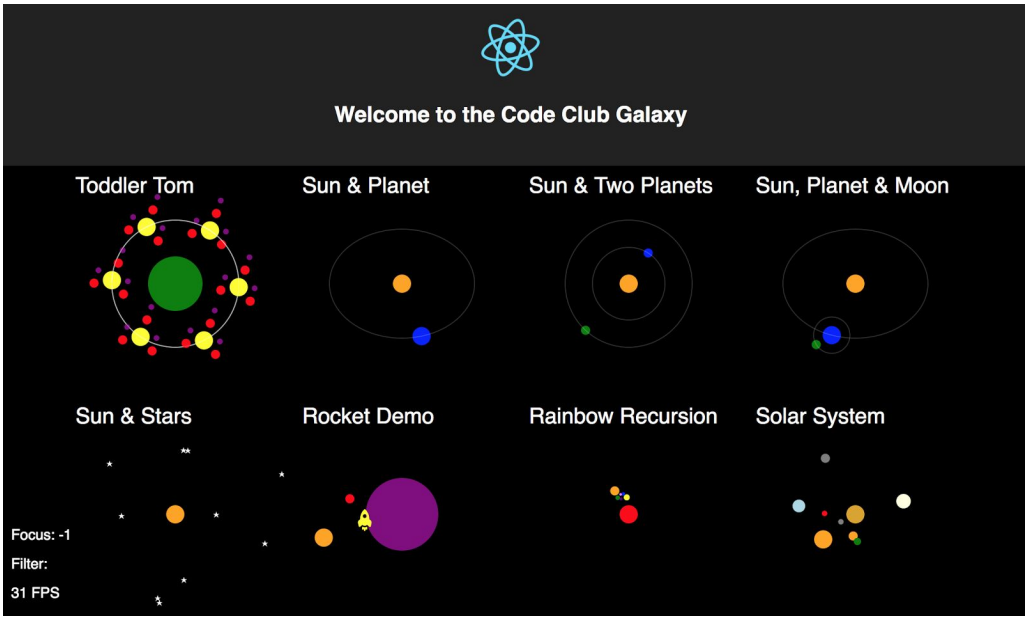
Code Club Galaxy

Today you will be building your own galaxy using coding.

Your galaxy will be put on a shared wall with those made by other people.

Hopefully by the end of the day we will have a whole universe that looks something like this.



Here is an example of the code that you will need to write (or you can edit one of the examples)

```
export default (galaxy) => {
   // Customise the name of your galaxy
   galaxy.setName('Sun & Planet')

   // Create a sun, assign it to a variable
   const sun = galaxy.newSun({
       colour: 'orange'
   })

   // Create a planet that orbits the sun, passing in some customisation
   galaxy.newPlanet({
       colour: 'blue'
   }).orbit(sun)

}
```

Getting Started

Your code club teacher will have started a file for you to work with, it should look like this:

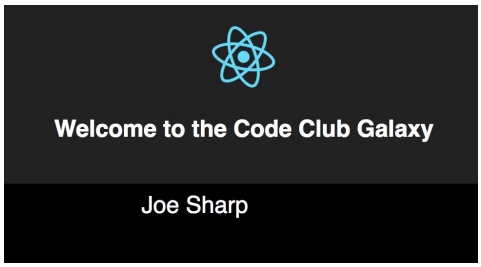
```
export default (galaxy) => {

}
```

First thing you should do is set the name of your galaxy, use the setName function on the galaxy as shown here:

```
export default (galaxy) => {
   galaxy.setName('Joe Sharp')
}
```

Your code club teacher should have set your browser up to visit http://localhost:3000/ and it should auto refresh and look like this.

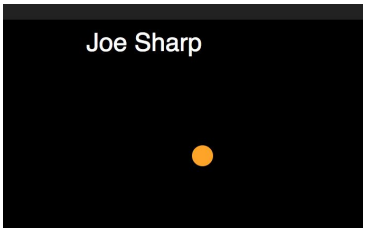


Create a Sun

A common pattern is to build a solar system around a central sun, create the sun by adding the following code to your function:

```
// Create a sun, assign it to a variable
const sun = galaxy.newSun({
    colour: 'orange'
})
```

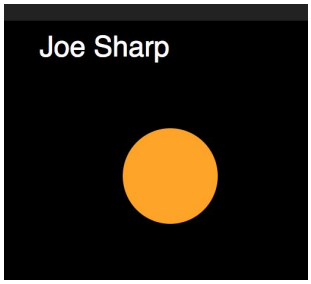
Reload the page and you should see an orange sun roughly in the centre of your galaxy.



You can customise your sun by adding properties to the creation code we wrote earlier. Try adding radius alongside the colour. Note that I have added a comma after ‘orange’ to keep the properties separate:

```
// Create a sun, assign it to a variable
const sun = galaxy.newSun({
    colour: 'orange', // add this comma
    radius: 40
})
```

It should then look like this, seems more like a sun now!



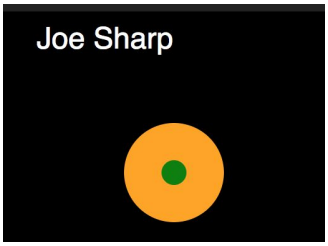
If you want to change the colour, there is a list of all the valid colour names here https://www.w3schools.com/cssref/css_colors.asp

Put a Planet into Orbit

Now it's time to create a planet, and put it in orbit around the sun. Creating a planet is basically the same as creating a sun. Here is some code to add a green planet to your galaxy:

```
const planet = galaxy.newPlanet({
    colour: 'green',
    radius: 10
})
```

That will put the planet in the same place as the sun, and it will just sit there.

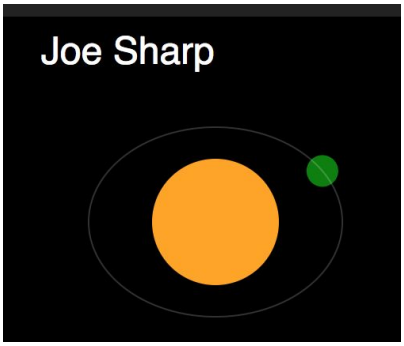


So let's put it into orbit. Add the following code after the creation of the planet:

```
planet.orbit(sun)
```

This function call puts the **planet**, in orbit around the **sun**. So we pass the **sun** in as an argument to the function.

Reload the page and it should now look like this, with the green planet rotating around the orange one:



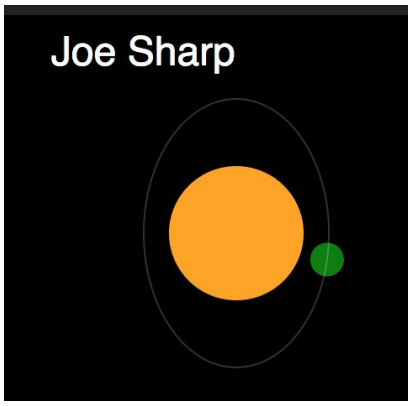
Customise the Orbit

The orbit can be customised in the same way as the planets themselves, by passing in a JavaScript object when we call orbit, we can change the way the orbit works.

You can change your call to .orbit() to look like this:

```
planet.orbit(sun, {
    frequency: -0.02,
    radiusX: 55,
    radiusY: 80,
    Phase: Math.PI / 4
})
```

A positive frequency will make the planet go clockwise, a negative one will go anti-clockwise. Having different radius values for X and Y causes the orbit to be elliptical. The phase can be used to have several planets in orbit at the same distance without clashing with each other. The values above should make the galaxy look like this:



Your Challenge

Can you create another planet, and either put it in orbit around the sun, or put it in orbit around your first planet so it acts like a moon? You will need to do the following:

- Call galaxy.newPlanet() and pass in the customisation object for the planet
- Take the return value from that function into a variable called **planet2** or **moon**.
- Call .orbit() on your **planet2/moon** and pass in the variable for the planet you wish to orbit.

It's all the same code that you have already written, so you can copy and paste what you have already and amend it to suit your needs.

Custom Animation

The galaxy is already animated for you with things like orbits. This is provided for you.

You can build your own animation into the galaxy by using the setInterval JavaScript function and calling various property functions on your planets/suns to modify them after creation.

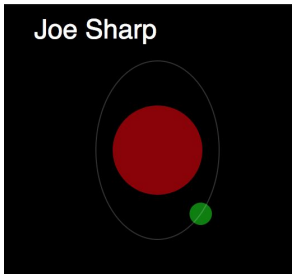
For instance, let's make your sun change colour with time, add the following code:

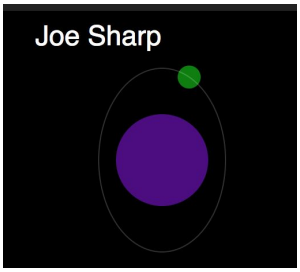
```
// The sun will rotate through these colours
const sunColours = ['red', 'darkred', 'indigo']

// This index will track which colour we are using
let i = 0

setInterval(() => {
    sun.setColour(sunColours[i])
    i += 1
    i %= sunColours.length
}, 500) // will change colour every 500 milliseconds
```

Here are some samples of the sun as its colour changes.





You can change whatever you want in your animation code, here is another example that causes the sun to pulse, by changing its radius with time.

```
// Demonstration of a pulsing sun using Math.sin to get a rotating value
setInterval(() => {
    const angle = 2 * Math.PI * new Date().getTime() / 1000;
    sun.setRadius(35 + (5 * Math.sin(angle)))
})
```

Galaxy API Reference

Here is a complete example galaxy that uses every single configuration option available.

```
export default (galaxy) => {
   galaxy.setName('API Reference')

   // Creating a sun will place a static circle onto the galaxy
   // Sun and Planet are functionally identical, I have created two
   // names just so it reads clearly.
   const sun = galaxy.newSun({
       colour: 'blue',
       radius: 50,
       positionX: 150,
       positionY: 130
   })

   // This actually creates the same thing as newSun
   // So the configuration is identical
   const planet = galaxy.newPlanet({
       colour: 'yellow',
       radius: 10,
       // not bothering to set position X and Y, as I will use orbit
   })

   planet.orbit(sun, {
       frequency: 0.05,
       phase: Math.PI / 2, // expressed in Radians
       radiusX: 90,
       radiusY: 70
   })

   // You can create a planet and put into orbit using method chaining
   // Here I will create a moon
   const planet2 = galaxy.newPlanet({
       colour: 'turquoise',
       radius: 5
   }).orbit(planet, {
       radius: 20,
       frequency: 0.5
   })
```

```
    // stars can be placed around the galaxy and they will pulse
    // Here I use a loop to create 10 stars
    const stars = []
    for (let x=0; x<10; x++) {
        let star = galaxy.newStar({
            positionX: Math.random() * 250,
            positionY: 40 + (Math.random() * 200),
            baseSize: Math.random() * 2, // stars will pulse,
            phase: Math.random() * 2 * Math.PI,
            sizeRange: 1.0, // how much the star should shrink and grow by
            frequency: 0.5, // set the speed of the pulse
        })
        stars.push(star)
    }

    // We can put rockets into our galaxy and set them to fly between whatever we want
    const planetRocket = galaxy.newRocket({
        colour: 'GreenYellow',
        size: 3,
        speed: 3
    })
    planetRocket.flyBetween([planet, planet2])

    // This rocket uses method chaining, and will fly between all the stars
    galaxy.newRocket({
        colour: 'Chartreuse',
        size: 2,
        speed: 1
    }).flyBetween(stars)
}
```

Have fun!