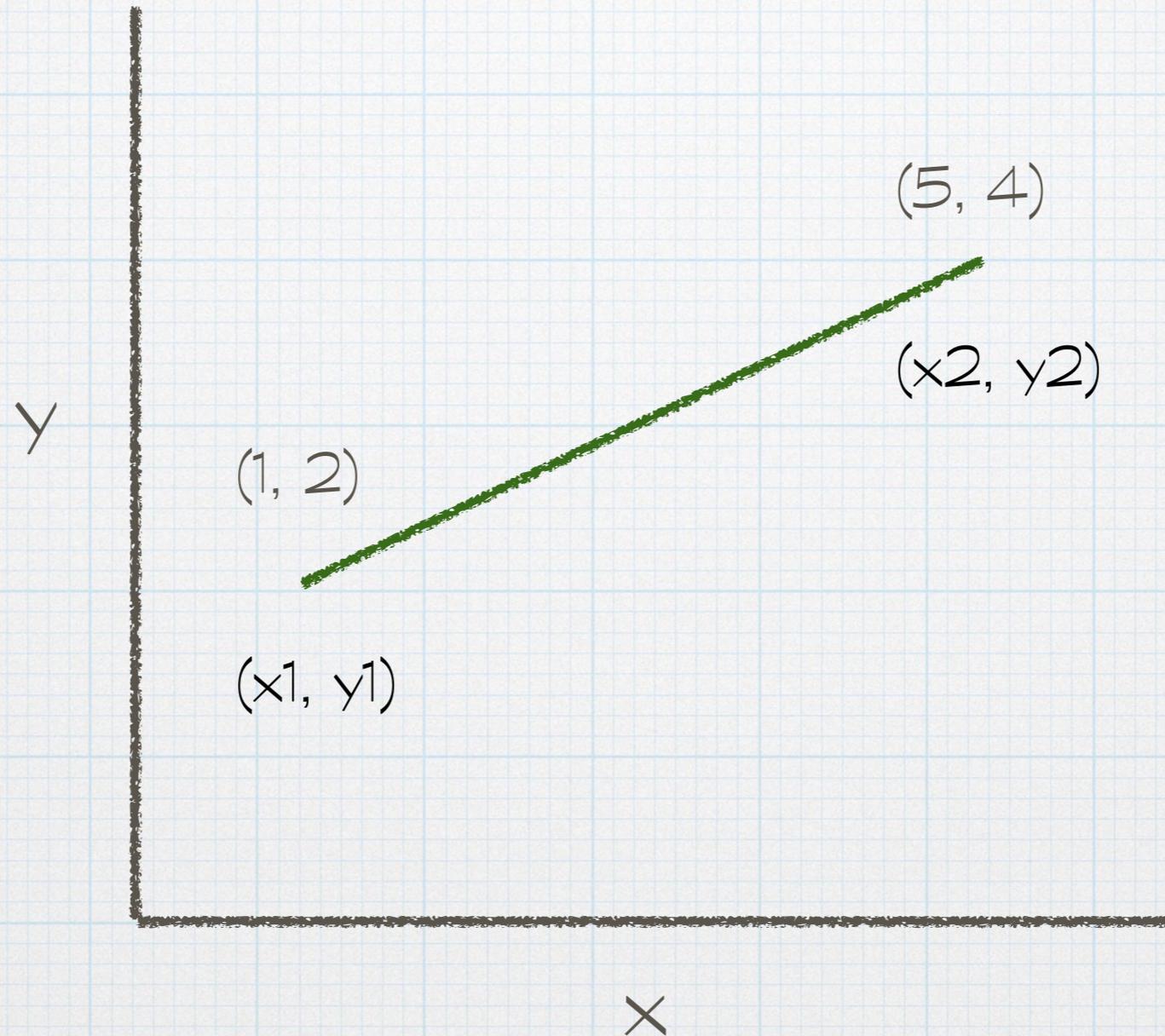
COORDINATE GRID



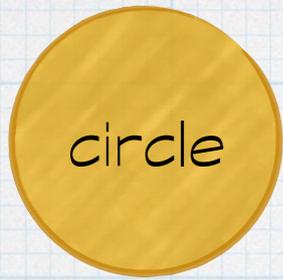
SVG LINE



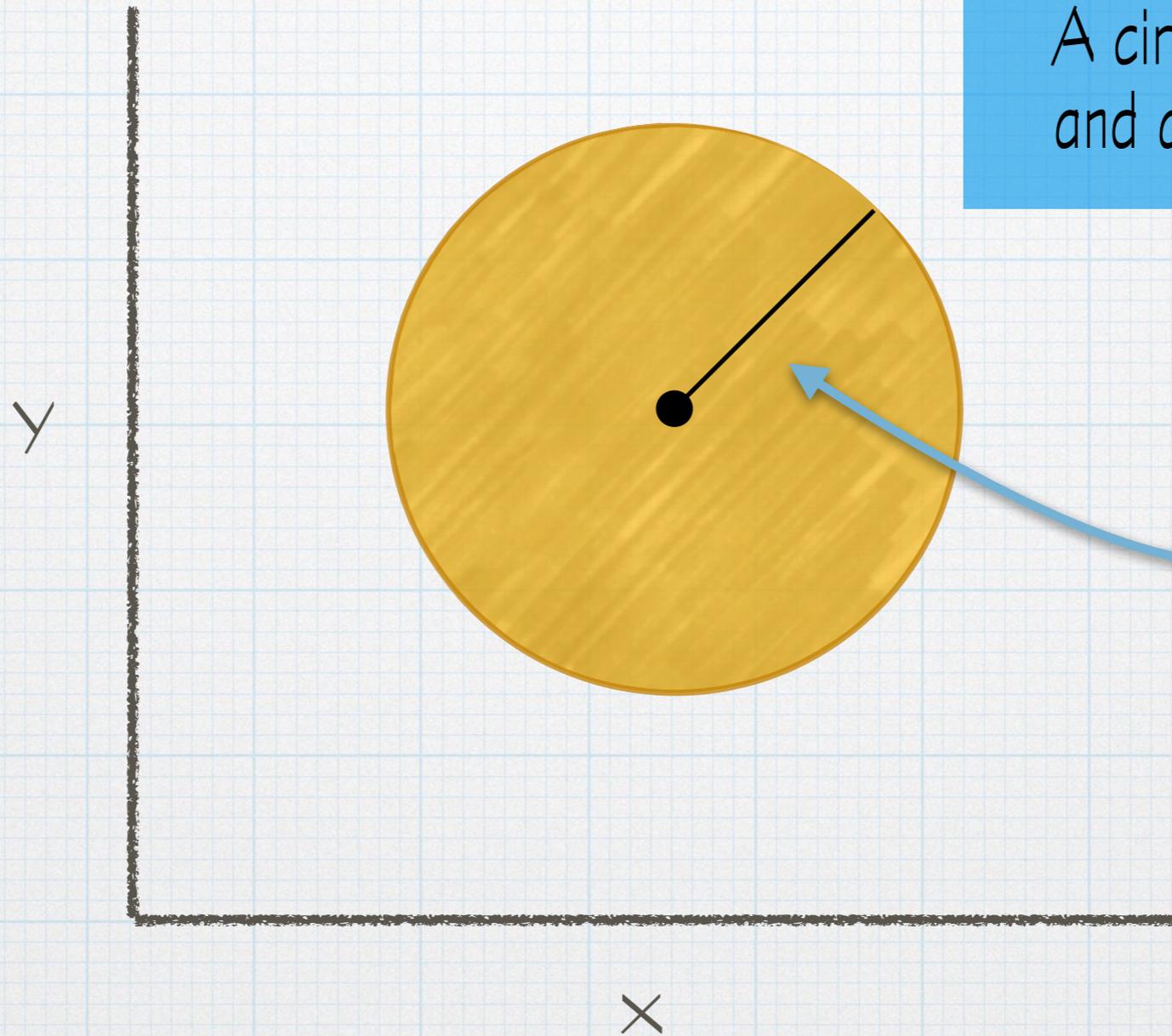
line



```
<line x1="1" y1="2" x2="5" y2="4" stroke="green" stroke-width="2" />
```



SVG CIRCLE



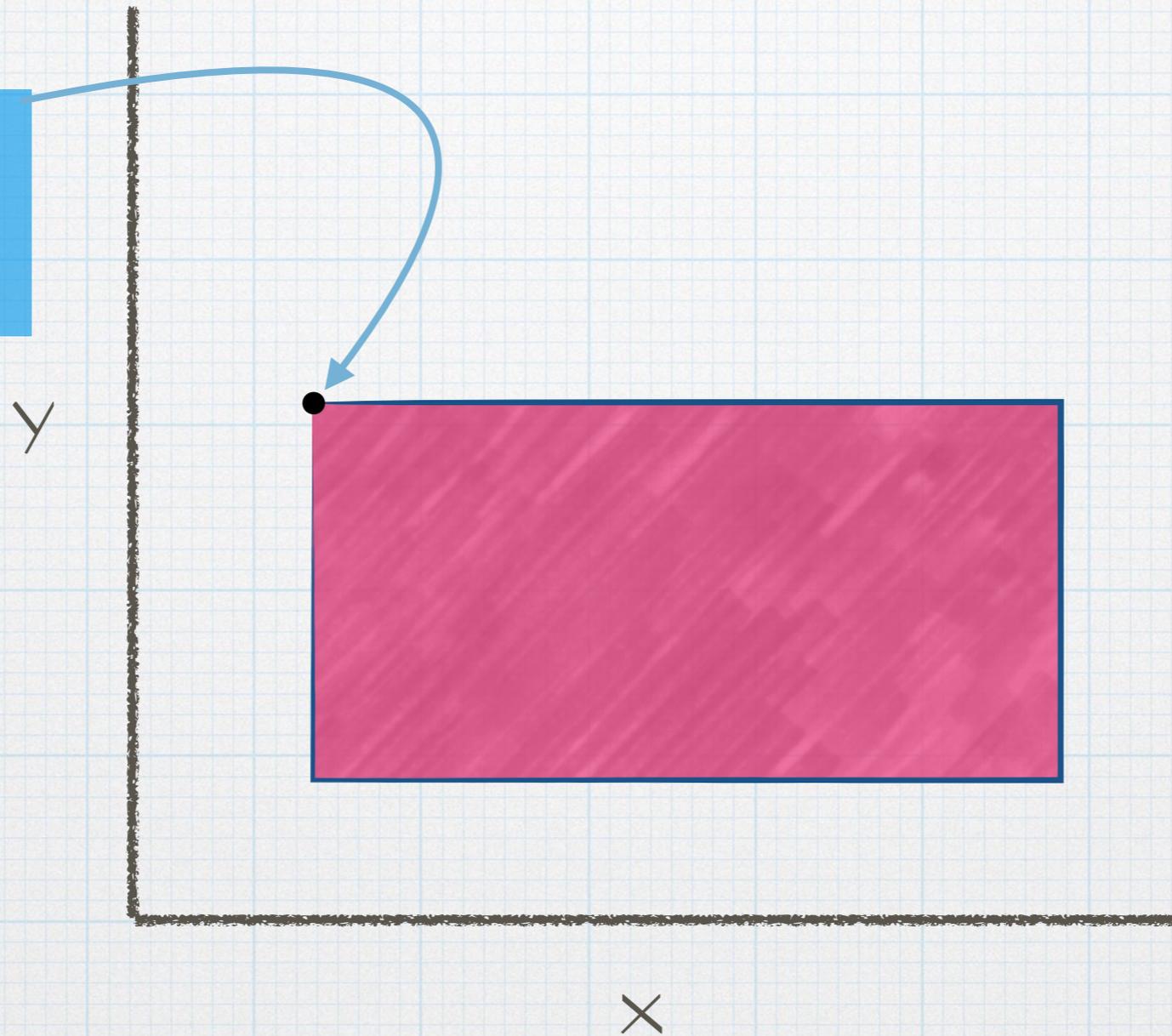
A circle has a centre and a radius.

```
<circle cx="4" cy="3" r="2" fill="yellow" stroke="yellow" stroke-width="2" />
```

rect

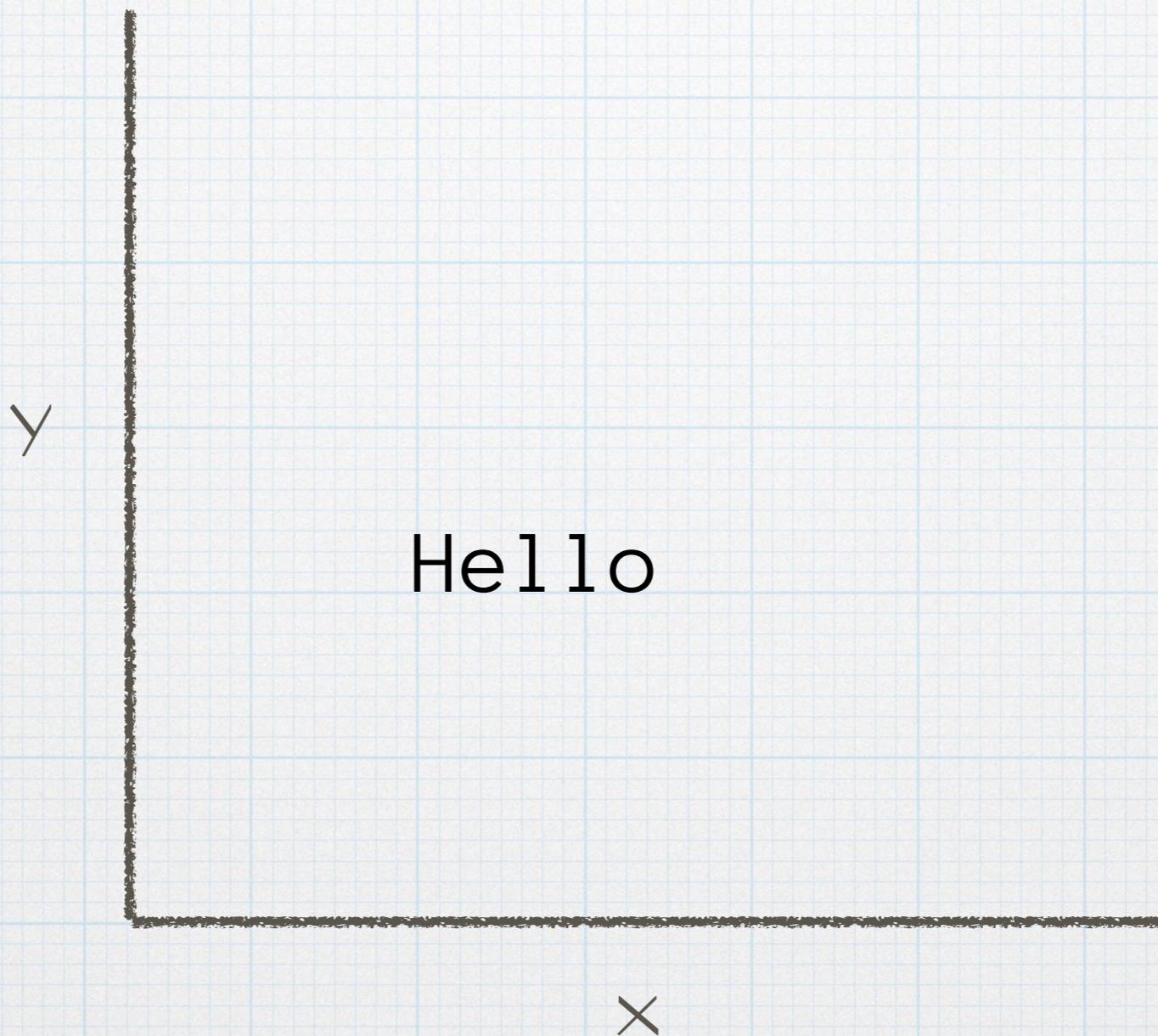
SVG RECTANGLE

x and y are the top-left corner



```
<rect x="2" y="3" height="3" width="4" fill="red" stroke="grey" stroke-width="2" />
```

SVG TEXT



```
<text x="2" y="10">Hello!</text>
```

SVG TEXT

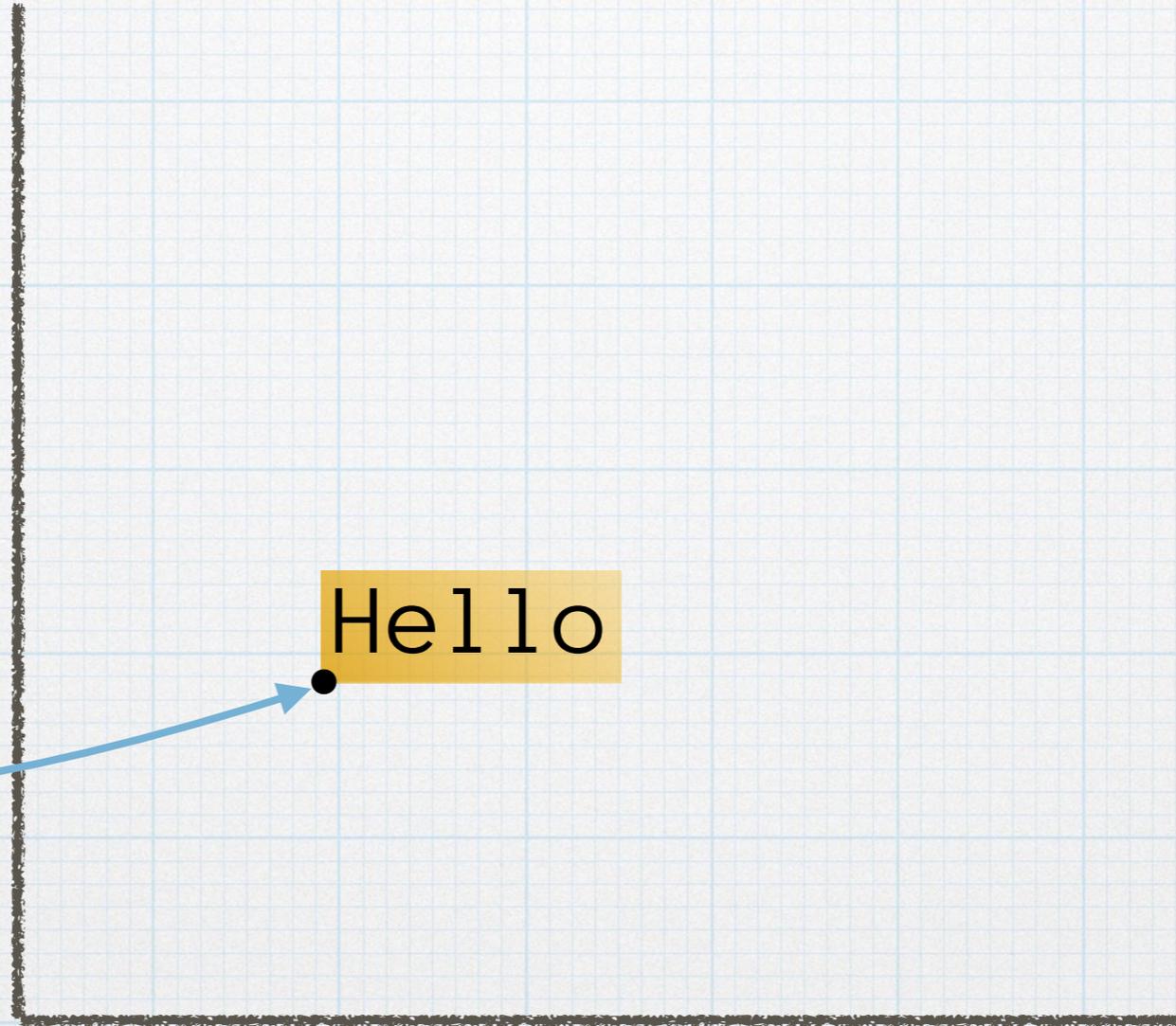
x and y are the
bottom-left corner

y

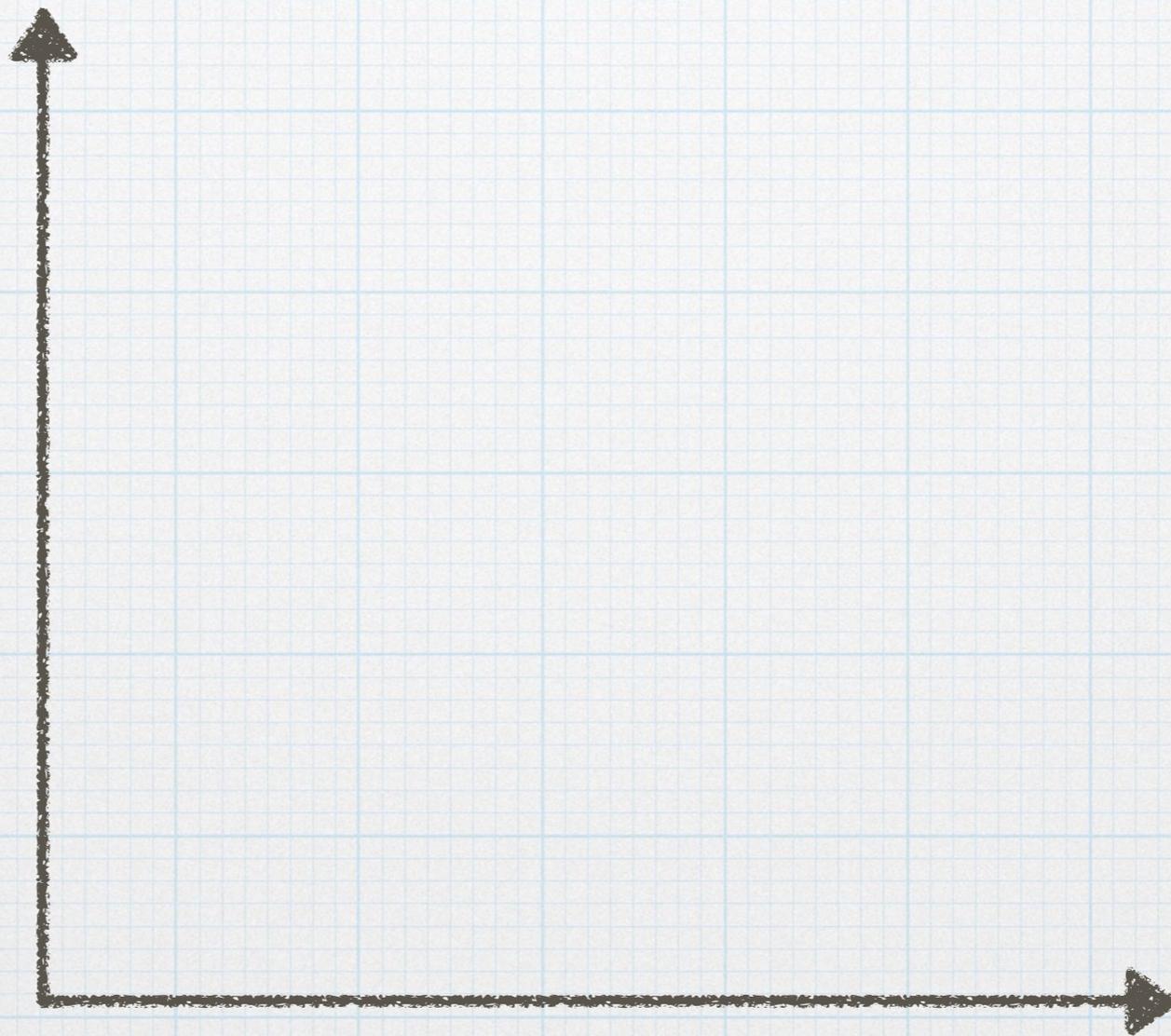
Hello

x

```
<text x="2" y="10">Hello!</text>
```



SVG COORDINATE GRID



y

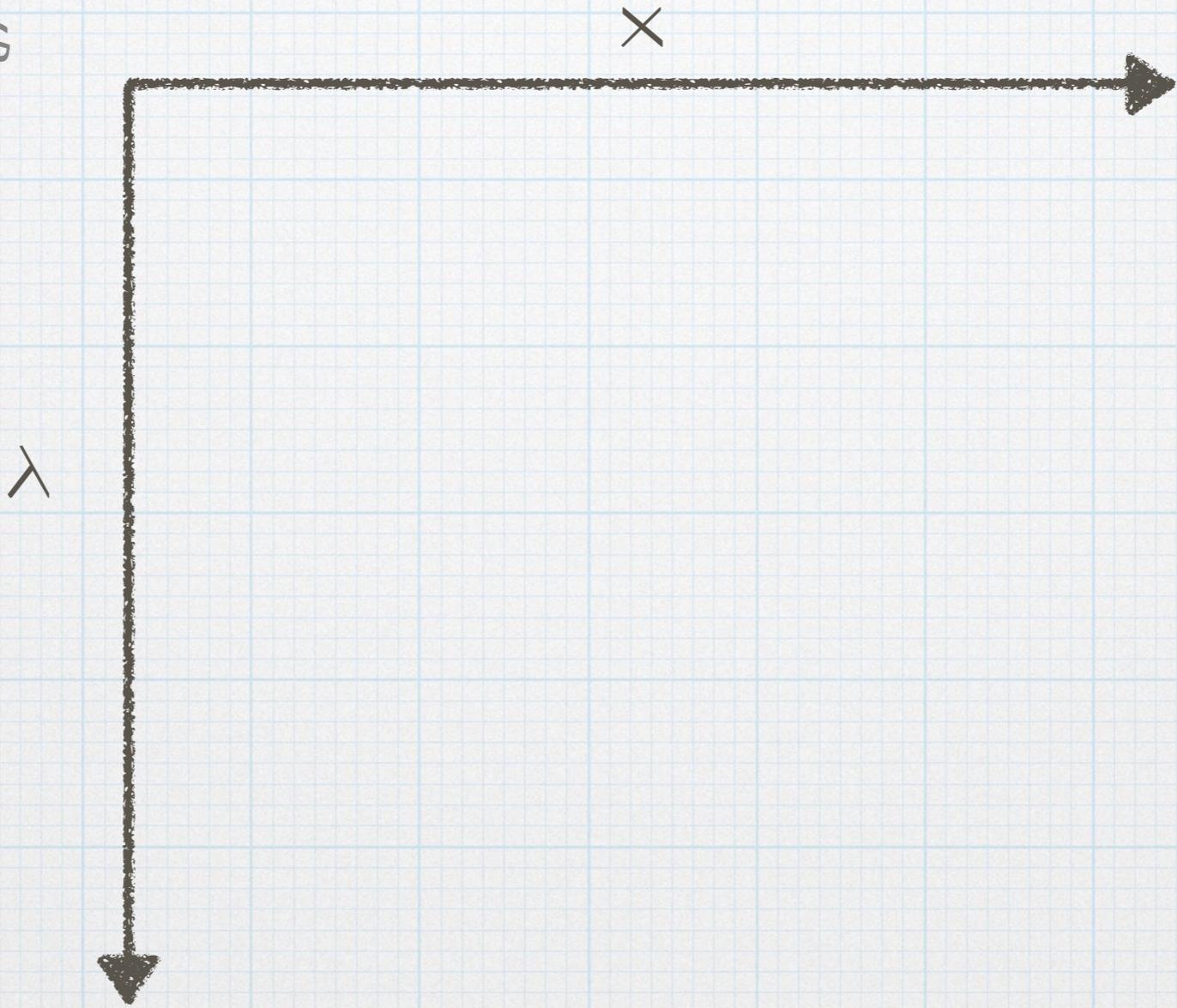
$(0, 0)$ is the origin

x

2D COORDINATE AXIS

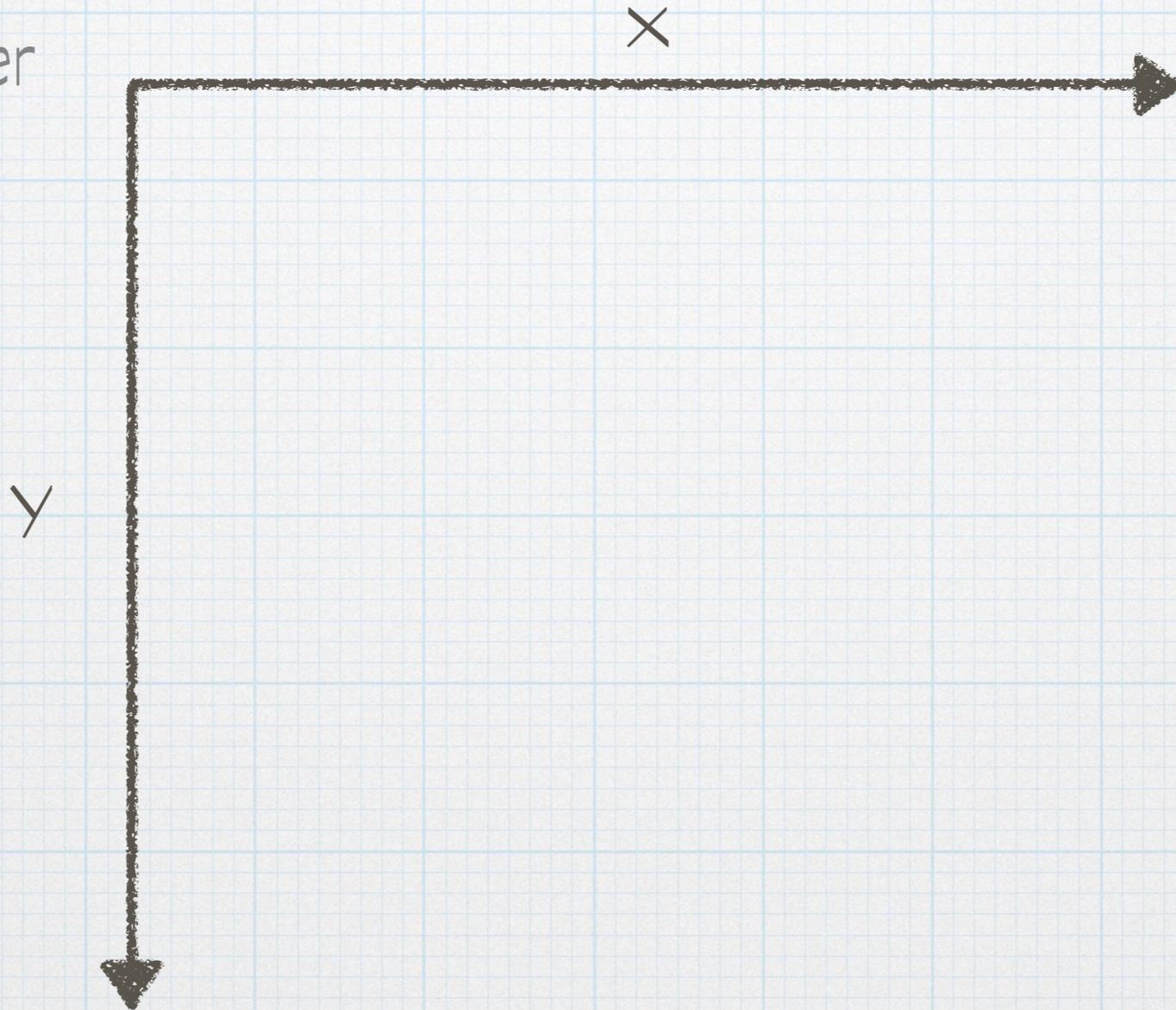
upper-left corner

$(0, 0)$ is in the



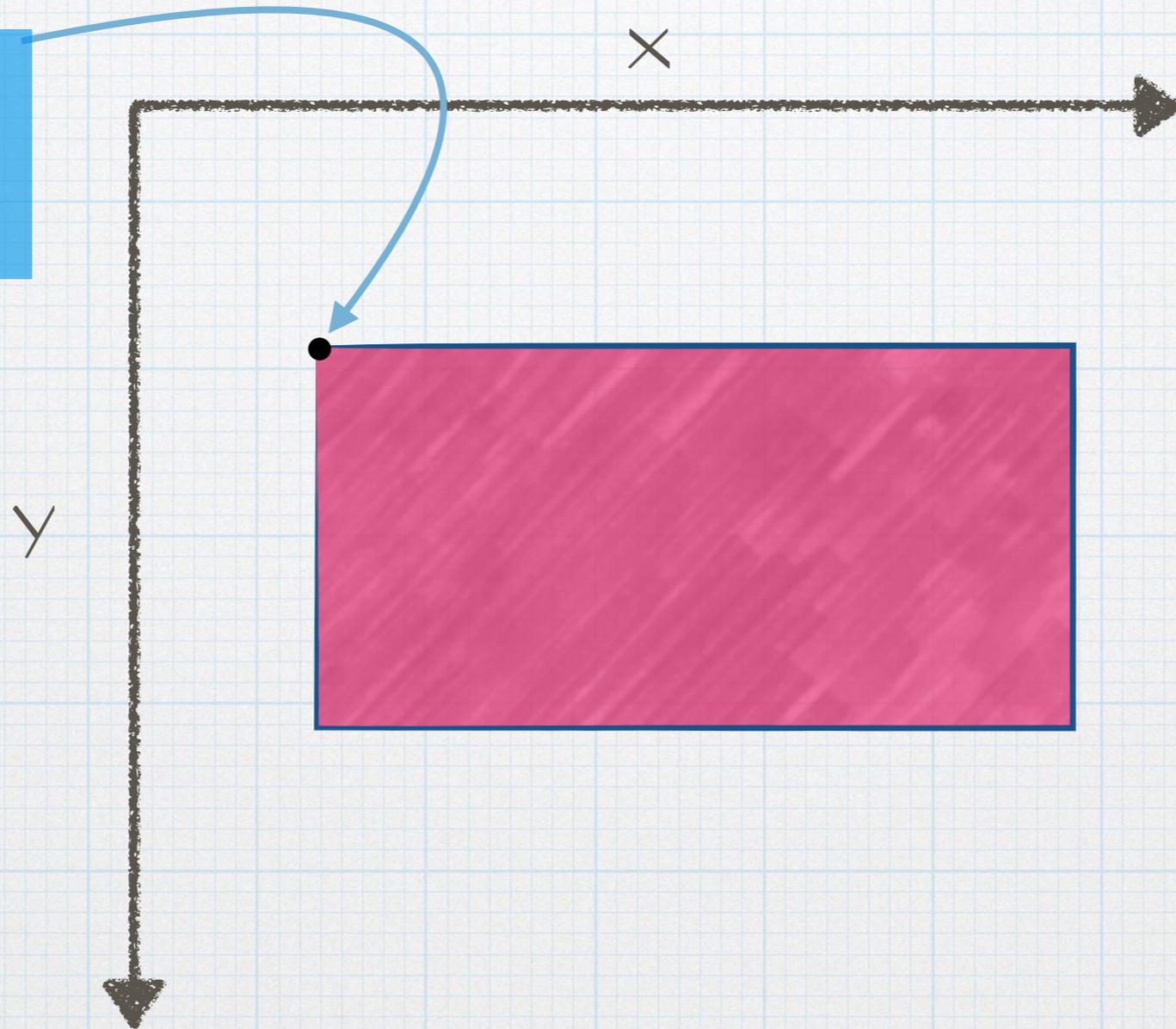
SVG COORDINATE GRID

$(0, 0)$ is in the
upper-left corner



SVG COORDINATE GRID

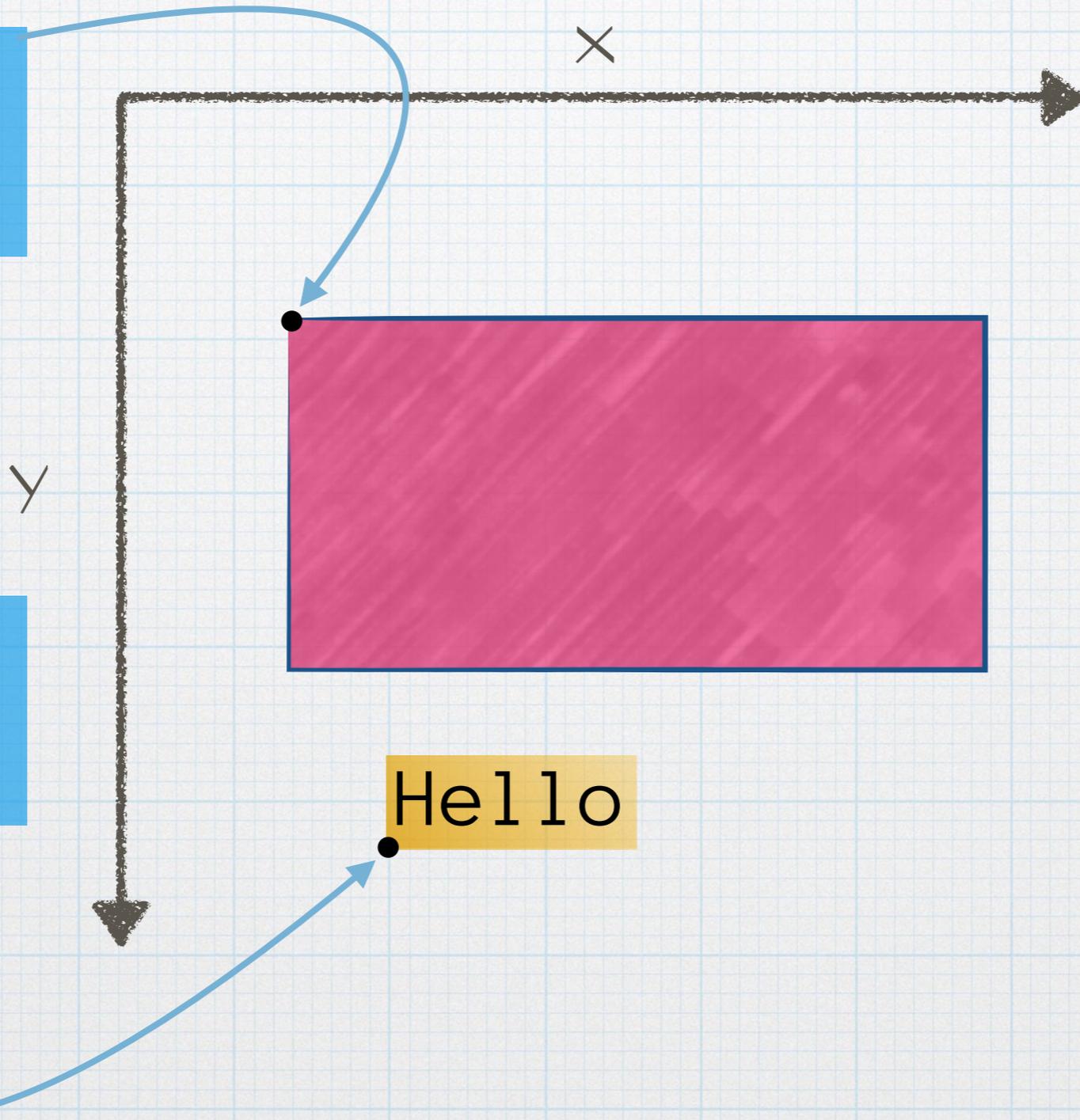
x and y still the
top-left corner!



SVG COORDINATE GRID

x and y still the top-left corner!

x and y are still the bottom-left corner!



SVG EXPERIMENT

Build an HTML document and put the following SVG container in the `<body>` section:

```
<svg width=500 height=500>
```

```
<line x1="0" y1="100" x2="500" y2="100" stroke="grey" stroke-width="1" />
```

```
<line x1="20" y1="0" x2="20" y2="500" stroke="grey" stroke-width="1" />
```

```
<rect x="20" y="100" height="30" width="40" fill="red" />
```

```
<rect x="20" y="200" height="30" width="40" fill="green" />
```

```
<rect x="20" y="300" height="30" width="40" fill="black" />
```

```
<circle cx="300" cy="300" r="50" fill="yellow" stroke="grey"  
stroke-width="1" />
```

```
</svg>
```

SVG EXPERIMENT

Build an HTML document and put the following SVG container in the `<body>` section:

```
<svg width=500 height=500>
```

```
<line x1="0" y1="100" x2="500" y2="100" stroke="grey" stroke-width="1" />
```

```
<line x1="20" y1="0" x2="20" y2="500" stroke="grey" stroke-width="1" />
```

```
<rect x="20" y="100" height="30" width="40" fill="red" />
```

```
<rect x="20" y="200" height="30" width="40" fill="green" />
```

```
<rect x="20" y="300" height="30" width="40" fill="black" />
```

```
<circle cx="300" cy="300" r="50" fill="yellow" stroke="grey"
stroke-width="1" />
```

```
</svg>
```

SVG EXPERIMENT

Build an HTML document and put the following SVG container in the `<body>` section:

```
<svg width=500 height=500>  
  
<line x1="0" y1="100" x2="500" y2="100" stroke="grey" stroke-width="1" />  
<line x1="20" y1="0" x2="20" y2="500" stroke="grey" stroke-width="1" />  
  
<rect x="20" y="100" height="30" width="40" fill="red" />  
<rect x="20" y="200" height="30" width="40" fill="green" />  
<rect x="20" y="300" height="30" width="40" fill="black" />  
  
<circle cx="300" cy="300" r="50" fill="yellow" stroke="grey"  
  stroke-width="1" />  
  
</svg>
```

ADDING A CSS CLASS

Notice that several elements have a grey stroke of width 1 pixel. We could create a CSS class for that.

```
<svg width=500 height=500>  
  
<line x1="0" y1="100" x2="500" y2="100" stroke="grey" stroke-width="1" />  
<line x1="20" y1="0" x2="20" y2="500" stroke="grey" stroke-width="1" />  
  
<rect x="20" y="100" height="30" width="40" fill="red" />  
<rect x="20" y="200" height="30" width="40" fill="green" />  
<rect x="20" y="300" height="30" width="40" fill="black" />  
  
<circle cx="300" cy="300" r="50" fill="yellow" stroke="grey"  
  stroke-width="1" />  
  
</svg>
```

CSS CLASS

Create a CSS document with the following stylization and link it to the HTML document.

```
.grey_stroke {  
    stroke: grey;  
    stroke-width: 1;  
}
```

```
<svg width=500 height=500>
```

```
<line x1="0" y1="100" x2="500" y2="100" class="grey_stroke" />
```

```
<line x1="20" y1="0" x2="20" y2="500" class="grey_stroke" />
```

```
<rect x="20" y="100" height="30" width="40" fill="red" />
```

```
<rect x="20" y="200" height="30" width="40" fill="green" />
```

```
<rect x="20" y="300" height="30" width="40" fill="black" />
```

```
<circle cx="300" cy="300" r="50" fill="yellow" class="grey_stroke" />
```

```
</svg>
```

SVG GROUP

We can group a series of elements together with a `<g>` container so that we can treat them all as one unit. For example:

```
<g>  
  <circle r="20" />  
  <text>This Circle's Label</text>  
</g>
```

That draws the group with the circle's centre at the origin of (0, 0), which is in the top-left corner.

SVG GROUP

We can group a series of elements together with a `<g>` container so that we can treat them all as one unit. For example:

```
<g>
  <circle r="20" />
  <text>This Circle's Label</text>
</g>
```

That draws the group with the circle's centre at the origin of (0, 0), which is in the top-left corner.

We can add a `transform` attribute to move (or `translate`*) that group so that we move the circle's centre 100 units right on the x-axis and 50 units down on the y-axis:

```
<g transform="translate(100, 50)">
  <circle r="20" />
  <text>This Circle's Label</text>
</g>
```

* In Latin, *trans-latus* means "having been carried across"

SVG TRANSFORM

To move the text, we can transform it as well:

```
<g transform="translate(250,250)">  
  <circle r="25" fill="yellow" class="grey-stroke" />  
  <text transform="translate(20,-20)">Hello!</text>  
</g>
```

We can also **scale** things and **rotate** things:

```
<g transform="translate(250,250) scale(2)">  
  <circle r="25" fill="yellow" class="grey-stroke" />  
  <text transform="translate(20,-20) rotate(45)">Hello!</text>  
</g>
```

SVG TRANSFORM

There are a number of transformations that we can apply to SVG elements and groups:

transformation	meaning
<code>translate(x, y)</code>	move object on the x-axis and on the y-axis by these many units
<code>scale(x, [y])</code>	resize by a factor (or, if two numbers are given, scale in both x-direction and in y-direction)
<code>rotate(angle)</code>	spin the object clockwise by this many degrees
<code>skewX(angle)</code>	slant an object along the x-axis by this many degrees
<code>skewY(angle)</code>	slant an object along the y-axis by this many degrees

SVG REFERENCE

SVG Intro Tutorial:

https://www.w3schools.com/graphics/svg_intro.asp

SVG **path** elements:

(We're not covering paths, but this is a good tutorial)

https://www.w3schools.com/graphics/svg_path.asp