

# Learning to Code with SVG

Lesson Plan: Coding a Simple Atom in SVG

Objective: Hands-on learning of SVG by drawing a simple atom without animation of electrons.

Lab Time: Approximately 1 hour, not including Lecture time. Students should copy and paste lines, and then change the attribute values to greatly reduce typing time and typos.

Age range: 3-8th grades, or any age student unfamiliar with SVG

Requirements: A Computer with a browser and a simple text editor like Notepad on Windows, or Text Wrangler (free in the App Store) on a Mac. Familiar with copy and paste shortcuts (ctrl-c and ctrl-v on windows and command-c and command-v on a Mac). Ability to save file with a .svg extension.

Should be familiar with adding `<circle>`, and `<ellipse>` elements in SVG

Resources: <http://steamcoded.org/lessons/numberedgrid400x400.svg>

<http://www.w3schools.com/svg/default.asp>

<https://www.w3.org/TR/SVG11/>

Free eBook for the iPad *Learn SVG Interactively*, by Jay Nick

Lecture: In this lesson `<circle>` and `<ellipse>` elements will be added to the numbered grid. The `<circle>` element has attributes; `cx`, `cy`, and `r` where the center of the circle is defined to be at point `(cx,cy)` and has a radius defined by the `r` attribute. A circle with a center at point `(200,200)` with a radius of 100 would look like `<circle cx="200" cy="200" r="100" />` Similarly, The `<ellipse>` element has attributes; `cx`, `cy`, `rx`, and `ry` where the center of the ellipse is defined to be at point `(cx,cy)`, has a x-radius defined by the `rx` attribute, and has a y-radius defined by the `ry` attribute. An ellipse with a center at point `(200,200)` with an x-radius of 50 and a y-radius or 150 would look like `<ellipse cx="200" cy="200" rx="50" ry="150" />` The lesson will style the circles and ellipses using a web standard called CSS, short for Cascading Style Sheets. CSS styles consist of name:value pairs separated by a semi-colon. Each name is separated from its value with a colon. For example, `style="name:value;name:value;"`

This lesson will add 3 CSS styles, `fill`, `stroke`, and `stroke-width`, where `fill` paints the inside of the element and `stroke` draws the outline of the element. The `stroke-width` sets the thickness of the outline. The `fill` and `stroke` styles take colors as their values or the word `none`. Since the default fill is usually black, this lesson will use the name:value pair `fill:none` regularly. If omitted, expect to see black where it is not desired. The `stroke-width` value is a number followed by its units. This lesson will define the `stroke-width` in pixels, abbreviated `px`.

Procedure: Start the editor application, i.e. Notepad or Text Wrangler. Then have students get an SVG template with 400x400 grid from: <http://steamcoded.org/lessons/numberedgrid400x400.svg.txt>  
Copy the code and paste it into a text editor.

Alternatively, put a copy of the file somewhere that the students have access to and let them copy it from there. Once the code has been copied to the editor, save the file giving the file the name `Mouse.svg` then open the file in a browser. The browser should display the 400 by 400 grid. Keep the text editor and browser windows open.

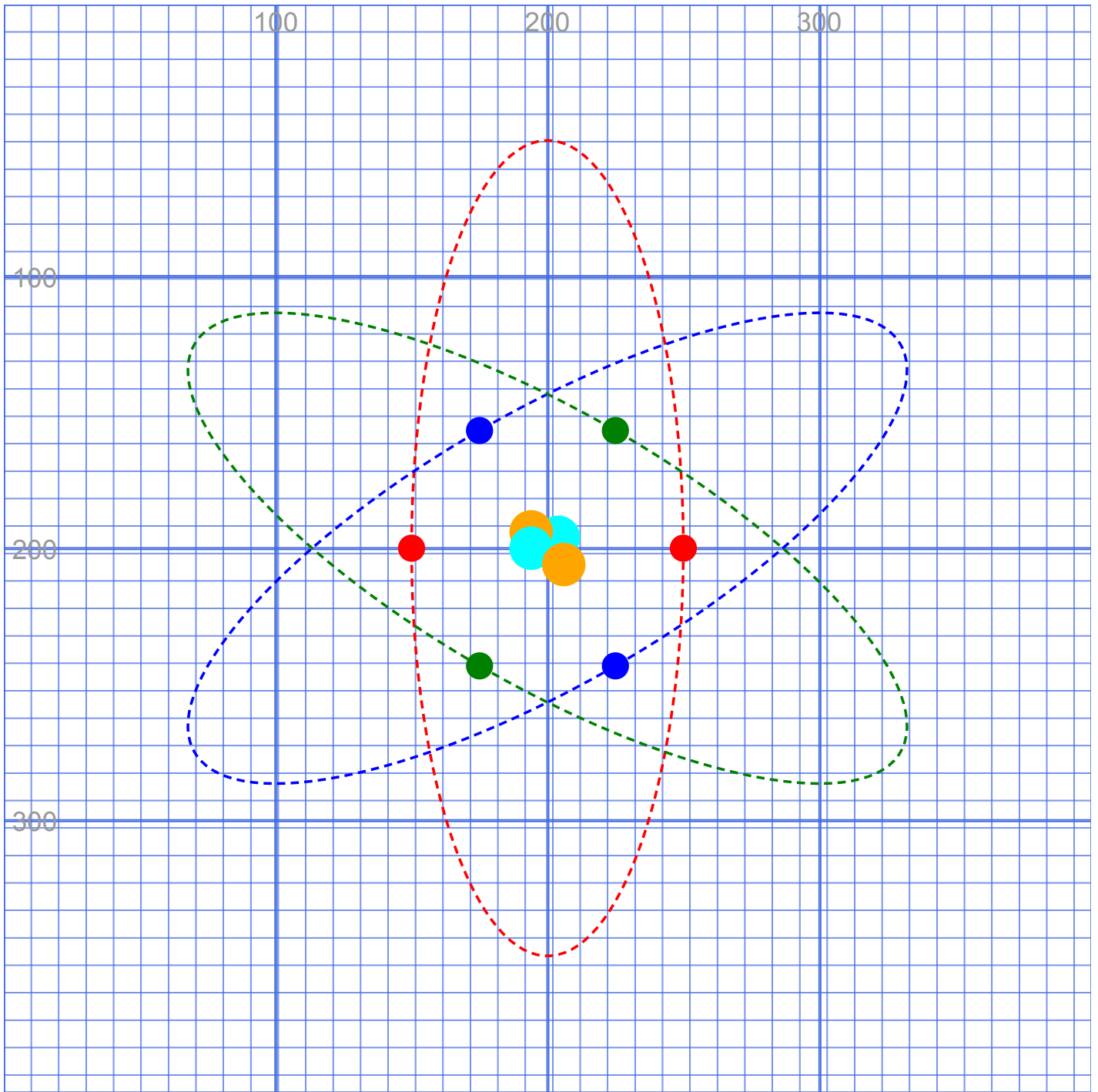
The file contains two grouping `<g>` elements. The first displays a 600 by 600 grid so the students can get a better understanding of the coordinate system of the image and understand why the circles and ellipses are placed where they appear on the screen. The SVG elements the students add should be placed in the second grouping and will be child elements of that `<g>` element. Note that this grouping has a CSS style of `opacity:0.5`. This will allow the background grid to show through the elements contained in that group. When the lesson is completed, the style attribute of these 2 groupings will be changed to hide the grid and not allow the background to show through.

Add SVG elements where indicated using the instructions on page 6 (see below). Students should copy and paste lines, and then change the attribute values to greatly reduce typing time and typos. The copy and paste shortcuts are (ctrl-c and ctrl-v on windows and command-c and command-v on a Mac).

**Important:** Students should save the file and refresh the browser after adding each SVG element to their file.

When complete, change the style attribute of the first `<g>` element from `"display:initial"` to `"display:none"` which hides the grid Then change the style attribute of the second `<g>` element from `"opacity:0.5"` to `"opacity:1"`

- Take Away: Students should understand the minimum requirements of an SVG file, including its coordinate system. Students should feel comfortable adding circles and ellipses to an SVG image and understand CSS styling; fill, stroke, stroke-width, and stroke-dasharray.
- Additional Activity Students can change colors of the electrons, protons, and neutrons. Students can change the position of the electrons. Advanced students can continue on with the animated atom lesson.



# Coding a Simple Atom in SVG on a 400 by 400 grid

To get started, copy the code at this link into your editor: <http://steamcoded.org/lessons/numberedgrid400x400.svg.txt>

Save the file as **atom.svg** and open the file in a browser. Keep both applications open.

In the editor, add the SVG elements (per instructions below) where indicated in the SVG code, i.e. inside the second grouping (<g> element).

**Important:** Save the file and refresh the browser after each step.

## Syntax Examples:

An circle centered at (200,200) with a radius of 100 would have the following code:

```
<circle cx="200" cy="200" r="100" />
```

An ellipse centered at (200,200) with an x-radius of 50 and a y-radius of 150 would have the following code:

```
<ellipse cx="200" cy="200" rx="50" ry="150" />
```

- 1: Draw a ellipse at (200,200) with a x-radius of 50 and y-radius of 150. Add the style attribute as shown.

```
<ellipse cx="200" cy="200" rx="50" ry="150" style="fill:none;stroke:red;" />
```

- 2: Add group element around the <ellipse> element so the code looks like:

```
<g transform="rotate(0,200,200)">
```

```
<ellipse cx="200" cy="200" rx="50" ry="150" style="fill:none;stroke:red;" />
```

```
</g>
```

- 3: Add style attribute stroke-dasharray:3,2; to the ellipse as shown.

```
<ellipse cx="200" cy="200" rx="50" ry="150" style="fill:none;stroke:red;stroke-dasharray:3,2;" />
```

- 4: On the line after the ellipse, draw a circle at (150,200) with a radius of 5 as shown

```
<g transform="rotate(0,200,200)">
```

```
<ellipse cx="200" cy="200" rx="50" ry="150" style="fill:none;stroke:red;stroke-dasharray:3,2;" />
```

```
<circle cx="150" cy="200" r="5" style="fill:red;" />
```

```
</g>
```

- 5: On the line after the circle, draw another circle at (250,200) with a radius of 5 as shown

```
<g transform="rotate(0,200,200)">
```

```
<ellipse cx="200" cy="200" rx="50" ry="150" style="fill:none;stroke:red;stroke-dasharray:3,2;" />
```

```
<circle cx="150" cy="200" r="5" style="fill:red;" />
```

```
<circle cx="250" cy="200" r="5" style="fill:red;" />
```

```
</g>
```

- 6: Copy the 5 lines of code shown in step 5 and paste on next line down.

Then change attribute transform="rotate(0,200,200) to transform="rotate(60,200,200) to rotate the ellipse and circles 60 degrees about point (200,200)

Then change the color red to blue in 3 places.

```
<g transform="rotate(60,200,200)">
```

```
<ellipse cx="200" cy="200" rx="50" ry="150" style="fill:none;stroke:blue;stroke-dasharray:3,2;" />
```

```
<circle cx="150" cy="200" r="5" style="fill:blue;" />
```

```
<circle cx="250" cy="200" r="5" style="fill:blue;" />
```

```
</g>
```

- 7: Copy the 5 lines of code shown in step 5 and paste on next line down after step 6.  
Then change attribute transform="rotate(0,200,200) to transform="rotate(120,200,200) to rotate the ellipse and circles 120 degrees about point (200,200)  
Then change the color red to green in 3 places.

```
<g transform="rotate(120,200,200)">  
  <ellipse cx="200" cy="200" rx="50" ry="150" style="fill:none;stroke:green;stroke-dasharray:3,2;" />  
  <circle cx="150" cy="200" r="5" style="fill:green;" />  
  <circle cx="250" cy="200" r="5" style="fill:green;" />  
</g>
```

- 8: On the next, draw a circle at (204,196) with a radius of 8 and color it cyan.  
**<circle cx="204" cy="196" r="5" style="fill:cyan;" />**
- 9: On the next, draw a circle at (194,194) with a radius of 8 and color it orange.  
**<circle cx="194" cy="194" r="5" style="fill:orange;" />**
- 10: On the next, draw a circle at (194,200) with a radius of 8 and color it cyan.  
**<circle cx="194" cy="200" r="5" style="fill:cyan;" />**
- 11: On the next, draw a circle at (206,206) with a radius of 8 and color it orange.  
**<circle cx="206" cy="206" r="5" style="fill:orange;" />**

When complete, change the style attribute of the first element from "display:initial" to "display:none" which hides the grid. Then change the style style attribute of the second element from "opacity:0.5" to "opacity:1"

# Coding a Simple Atom in SVG on a 400 by 400 grid

## Answer Sheet

Common mistakes are missing double quote marks around attribute values, missing space between attributes, missing the start < and ending /> symbols.

- 1: `<ellipse cx="200" cy="200" rx="50" ry="150" style="fill:none;stroke:red;" />`
- 2: `<g transform="rotate(0,200,200)">  
<ellipse cx="200" cy="200" rx="50" ry="150" style="fill:none;stroke:red;" />  
</g>`
- 3: `<ellipse cx="200" cy="200" rx="50" ry="150" style="fill:none;stroke:red;stroke-dasharray:3,2;" />`
- 4: `<g transform="rotate(0,200,200)">  
<ellipse cx="200" cy="200" rx="50" ry="150" style="fill:none;stroke:red;stroke-dasharray:3,2;" />  
<circle cx="150" cy="200" r="5" style="fill:red;" />  
</g>`
- 5: `<g transform="rotate(0,200,200)">  
<ellipse cx="200" cy="200" rx="50" ry="150" style="fill:none;stroke:red;stroke-dasharray:3,2;" />  
<circle cx="150" cy="200" r="5" style="fill:red;" />  
<circle cx="250" cy="200" r="5" style="fill:red;" />  
</g>`
- 6: `<g transform="rotate(60,200,200)">  
<ellipse cx="200" cy="200" rx="50" ry="150" style="fill:none;stroke:blue;stroke-dasharray:3,2;" />  
<circle cx="150" cy="200" r="5" style="fill:blue;" />  
<circle cx="250" cy="200" r="5" style="fill:blue;" />  
</g>`
- 7: `<g transform="rotate(120,200,200)">  
<ellipse cx="200" cy="200" rx="50" ry="150" style="fill:none;stroke:green;stroke-dasharray:3,2;" />  
<circle cx="150" cy="200" r="5" style="fill:green;" />  
<circle cx="250" cy="200" r="5" style="fill:green;" />  
</g>`
- 8: `<circle cx="204" cy="196" r="5" style="fill:cyan;" />`
- 9: `<circle cx="194" cy="194" r="5" style="fill:orange;" />`
- 10: `<circle cx="194" cy="200" r="5" style="fill:cyan;" />`
- 11: `<circle cx="206" cy="206" r="5" style="fill:orange;" />`