JavaScript Classes

This lesson will cover all the basic concepts of Classes in JavaScript including the implementation part.

WE'LL COVER THE FOLLOWING

Most object-oriented languages use classes as *abstractions* for the ideas or concepts manipulated by a program. A *class* is used to create objects representing a concept. It offers a convenient syntax to give both *data* and *behavior* to these objects. JavaScript is no exception and supports programming with classes (but with a twist — more on that later).

Creating a Class

Our example RPG deals with characters, so let's create a `Character` class to express what a character is.

```
class Character {
  constructor(name, health, strength) {
    this.name = name;
    this.health = health;
    this.strength = strength;
    this.xp = 0; // XP is always zero for new characters
  }
  // Return the character description
  describe() {
    return `${this.name} has ${this.health} health points, ${this
      .strength} as strength and ${this.xp} XP points`;
  }
}
```

This example demonstrates several key facts about JavaScript classes:

- A class is created with the `class` keyword, followed by the name of the class (usually starting with an uppercase letter).

- Contrary to object literals, there is no separating punctuation between the elements inside a class.
- A class can only contain *methods*, not data properties.
- Just like with object literals, the `this` keyword is automatically set by JavaScript inside a method and represents *the object on which the method was called*.
- A special method named `constructor()` can be added to a class definition. It is called during object creation and is often used to give it data properties.

Using a Class

Once a class is defined, you can use it to create objects. Check out the rest of the program.

```
const aurora = new Character("Aurora", 150, 25);
const glacius = new Character("Glacius", 130, 30);

// Aurora is harmed by an arrow
aurora.health -= 20;

// Aurora gains a strength necklace
aurora.strength += 10;

// Aurora learns a new skill
aurora.xp += 15;

console.log(aurora.describe());
console.log(glacius.describe());
```

The `aurora` and `glacius` objects are created as characters with the `new` operator. This statement calls the class constructor to initialize the newly created object. After creation, an object has access to the properties defined inside the class. Here's the canonical syntax for creating an object using a class.

```
class MyClass {
  constructor(param1, param2, ...) {
    this.property1 = param1;
    this.property2 = param2;

    // ...
  }
  method1(/* ... */) {
    // ...
  }
  method2(/* ... */) {
    // ...
  }
  // ...
}

const myObject = new MyClass(arg1, arg2, ...);
myObject.method1(/* ... */);
// ...
```