JavaScript Refresher

In this exercise we will refresh our knowledge of JavaScript. The exercises are mainly focused on writing JavaScript for use with the HTML5 Canvas. The examples given at each step are about dogs. At each step, you should adapt the examples to insert the relevant code in canvas.html to model a ball.

Exercises

1. Below is JavaScript code for declaring a variable called age, and setting it to 1, and another variable called name and setting it to Toby. In canvas.html, create two variables, called xpos and ypos. Set xpos to 10 and ypos to 20.

```
var age = 1;
var name = "Toby";
```

2. Open canvas.html in your browser. Open the javascript console, check that xpos and ypos have been defined. Then change the code to log the values of xpos and ypos to the console automatically.

```
>> xpos
← 10
```

3. Below is an example, of a function called "add_to_age" which adds "`years`" to "age". In canvas.html, create a JavaScript function called move(), which take two arguments, one value to add to xpos, and the other to add to ypos. Check that your function works at the JavaScript console.

```
var age = 1;

function add_to_age(years) {
  age = age + years;
}
```

4. In the example below we create an object called dog, with three properties: a name, a breed and an age. In your own code, create an object called ball, that has the following properties: xpos (its x co-ordinate), ypos (its y co-ordinate) and r (its radius).

```
var dog = {
  name: "Toby",
  breed: "Labrador",
  age: 5
};
```

5. Below we add a method in the dog object to increase the dog's age by "years". Define a function/method in ball called move(), which moves (increments) the xpos and ypos co-ordinates according to 2 input parameter values.

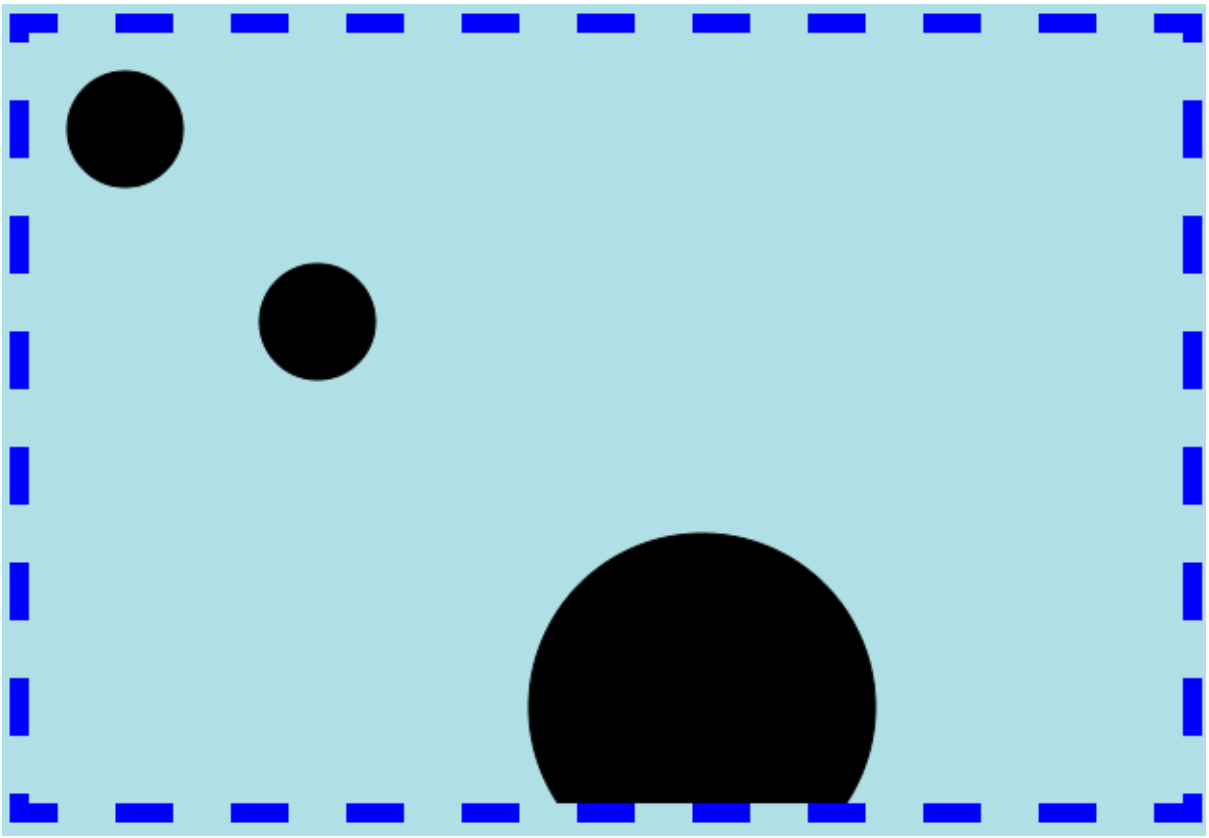
```
var dog = {                 //Define 'dog' object
  name: "Toby",
  breed: "Labrador",
  age: 5,
  add_to_age: function(years) {     //Function definition
      this.age = this.age + years;
  }
};

dog.add_to_age(5);  //Example of calling the function (which is inside dog)
```

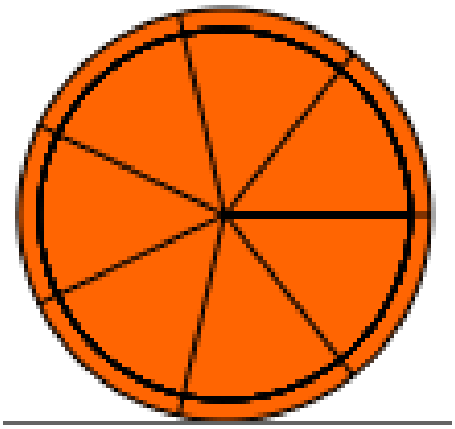
6. Create a second method in ball, called resize, which takes one argument. This should set the radius (r) of the ball to the value of the argument.

7. Create a third method in ball, called draw. The function should take no arguments but should draw the ball according to the xpos, ypos, r parameter settings

8. Call the methods so that each of the draw, move and resize methods are demonstrated on the canvas. E.g.



9. Add to the ball class so that the ball now looks like the pizza from Monday's lecture but divided into only 7 slices. Redefine the move, draw and resize methods for the new object and demonstrate their functionality. Include comments in your code to describe your thought process.



Advanced exercises

1. Add a rotate method (that rotates the pizza by a certain number of degrees/radians). This will give the illusion that that the ball is rolling when animated.

2. Create a method called contains() in ball, that takes two arguments - one an x co-ordinate and the other a y co-ordinate. The method should return true if the point given by those co-ordinates is in or on the circle.

3. Re-create the ball object using a constructor function instead of an object literal. You should name this object Ball instead of ball.

4. Create a method in the Ball object called intersect, which takes another Ball as an argument and returns true if the Balls are touching/intersecting and false otherwise.

Note

- There is an excellent JavaScript refresher tutorial here: A re-introduction to JavaScript (JS tutorial)
- Mozilla have a piece on objects here: Working with objects.