**Lesson 8: Positioning HTML Elements**

**How can we reposition elements on a web page?**

| **Overview** | |
| --- | --- |
| In this lesson, students will learn how to reposition HTML elements using the position() function. They will also learn how to add and position images to their web pages outside of the canvas, comparing the image() and createImg() functions. | |
| **Lesson Objectives** | |
| Students will be able to…   * Compare and contrast the image() and createImg() functions * Add images to a web page using createImg() * Position HTML elements on a web page using position() | |
| **Suggested Duration** | |
| One period (45 minutes) | |
| **Blueprint Foundations Student Outcomes (**https://blueprint.cs4all.nyc/outcomes/) | |
| Algorithms  Prototype | **Explain why** I used specific instructions to complete a task. |
| Programming Communicate | **Discuss** what can and cannot be done with a specific set of commands. |
| Algorithms Communicate | **Compare and contrast** my instructions with other instructions that complete the same task. |
| **Vocabulary** | |
| * N/A | |
| **Planning Notes** | |
| For students who want to try the Spicy Challenge:   * It is recommended that they use an online tool like [Photo splitter](https://safeimagekit.com/split-photo) or [ImageSplitter](https://postcron.com/image-splitter/editor/en/upload-image). Make sure these sites are unblocked by your school. * The tools above will package the new image “pieces” in a zip file. If possible, find a video tutorial on how to open up a zip file on the device your students use and make that available as an additional resource. | |
| **Resources** | |
| * Video Tutorial: [Manipulating DOM Elements with html() and position()](https://www.youtube.com/watch?v=YfaJ20vXcK8&ab_channel=TheCodingTrain) * p5 Reference: [createImg()](https://p5js.org/reference/#/p5/createImg) * p5 Reference: [position()](https://p5js.org/reference/#/p5.Element/position) * p5 Reference: [size()](https://p5js.org/reference/#/p5.Element/size) | |
| **Assessments** | |
| * Assess the **Do Now**. Check for the ability to:   + Compare the inputs and outputs of the image() and createImg() functions * Assess the **Student Activity**. Check for the ability to:   + Add an image to a web page using createImg()   + Move an image to a new location using the position() method | |

| **Do Now:** |
| --- |
| * **[Design Journal]** Look over the code in [this sketch](https://editor.p5js.org/mparker/sketches/n92z8k0N-). This sketch uses two images, one of a map and a GIF of a biker. The map was loaded using **loadImage()** and drawn with the **image()** function on Line 19, and the biker GIF was added using the **createImg()** function on Line 14.   + What are some similarities and differences in the parameters of **image()** and **createImg()**? HINT: You may want to look at the [p5 reference page](https://p5js.org/reference/)!   + What else do you notice about how these images are drawn on the web page? In other words, how do these different image functions affect the **output**? |
| **Discussion:** |
| * Have students share their responses from the **Do Now**. Expand on these responses by highlighting the following points:   + **image()** has parameters for the image variable, location (x and y), and size (width and height)–this means you can control where the image is drawn on the canvas and how big it is.   + **createImg()** has a parameter for the image file name (or URL) and for an alternate text description–this means you *can’t* control where the image is placed or how big it is.     - Note: Your students may wonder about the purpose of alt text, since it isn’t visible on the web page. Explain that when blind and vision-impaired people use computers, they may use a screen reader which explains what’s on the screen by reading the HTML and alt text out loud to the user.   + **image()** draws an image directly on the **canvas** (and can’t be used outside the canvas), but **createImg()** adds an image directly to the **web page**.   + Both of these functions are able to draw static images and gifs–the decision to use one or the other depends on whether you want the image to be created **onto or outside of the canvas**. |
| **Teacher Demo: Using position() and size() methods** |
| * A key takeaway from the **Do Now** is that **createImg()** doesn’t let you control the image’s position. Explain to students that they will need to use a new method called **position()** if they want to move the image to a new location on the web page!   + Note: Methods are object-specific functions. In Lesson 3, students used the **html()** method to change the text of a paragraph. * For this demo, you’ll build on the sketch from the Do Now. Here is the [completed example](https://editor.p5js.org/mparker/sketches/n92z8k0N-) for your reference. * First, use the **position()** method with **no arguments**. This will return the current position of the image without moving it:      * Point out that the coordinate system of the web page as a whole is the same as for the canvas, with (0, 0) is in the top-left corner. * Ask**:** If the image is currently at (0, 515) and we want to move it to the top of the web page, what new coordinates would make sense? Answer: *(0, 0) to put it in the top left. Any y-value close to zero will place the image near the top of the web page.* * Now, use the **position()** method with new x and y values as **arguments**:      * When you hit play, the image will move to the top of the web page, but it will also cover up the poster! Use a different method, **size()**, to make the gif smaller:      * Just like **position()**, you can call this method without arguments to get the current size of the image. Then, explain to students that you can divide (or multiply) the width and height by the **same amount** to keep the sides of the image **proportional**.   + Note: The original dimensions of the gif in the example are 492 x 498, so you can round the width and height to 500 pixels to make the math easier. * Finally, move the canvas using the same **position()** method. In this example, the canvas is being moved below the gif so they no longer overlap: |
| **Student Activity: position() Puzzle** |
| * Have students choose one of the following options for the **Student Activity**:   + Mild Challenge: Missing Piece! ([starter code](https://editor.p5js.org/mparker/sketches/jbSsr0M1N))   + Medium Challenge: All Mixed Up! ([starter code](https://editor.p5js.org/mparker/sketches/Xu65RE-MI), [complete example](https://editor.p5js.org/mparker/sketches/VDJ6a5q7z))   + Spicy: Remix the mild or medium code challenge using an image you found yourself. After you download it, you should use an online tool like [this one](https://safeimagekit.com/split-photo) or [this one](https://postcron.com/image-splitter/editor/en/upload-image) which will split the image into different “pieces” |
| **Wrap Up** |
| * Make sure your students save their sketches and share them before they leave. |
| **Extensions:** |
| * N/A |