

# 11 Encapsulation

## ■ Grouping Data & Actions

Encapsulation bundles data (variables) and methods (functions) together within an object.

## ■ Hides Complexity

It hides the inner workings, allowing interaction only through a set of well-defined methods

## ■ Data Protection

Protects data from direct access or modification from outside the object

## ■ Controlled Access

Enables controlled access to the data, ensuring it's only changed in expected ways.

## ■ Enhances Maintainability

Encapsulation makes code more manageable by organizing related functionalities and protecting internal data.

## Encapsulation Using Closures

```
function createCounter() {
  let count = 0;

  return {
    increment: function() {
      count++;
    },
    getCount: function() {
      return count;
    }
  };
};

const counter = createCounter();
counter.increment();
console.log(counter.getCount()); // Output: 1
console.log(counter.count); // Output: undefined (count is encapsulated)
```

## Using Constructor Functions

```
function Car(make, model) {
  let mileage = 0;

  this.getMake = function() {
    return make;
  };

  this.getModel = function() {
    return model;
  };

  this.getMileage = function() {
    return mileage;
  };

  this.drive = function(distance) {
    mileage += distance;
  };
}

const myCar = new Car('Toyota', 'Corolla');
console.log(myCar.getMake()); // Output: Toyota
```

```
console.log(myCar.getMileage()); // Output: 0
myCar.drive(100);
console.log(myCar.getMileage()); // Output: 100
console.log(myCar.mileage); // Output: undefined (mileage is encapsulated)
```

## Using ES6 Classes

```
class BankAccount {
  #balance = 0; // Private field

  deposit(amount) {
    this.#balance += amount;
  }

  withdraw(amount) {
    if (amount <= this.#balance) {
      this.#balance -= amount;
    } else {
      console.log("Insufficient balance");
    }
  }

  getBalance() {
    return this.#balance;
  }
}

const acc = new BankAccount();
acc.deposit(100);
console.log(acc.getBalance()); // Output: 100
console.log(acc.#balance); // Output: Error (private field is encapsulated)
```

#JavaScript\_OOP