Stopping Your Animation Loop

Once your `requestAnimationFrame` loop starts running, rarely will you ever need to tell it to stop. If you do need to stop your animation loop from doing unnecessary work, you can do something like the following:

```
var running = true;

function animate() {
    if (running) {
        // do animation or drawing stuff

    }
    requestAnimationFrame(animate);
}
```

If your `running` variable were to ever be set to **false**, your animation loop will stop doing whatever work is being done. Your `animate` function will still get called by virtue of it being attached to your `requestAnimationFrame`. It just won't be doing any work since the `if (running)` check will return false.

Now, if you really REALLY need to stop your `requestAnimationFrame` from calling some poor function around 60 times a second, you do have `requestAnimationFrame`'s evil twin, `cancelAnimationFrame`. The best way to explain how it prevents `requestAnimationFrame` from working is by looking at a simple example:

```
// store your requestAnimatFrame request ID value
var requestId;

// setting up a click event listener
var bodyElement = document.querySelector("body");
bodyElement.addEventListener("click", stopAnimation, false);

function animate() {

    // doing some animation stuff

    // get the requestID as part of calling animate()
    requestId = requestAnimationFrame(animate);
```

```
}
animate();

function stopAnimation(e) {
    // use the requestID to cancel the requestAnimationFrame call
    cancelAnimationFrame(requestId);
}
```

This simple example should readily highlight how the `cancelAnimationFrame works`. The thing that I didn't call out about `requestAnimationFrame` is that it returns an ID value whenever it gets called:

```
requestId = requestAnimationFrame(animate);
```

Normally, you don't care about this ID. The only time you really need to know this ID value is when wanting to use `cancelAnimationFrame`. The reason is that `cancelAnimationFrame` uses the ID value to identify the right `requestAnimationFrame` function to stop:

```
cancelRequestAnimationFrame(requestId);
```

That's all there is to the `cancelAnimationFrame` function. I should emphasize that you really don't need to go through all this trouble to cancel a `requestAnimationFrame` setup. The initial approach I outlined with the `running` variable is a good enough approach to use. With that said...if you really want to go the extra mile, you have a friend in `cancelAnimationFrame`!