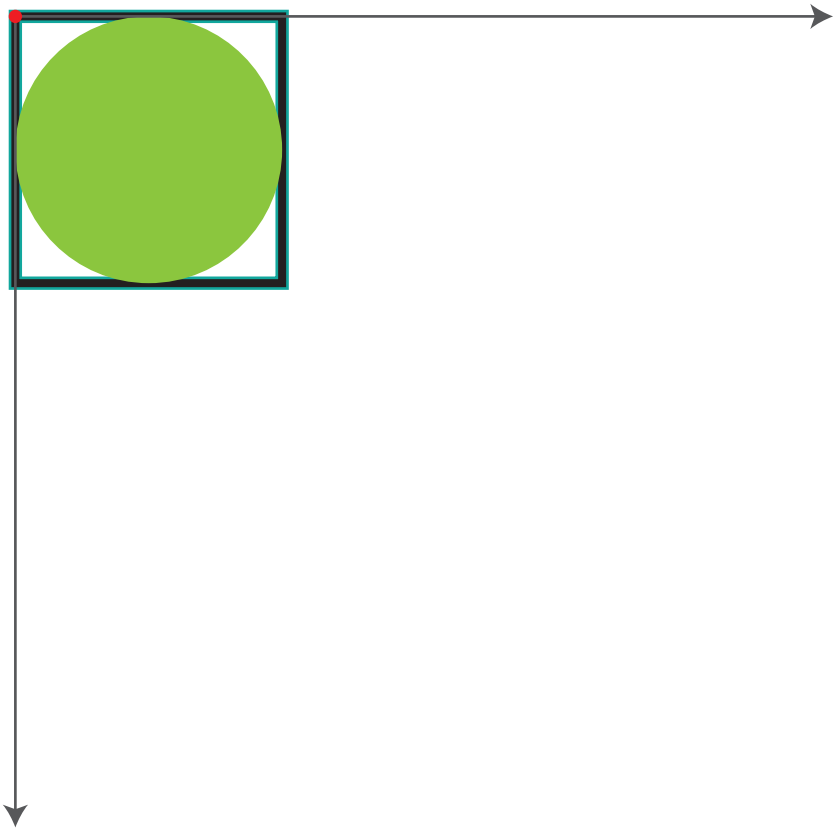
lesson 6

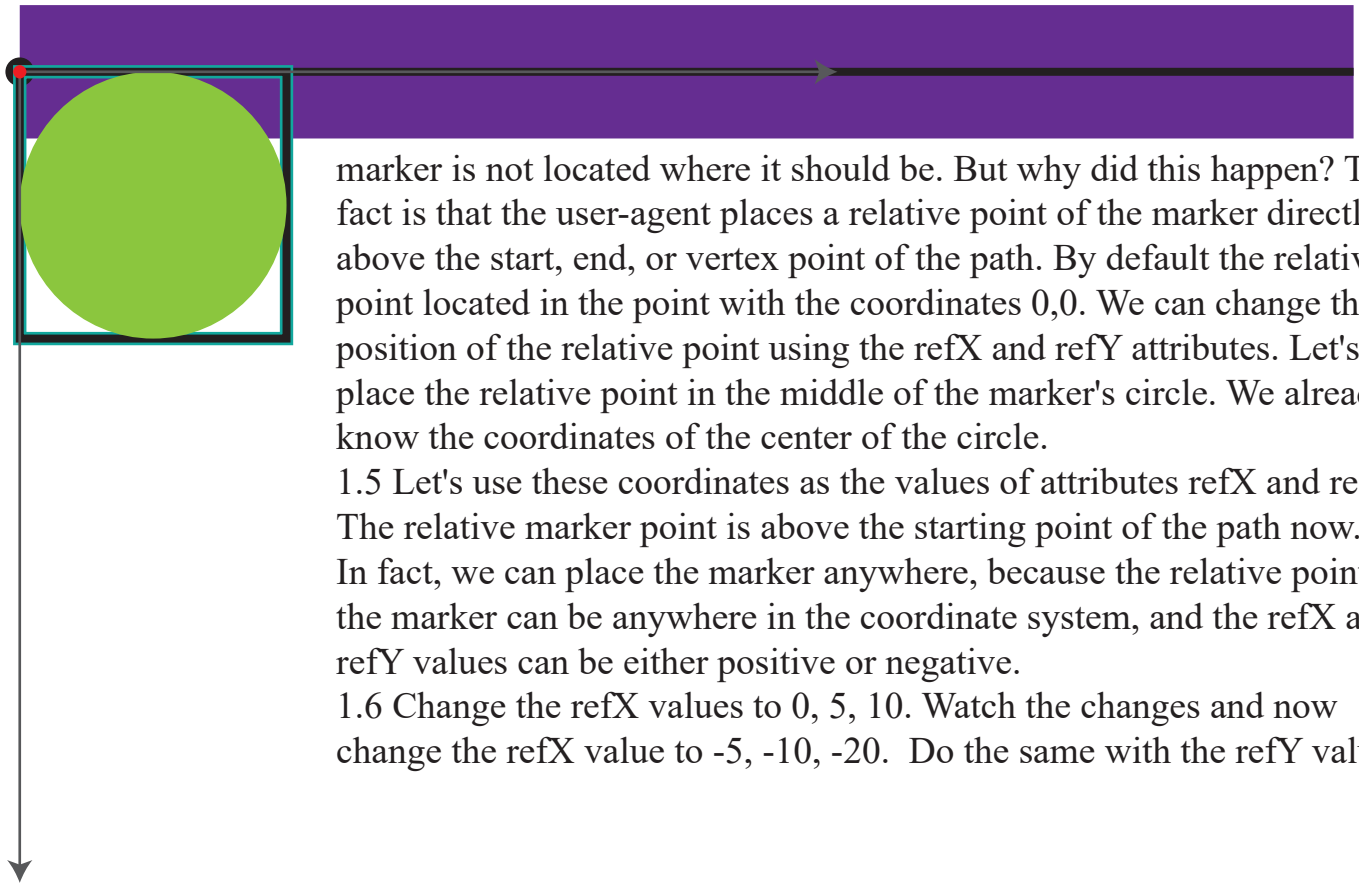
MARKERS



In the first module, we will add the markers to the line element with the st0 class.
1.1 We should add the marker element with an id="circle-marker" below the line element and add the circle element with class="st1" and the following geometry properties cx="5", cy="5", r="5".
As you could notice marker element's behavior is like the symbol
element behavior. We should invoke the marker inside the particular element to render it the same way as we should invoke the symbol element by using the use element.
But in addition, we should specify the viewBox and the viewport of the marker the same way we did for SVG or symbol elements.
As you could notice in the video marker element hasn't height and width properties, but it has markerHeight and markerWidth properties.
My opinion is that this was done in order to avoid using the marker element in an inappropriate way, for example, to invoke it by using the use element.
It is important to note that the marker element as both the svg and symbol elements creates two coordinate systems and three layers.
1.2 Let's add a viewBox to the marker element with a value of "0 0 10 10" and markerWidth and markerHeight both with a value of "3". What does the value of markerWidth and markerHeight mean I will explain in the next module.
1.3 Let's invoke the marker on the line element for this we need to add the marker-start attribute with the URL function as the value to line element(marker-start="url(#circle-marker)").
1.4 You can see that marker appear in the wrong place. As you may have noticed, the

lesson 6

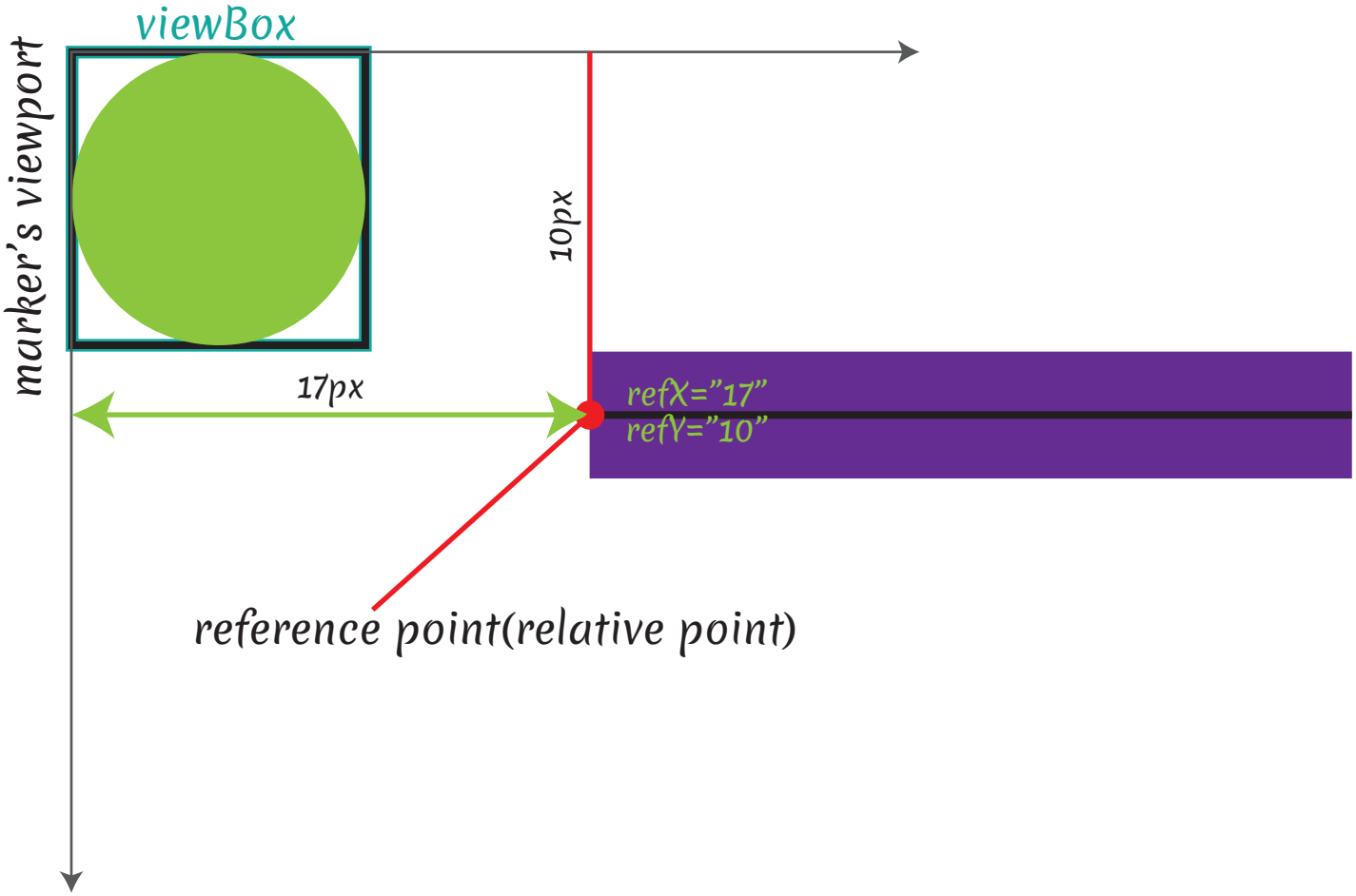
MARKERS



marker is not located where it should be. But why did this happen? The fact is that the user-agent places a relative point of the marker directly above the start, end, or vertex point of the path. By default the relative point located in the point with the coordinates 0,0. We can change the position of the relative point using the refX and refY attributes. Let's place the relative point in the middle of the marker's circle. We already know the coordinates of the center of the circle.

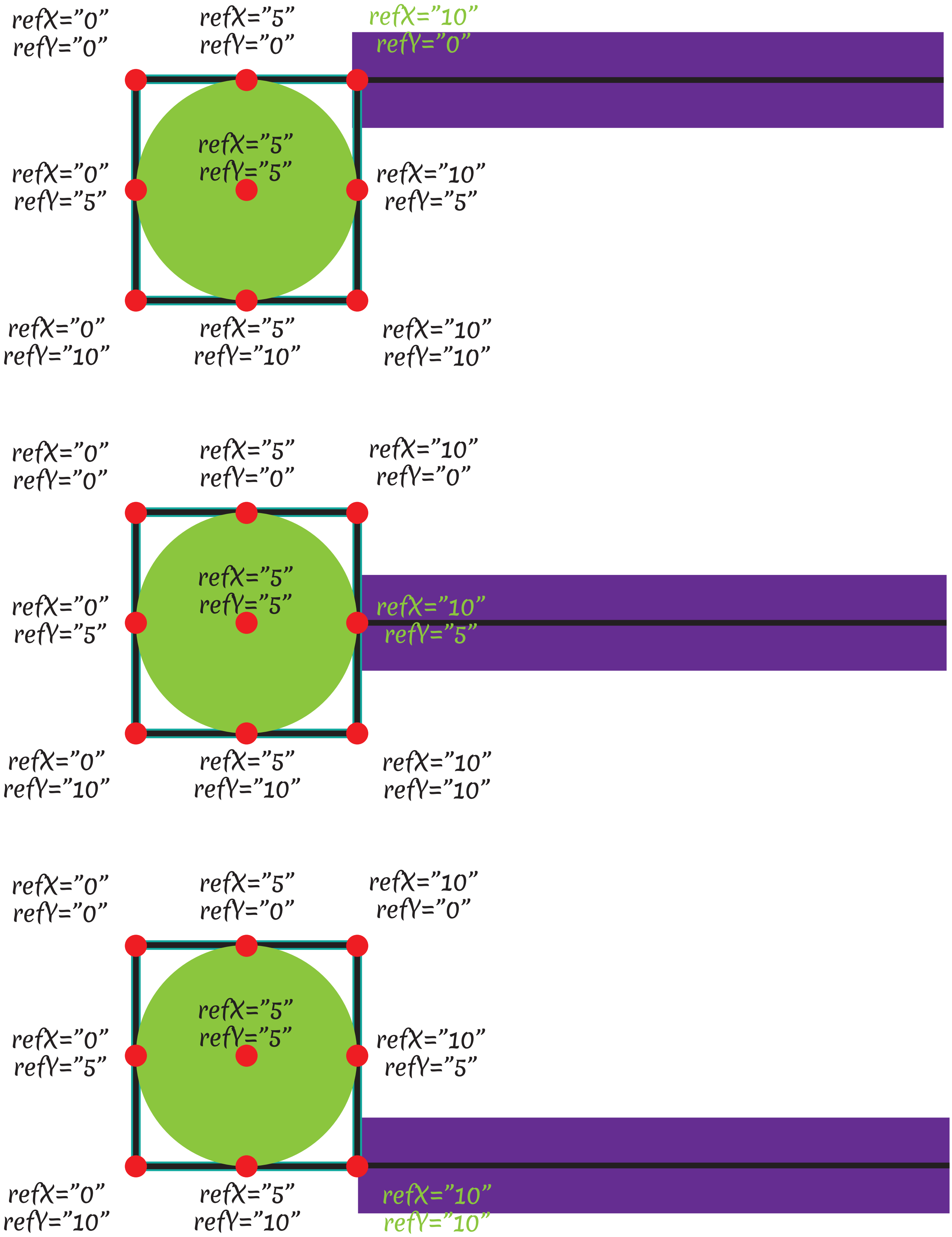
1.5 Let's use these coordinates as the values of attributes refX and refY. The relative marker point is above the starting point of the path now. In fact, we can place the marker anywhere, because the relative point of the marker can be anywhere in the coordinate system, and the refX and refY values can be either positive or negative.

1.6 Change the refX values to 0, 5, 10. Watch the changes and now change the refX value to -5, -10, -20. Do the same with the refY values.



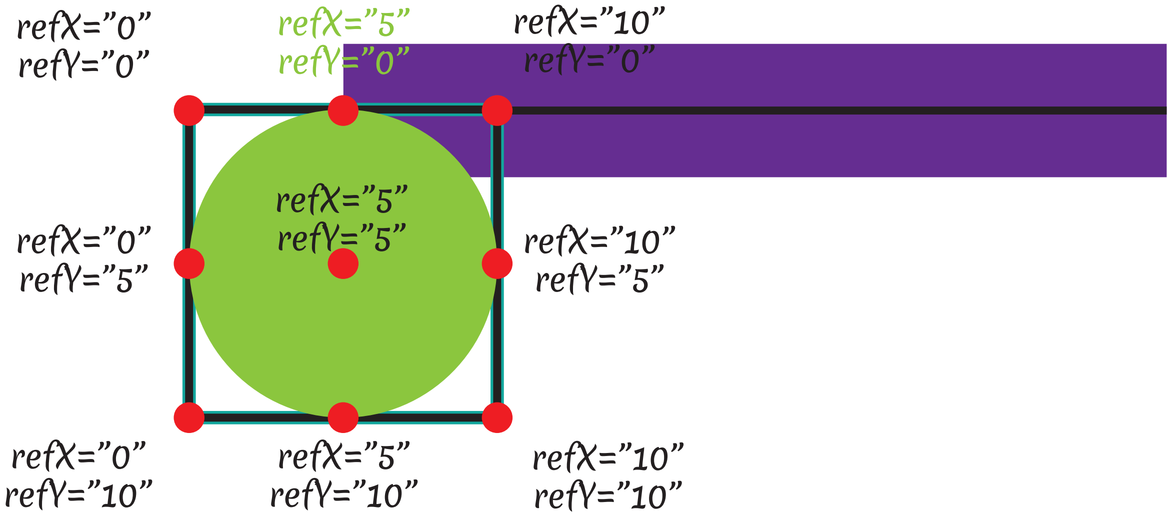
lesson 6

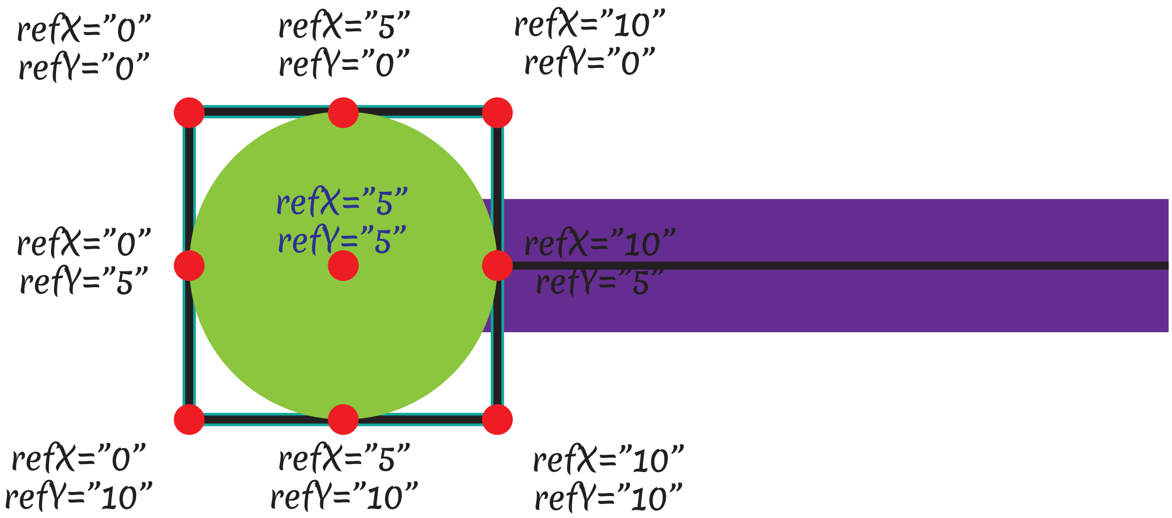
MARKERS

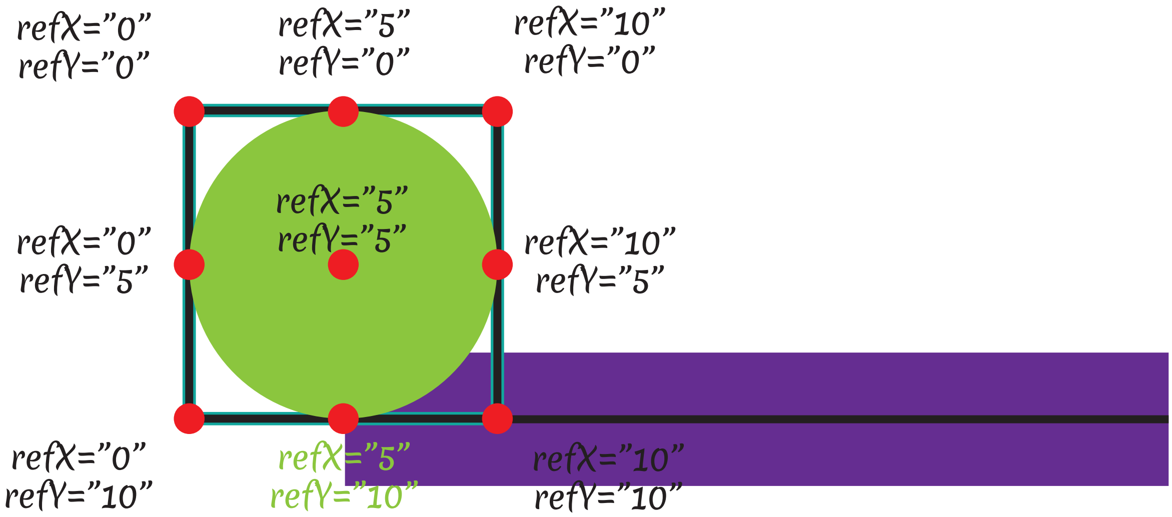


lesson 6

MARKERS

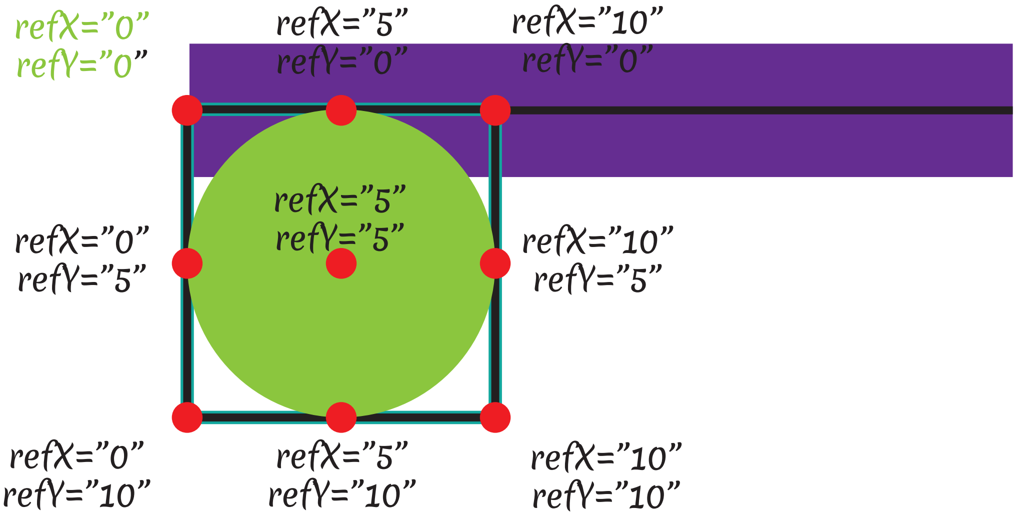


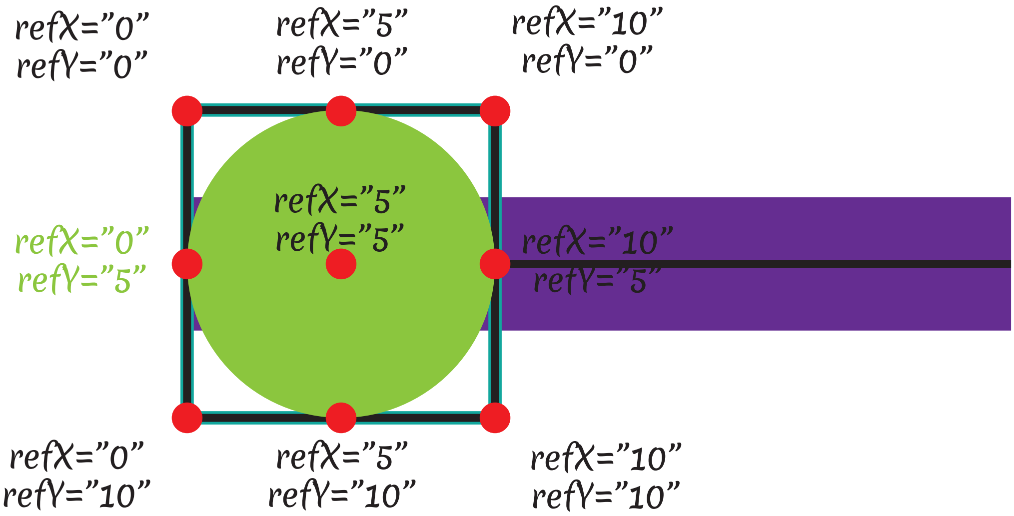


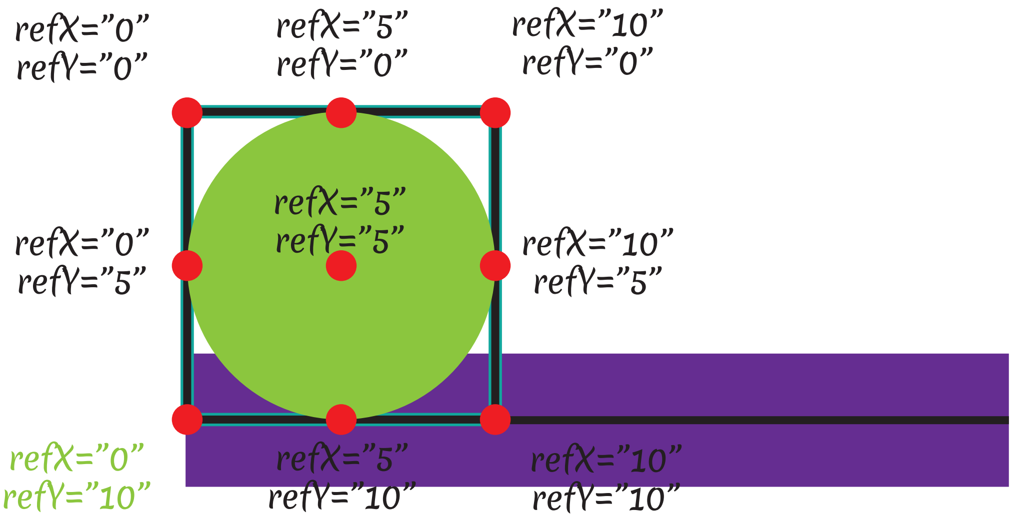


lesson 6

MARKERS





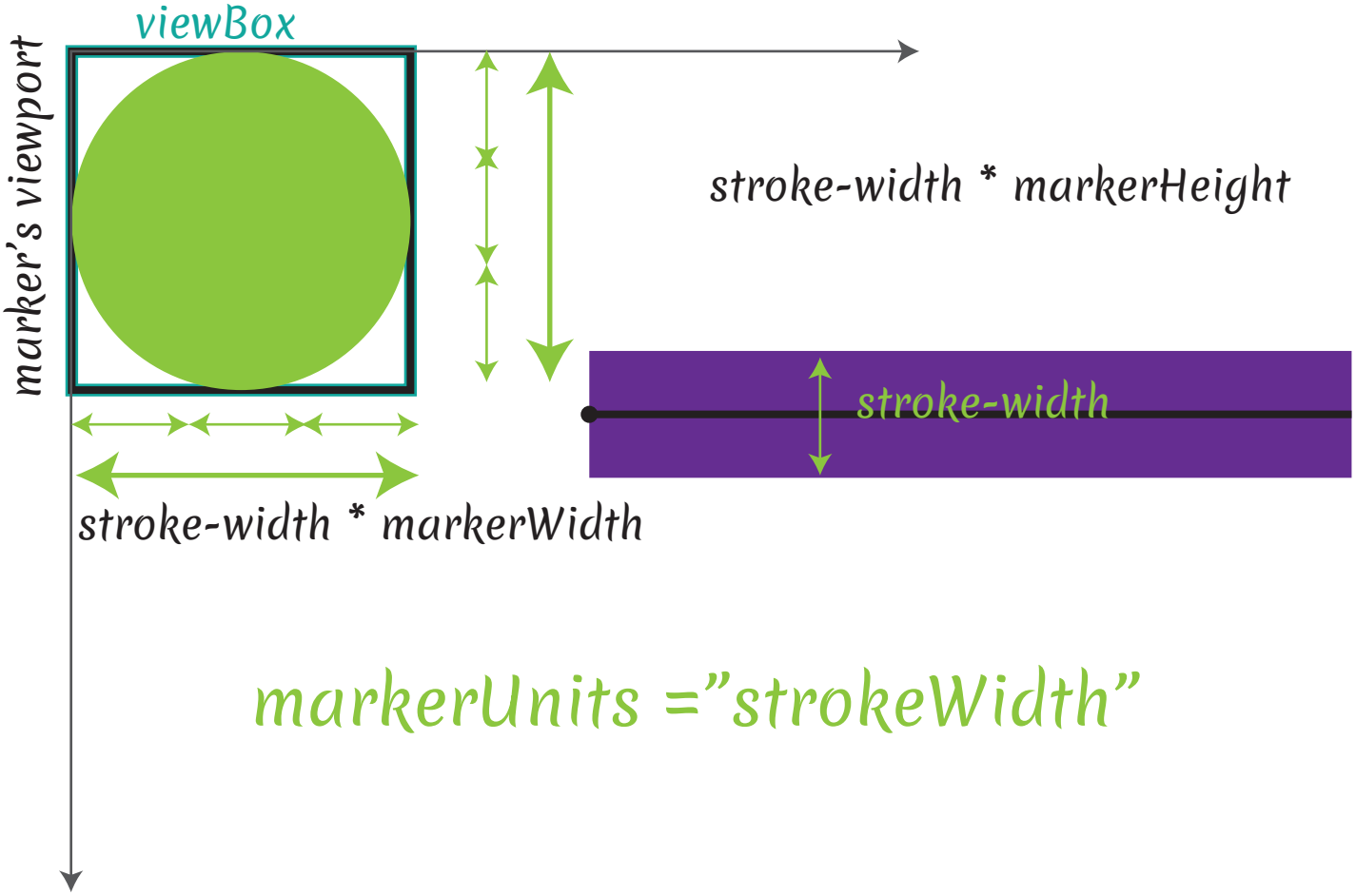


lesson 6

MARKERS

1.7 Now let's add another marker element with the same values of the viewBox, the viewport refX, and refY. The marker should have an id of ”rect-marker”. Inside marker element we should place the rect element with class="st2" and with the following geometry properties x ="0", y="0", height="10" and width="10”.

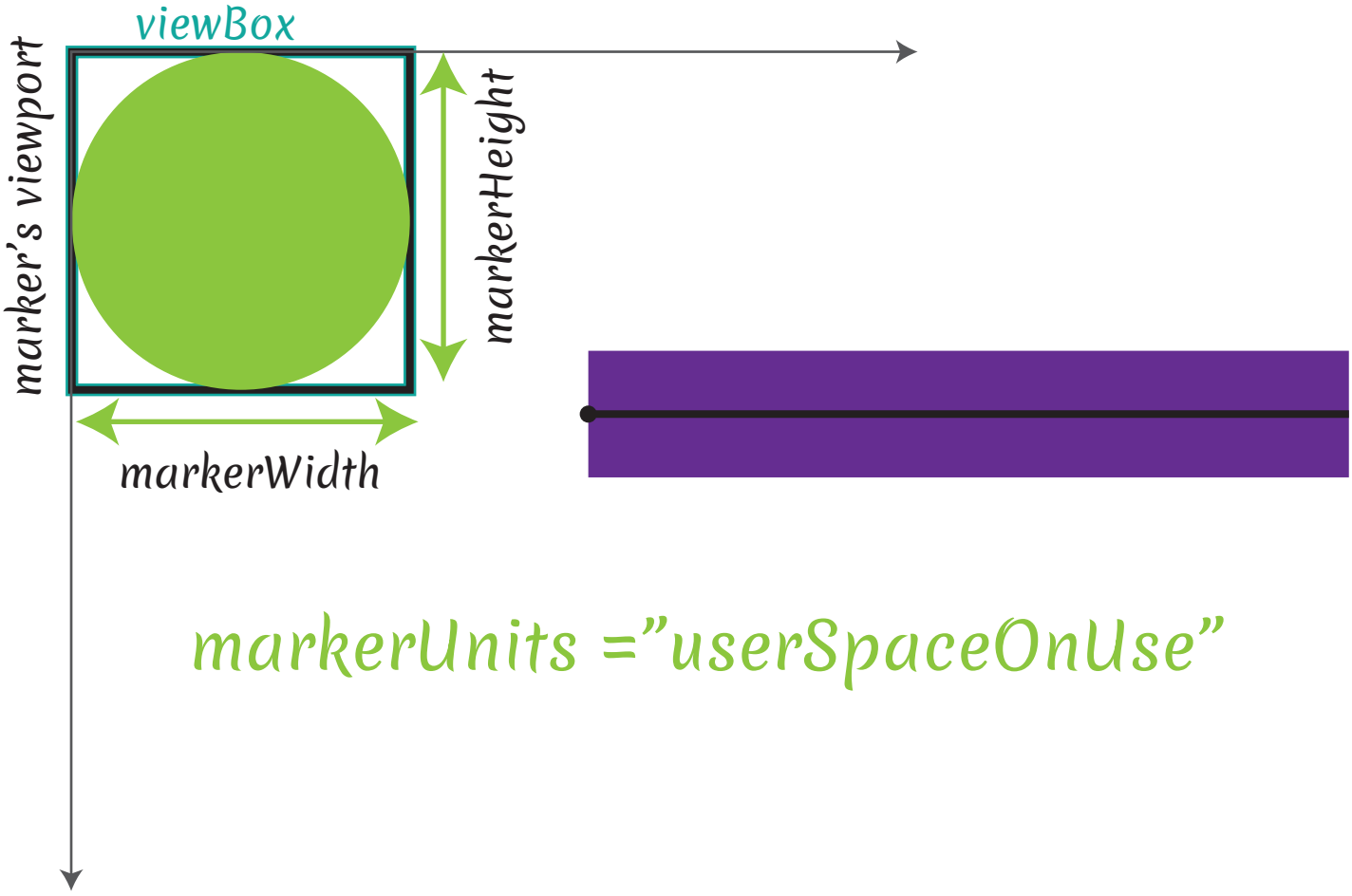
In the second module, we will examine the markerUnits attribute. In the previous module, I had not considered the values of the markerWidth and the markerHeight because they can be interpreted in different ways. What do I mean? Let's run the code of the second module. We can see that the markers are at the start and at the end points of the path.
2.1 Let's add the markerUnits attribute with the value of strokeWidth. Nothing has changed. Now change the value of the markerUnits attribute to userSpaceOnUse. We see that the size of the markers has changed. But why did this happen?
When the value of markerUnits is set to strokeWidth, the height of the viewport marker is equal to the value of markerHeight multiplied by the value of stroke-width, and the width of the viewport marker is equal to the value of markerWidth multiplied by the value of stroke-width.
If the value of markerUnits is set to userSpaceOnUse, the values of markerWidth and markerHeight are interpreted as the user's units.



markerUnits =”strokeWidth”

lesson 6

MARKERS



markerUnits =”userSpaceOnUse”

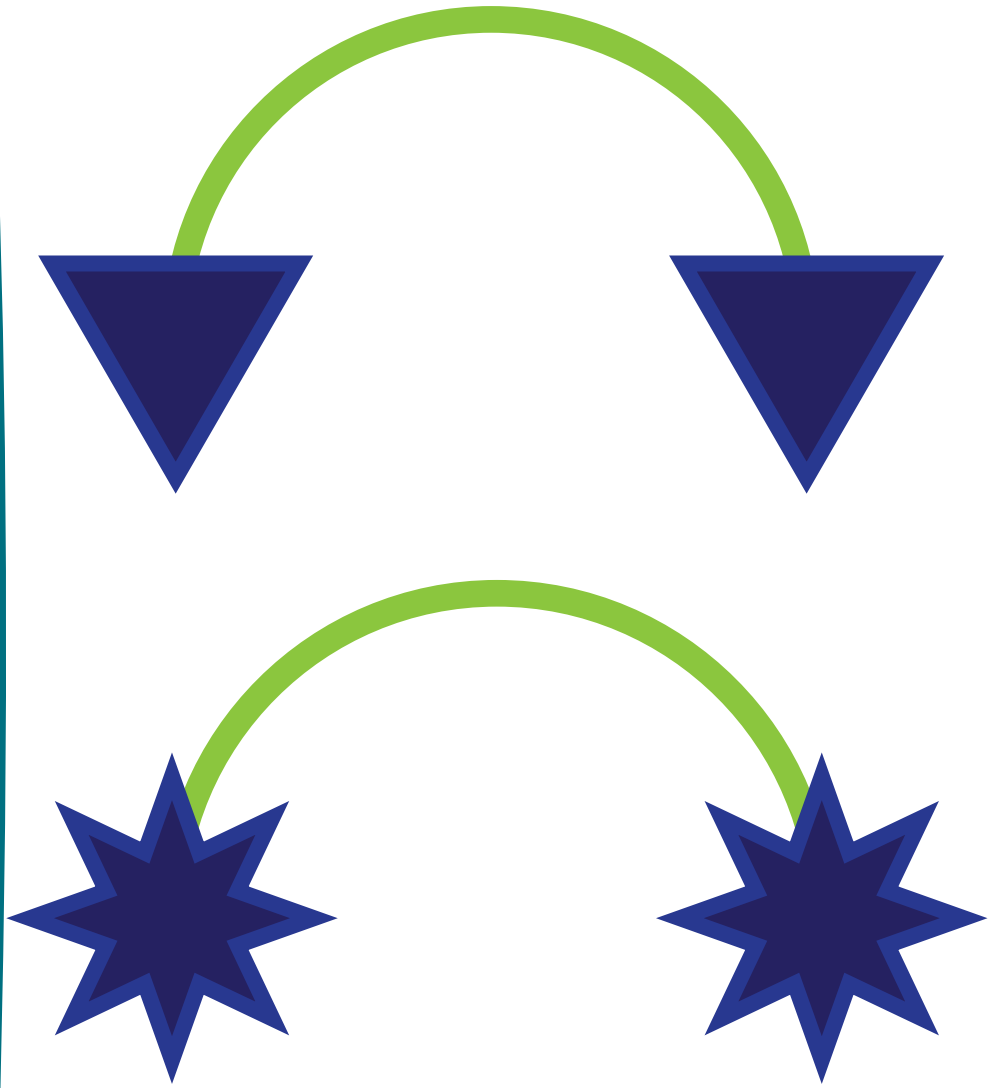
3.1 In the third module, I added two svg elements each containing a path element from the previous module and marker elements. Let's examine an svg element with id=’second’. Inside the svg element, you can see the path element which contains the marker-start and marker-end attributes with url (#triangle-marker) as the values. The marker element with id=”triangle-marker” has the orient attribute which is set to auto. Let's change the value of the orient attribute to "auto-start-reverse". As you can see, the direction of the marker at the beginning of the path has changed, but the direction of the marker at the end of the path has not changed.

The marker has the same direction as it had before. In order to fix this, we need to change the direction of the shape inside the marker manually. Let's change the value of the marker-end attribute to "url(#triangle-marker2)". Now We use a marker element with a url (#triangle-marker2) as the value.

Let's change the orientation of the triangle by adding transform=”rotate(180, 400, 400)” function. We've rotated the triangle on the 180 degrees. Now let's change the orient attribute's value to "auto" again. We can see the result what we want.

lesson 6

MARKERS



As you may have noticed, the orient attribute is responsible for the orientation of the markers on the shape. In the previous part of the task, the value of the orient attribute that was set to auto-start-reverse changed the orientation of the marker that was at the beginning of the path on 180degrees. But the orient attribute can also take numeric values, which will be interpreted as degrees.
Inside the marker element that is inside the svg element with id="first", I have added the animation element. The animation duration is 10s. We will animate the orient property that's why I added the attributeName attribute with the value "orient".
The value of the "from" attribute indicates where the animation starts and the value of the "to" attribute indicates where the animation end. You should change the value of the "from" attribute to 0 and the value of the "to" attribute to 360. This means that the markers will rotate.