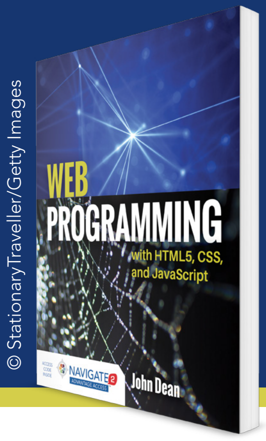


CHAPTER 12
PROJECT

Web Programming with HTML5, CSS, and JavaScript/First Edition
John Dean

Implement a web page that displays a beryllium hydride molecule.

For details, study the screenshot shown below. In the screenshot, the blue bidirectional arrows and their number labels are not part of the web page; they show the pixel dimensions.

Color specifications:

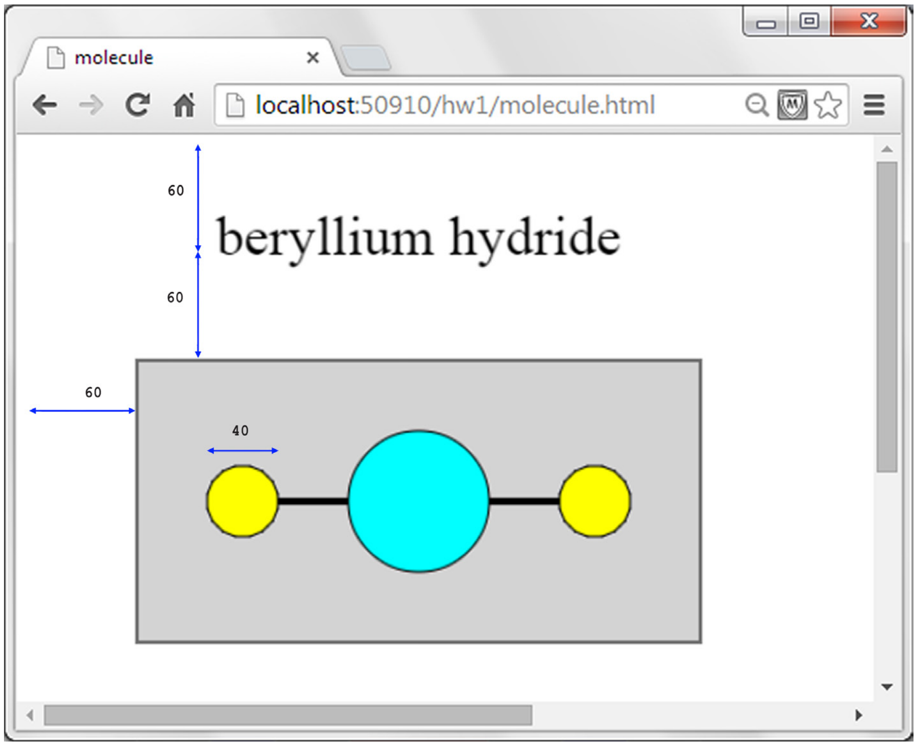
- Use yellow for the two small hydrogen atoms.
- Use blue for the beryllium atom.
- Use light gray for the rectangle's fill color.
- Use black for the rectangle perimeter, the circle perimeters, and the atomic bond lines.

Use proper style and elegance. In particular, you must declare appropriate named constants and you must use them appropriately. Use these named constants:

- Spacer distance = 60 pixels, where the spacer distance is used for each of the following:
  - The distance between the window's left edge and the drawn rectangle
  - The distance between the window's top edge and the molecule name string's baseline
  - The distance between the molecule name string's baseline and the drawn rectangle
- Diameter of each of the two yellow hydrogen atoms = 40 pixels.

In drawing all your graphics items, don't use hardcoded coordinate positions. Instead, use multiples of the above named constants. For example, the diameter of the beryllium atom should be two times the named constant of the hydrogen atoms' diameter.

To accommodate old browsers, include fallback text in your canvas container element.



beryllium hydride

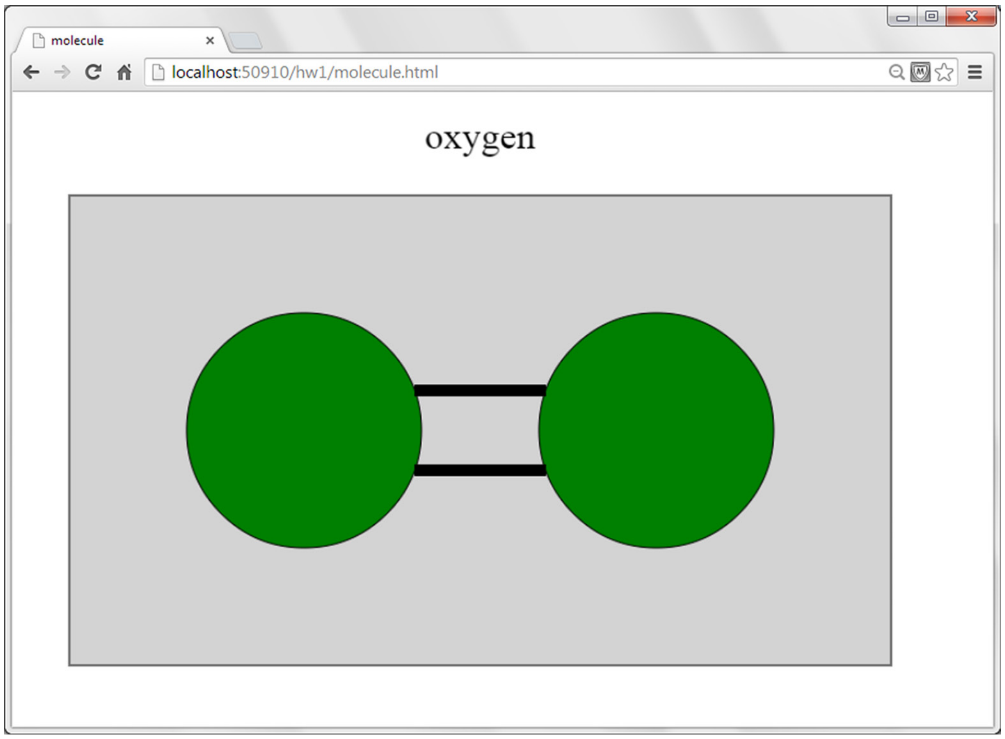
EXTRA CREDIT:

Implement a web page that displays an oxygen molecule. For details, study the screenshot shown below.

Note:

- <BL>Use a named constant of 40 pixels for the spacer distance as described above.
- Use a named constant of 100 pixels for the radius of each oxygen atom.
- Use a named constant of .35 for the angle in radians up to the top covalent bond.
- Use green for the oxygen atoms.</BL>

The key to this problem is having the connecting lines (double covalent bonds) match precisely with the circle perimeters. In the interest of scalability and precision, you're required to calculate the end points using trigonometry. You'll get no credit if you just guess end points by trial and error and hardcode your findings. You'll probably want to use the `Math.cos` a nd/or `Math.sin` methods.


molecule
localhost:50910/hw1/molecule.html
oxygen