JS Components

1 Javascript Objects

1.1 Collection of properties

Each property is named (with a key) and has a value
In Javascript Object Notation (JSON) we can write

```
let ball = {x: 200, y: 300, radius: 50};
```

1.2 obj.prop

Access and properties like this

```
ellipse(ball.x, ball.y, ball.radius*2, ball.radius*2);
ball.x += 5;
ball.z = 8;
ball["colour"] = "red";
```

The bottom line has the same effect as ball.colour=”red”;

1.3 Function-valued properties

Object properties can be any type, including functions

```
ball.draw = function(){ alert("I am a ball");}
ball.draw();
```

1.4 this

`this` refers to the object it was called on

```
ball.draw = function(){
ellipse(this.x, this.y, this.radius*2, this.radius*2);
}
ball.draw();
```

1.5 Prototypal Inheritance

- Every object has a property `__proto__` which refers to another object
- If a property isn't found in an object's own properties, then `__proto__` is checked
- Every function has a property `prototype` which can be used when creating an object
- The `new` keyword is used with a constructor function to create an object and set its `__proto__`
- Read more at MDN

1.6 Inheriting behaviour

- In other languages (e.g Java, C#) every object belongs to a *class*
  - Data values (fields) are associated with objects
  - Behaviour (methods) are associated with classes
- Things of the same type (class) can do the same things
- JS is more flexible: each object can define its own behaviour
- JS allows inheritance (common behaviour) through prototypes
- Java uses *class-based inheritance* (object to class)
- JS use *prototypal inheritance* (object to object)

1.7 Emulating classes in JS

Simple syntax for constructors and prototype functions

```
class Ball{
constructor(x, y, r){
this.x = x;
this.y = y;
this.radius = r;
}
draw(){
ellipse(this.x, this.y,
this.radius*2, this.radius*2);
}
}
let b = new Ball(400,300,20);
b.draw();
```

1.8 Why classes?

- Reduces cut-and-paste: eases maintenance
- Encourages *encapsulation*: hide the details so they can be changed easily
- Make reusable compoments with classes
- Reuse in the same project (multiple balls) or in different projects