Chapter 25: Setters and Getters

Setters and getters are object properties that call a function when they are set/gotten.

Section 25.1: Defining a Setter/Getter Using Object.defineProperty

```javascript
var setValue;
var obj = {};
Object.defineProperty(obj, "objProperty", {
    get: function(){
        return "a value";
    },
    set: function(value){
        setValue = value;
    }
});
```

Section 25.2: Defining an Setter/Getter in a Newly Created Object

JavaScript allows us to define getters and setters in the object literal syntax. Here's an example:

```javascript
var date = {
    year: '2017',
    month: '02',
    day: '27',
    get date() {
        // Get the date in YYYY-MM-DD format
        return `${this.year}-${this.month}-${this.day}`
    },
    set date(dateString) {
        // Set the date from a YYYY-MM-DD formatted string
        var dateRegExp = /(\d{4})-(\d{2})-(\d{2})/;

        // Check that the string is correctly formatted
        if (dateRegExp.test(dateString)) {
            var parsedDate = dateRegExp.exec(dateString);
            this.year = parsedDate[1];
            this.month = parsedDate[2];
            this.day = parsedDate[3];
        }
        else {
            throw new Error('Date string must be in YYYY-MM-DD format');
        }
    }
};
```

Accessing the `date.date` property would return the value `2017-02-27`. Setting `date.date = '2018-01-02` would call the setter function, which would then parse the string and set `date.year = '2018'`, `date.month = '01'`, and `date.day = '02'`. Trying to pass an incorrectly formatted string (such as `"hello"`) would throw an error.

Section 25.3: Defining getters and setters in ES6 class

```javascript
class Person {
  constructor(firstname, lastname) {
```

```javascript
    this._firstname = firstname;
    this._lastname = lastname;
  }

  get firstname() {
    return this._firstname;
  }

  set firstname(name) {
    this._firstname = name;
  }

  get lastname() {
    return this._lastname;
  }

  set lastname(name) {
    this._lastname = name;
  }
}

let person = new Person('John', 'Doe');

console.log(person.firstname, person.lastname); // John Doe

person.firstname = 'Foo';
person.lastname = 'Bar';

console.log(person.firstname, person.lastname); // Foo Bar
```