ES6 classes

Classes are a template for creating objects They encapsulate the data and your methods (finally everything in one place)

So here's what our code will look like

```
class Player {
        constructor(name, battery) {
                this.name = name;
                this.battery = battery;
        }
        charge() {
                this.battery = 100;
        }
}
```

Classes have constructor functions that run every time we instantiate a class (use the new keyword)

All the methods (charge) can also be encapsulated inside the class itself

Creating an instance of a class simply means we're creating an object from the class.

> let player1 = new Player('John', 20)

by using the new keyword we're creating an instance (Obj player1)

Note :- Classes are only syntactic sugar, the class keyword is just like using prototypal inheritance (behind the scene) So it dosen't have the real class concept like it exists in other languages.

Interview question : Why are methods not put inside the constructor?

→ When we use new keyword, an instance is created based on the constructor function (name, battery) If we put the method (charge) inside constructor then it will be created ~~by~~ with every object which is memory inefficient. So by keeping it outside all objects will reference to charge which is present in one place in the memory