Getter and setter functions

ES6 has started supporting getter and setter functions. Using the following example:

```
class Employee {

    constructor(name) {
        this._name = name;
    }

    get name() {
      if(this._name) {
        return 'Mr. ' + this._name.toUpperCase();
      } else {
        return undefined;
      }
    }

    set name(newName) {
      if (newName == this._name) {
        console.log('I already have this name.');
      } else if (newName) {
        this._name = newName;
      } else {
        return false;
      }
    }
}

var emp = new Employee("James Bond");

// uses the get method in the background
if (emp.name) {
  console.log(emp.name);  // Mr. James Bond
}

// uses the setter in the background
emp.name = "Bond 007";
console.log(emp.name);  // Mr. Bond 007
```

Latest browsers are also supporting getter/setter functions in Objects and we can use them for computed properties, adding listeners and preprocessing before setting/getting:

```
var person = {
  firstName: 'James',
  lastName: 'Bond',
  get fullName() {
      console.log('Getting FullName');
      return this.firstName + ' ' + this.lastName;
  },
  set fullName (name) {
      console.log('Setting FullName');
      var words = name.toString().split(' ');
      this.firstName = words[0] || '';
      this.lastName = words[1] || '';
  }
}

person.fullName; // James Bond
person.fullName = 'Bond 007';
person.fullName; // Bond 007
```