Simple Example

Towards the top of this page, you saw an example of an animation with circles buzzing around a fixed point. You can also view it in a new window if you want to see it in isolation.

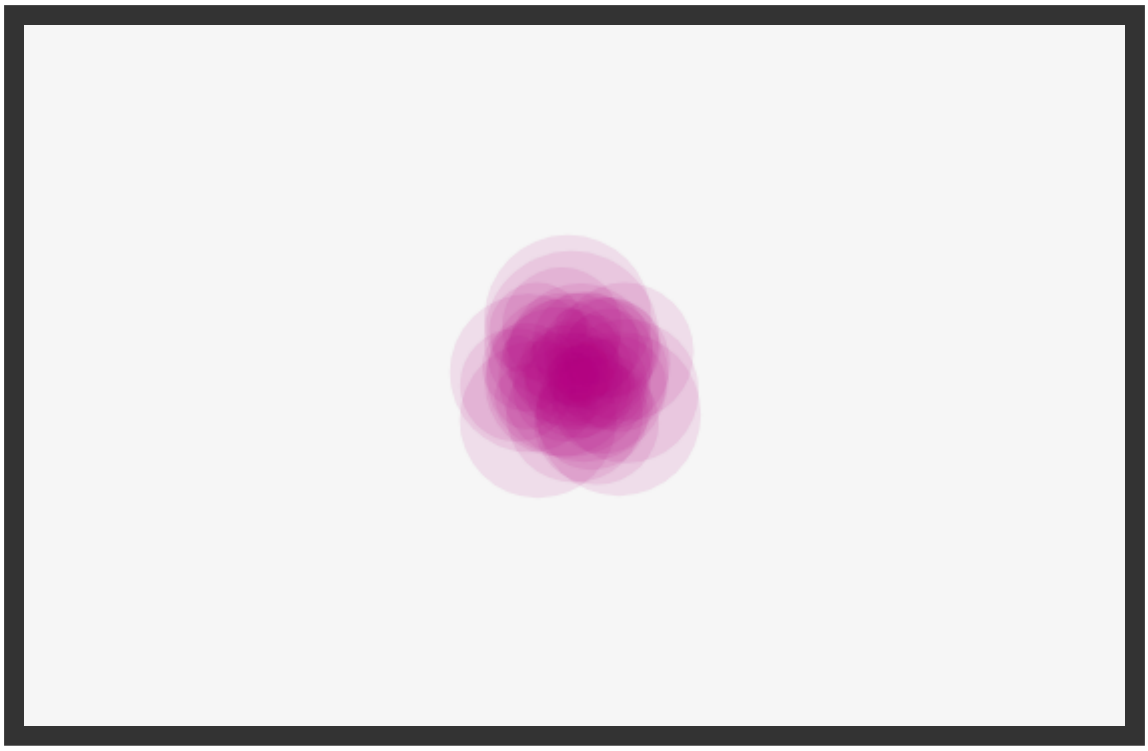
The script for making that animation work looks as follows:

HTML JavaScript

javascript

```javascript
var mainCanvas = document.getElementById("myCanvas");
var mainContext = mainCanvas.getContext('2d');

var canvasWidth = mainCanvas.width;
var canvasHeight = mainCanvas.height;

// this array contains a reference to every circle that you will create
var circles = new Array();

//
// The Circle "constructor" is responsible for creating the circle objects and defining
// they have
//
function Circle(angle, sign, radius, rotationRadius, initialX, initialY) {
    this.angle = angle;
    this.sign = sign;
    this.radius = radius;
    this.rotationRadius = rotationRadius;
    this.initialX = initialX;
    this.initialY = initialY;
    this.incrementer = .01 + Math.random() * .1;
}

Circle.prototype.update = function () {

    this.angle += this.incrementer;

    this.currentX = this.initialX + this.rotationRadius * Math.cos(this.angle);
    this.currentY = this.initialY + this.rotationRadius * Math.sin(this.angle);

    if (this.angle >= (Math.PI * 2)) {
```

output



This all seems like a lot of code, but don't worry about all of that for now. Instead, just pay attention to the highlighted lines. In the two highlighted lines, we call `requestAnimationFrame` with the `draw` function specified as the callback.

Below is that code with everything extraneous omitted:

```
function createCircles() {
       .
       .
       .
    requestAnimationFrame(draw);
}
createCircles();

function draw() {
      .
      .
      .
    requestAnimationFrame(draw);
}
```

Notice that you are essentially creating a loop where you are calling the `draw` function using the `requestAnimationFrame`, and inside the `draw` function, you have another `requestAnimationFrame` call with the same `draw` function.

have another `requestAnimationFrame` call with the same `draw` function specified. You don't have to use `requestAnimationFrame` to make the first call to the draw function. You could have just called draw directly just like I called the animate method a few sections earlier. There is no right or wrong way to do this, but by using `requestAnimationFrame` even for the initial call, you are deferring when to call the `draw` method to your browser as opposed to forcing it to be called instantaneously.

Inside the `draw` method, while this looks like it might cause an infinite loop, your browser knows what to do and ensures that nothing bad like that happens. Don't try something like this with your own vanilla functions, though :P