Getter and setter

Getter and setter methods are special functions in JavaScript classes that allow you to define how properties are accessed and assigned. They provide a way to control access to object properties and add validation or custom behavior when getting or setting values.

Here's an explanation with examples for both getter and setter methods:

Getter Methods:

Getter methods are used to retrieve the value of a property from an object. They are defined using the get keyword followed by the method name.

```
class Circle {
    constructor(radius) {
        this.radius = radius;
    }

    // Getter method for the area property
    get area() {
        return Math.PI * this.radius ** 2;
    }
}

const circle = new Circle(5);
console.log(circle.area); // Output:
78.53981633974483
```

In this example:

- **We define a Circle class with a constructor that initializes the radius property.**
- **We define a getter method area using the get keyword. When circle.area is accessed, this method is called, and it calculates and returns the area of the circle based on the radius property.**
- **Setter Methods:**
- **Setter methods are used to update the value of a property in an object. They are defined using the set keyword followed by the method name.**

Setter Methods:

Setter methods are used to update the value of a property in an object. They are defined using the `set` keyword followed by the method name.

```
class Circle {
    constructor(radius) {
        this.radius = radius;
    }

    // Getter method for the area property
    get area() {
        return Math.PI * this._radius ** 2; //
Corrected to _radius
    }

    // Setter method for the radius property
    set radius(value) {
        if (value <= 0) {
            throw new Error('Radius must be a
positive number');
        }
        this._radius = value; // Corrected to
_radius
    }
}

const circle = new Circle(5);
circle.radius = 7;
console.log(circle.area); // Output:
153.93804002589985
```

In this example:

- **We define a Circle class with a constructor that initializes the radius property.**
- **We define a getter method area to calculate the area of the circle based on the radius property.**
- **We define a setter method radius using the set keyword. When circle.radius = 7 is called, this method is invoked, and it sets the radius property to the provided value after validating it. In this case, it also checks if the value is positive, throwing an error if not.**

Using Getters and Setters:

```
class Circle {
    constructor(radius) {
        this.radius = radius;
    }

    // Getter method for the area property
    get area() {
        return Math.PI * this._radius ** 2;
    }

    // Setter method for the radius property
    set radius(value) {
        if (value <= 0) {
            throw new Error('Radius must be a
positive number');
        }
        this._radius = value;
    }
}

const circle = new Circle(5);
console.log(circle.area); // Output:
78.53981633974483
circle.radius = 7;
console.log(circle.area); // Output:
153.93804002589985
```

In this final example:

- **We create a Circle object with a radius of 5.**
- **We access the area property using the getter method circle.area, which calculates and returns the area of the circle.**
- **We update the radius property to 7 using the setter method circle.radius = 7, which sets the new radius value after validation.**
- **We again access the area property, which now reflects the updated area of the circle based on the new radius value.**