Slicing an Image

WE'LL COVER THE FOLLOWING

The last image-related manipulation we will look at involves taking an image, trimming the parts of it you don't care much about, and taking this smaller image to then display in our `canvas`. This manipulation is more commonly (and concisely!) known as **slicing**. The way it works is by using yet another variant of the `drawImage` method:

```
// this is another drawImage variant!
context.drawImage(image, x, y, w, h, x2, y2, w2, h2)
```

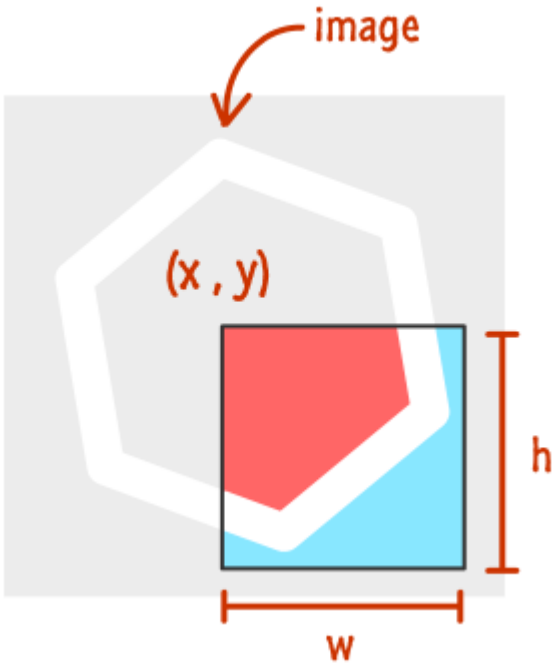
This variant takes nine (YES, NINE!) arguments, and they don't make any sense if you see them for the first time. Parsing the arguments, the x and y stand for position values. The w and h values stand for the width and height. Besides that, it's all still pretty nonsensical. We are going to make sense of all this by looking at all these arguments across two parts.

First Part: Cutting the Original Image

Let's say this is what our original image looks like:

In this image, we would like to only keep the following highlighted region and discard everything else:

What I've just written is a roundabout way of stating that I basically want to cut out a chunk of our original image. The first five arguments to the `drawImage` method allow you to specify the location and size of the region you want to cut:



Let's pair this annotated image up with our `drawImage` definition from earlier:

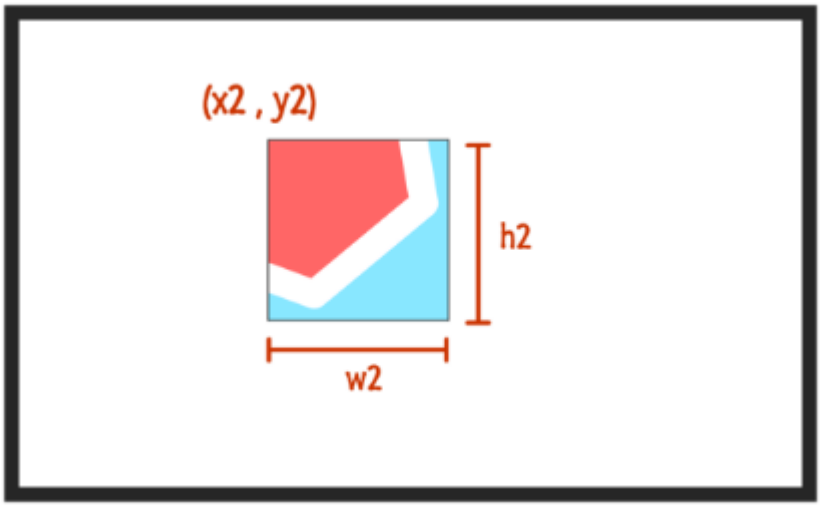
```
context.drawImage(image, x, y, w, h, x2, y2, w2, h2)
```

The **image** argument points to our original image. The **x** and **y** arguments refer to the top-left position of the portion of the image we want to keep. The **w** and **h** arguments refer to the width and height of the portion of the image we want to keep. You put all of this together, we just figured out what more than half of the arguments to this variant of `drawImage` do!

Second Part: Pasting the Cut Image

What we are left with right now is just the part of the original image we decided to keep:

The remaining four arguments to our `drawImage` function help you to place and scale this image into the appropriate location on our `canvas`:



For reference, let's bring our `drawImage` method and its arguments back:

```
context.drawImage(image, x, y, w, h, x2, y2, w2, h2)
```

What we've done is specified the last four arguments, and these arguments are identical to what you saw earlier when learning how to scale your image. The **x2** and **y2** arguments specify the location you want the image to appear on the `canvas`. The **w2** and **h2** arguments allow you to specify the width and height of your image. Keep these values the same as your **w** and **h** arguments if you do not wish to scale your image when "pasting" it into your `canvas`.