# Part 5

## Classes

JavaScript classes were introduced as part of ES6. The syntax is similar to other languages. Classes formalise the JavaScript pattern of class-like inheritance from ES5. This used to be achieved with functions and prototypes.

class Dog {

constructor(name, age) {

this.name = name;

this.age = age;

}

bark() {

console.log('woof');

}

}

let dog = new Dog('bob', 10);

dog.bark();

In this example a class of Dog is defined with properties name and age, and a method called bark. When call **new Dog(…)**, we’re getting a new instance of that class. Calling **dog.bark()** is then calling the method on that instance, thus it has access to the name and age variables of that instance.

Using the **this** keyword in classes refers to the class instance.

### Static Members

Static methods aren’t called on the instance of a class, **they’re called on the class itself**.

They’re often used for utilities that don’t need context of a class instance.

class Dog {

constructor(name, age, pawSize) {

this.name = name;

this.age = age;

this.pawSize = pawSize;

}

static defaultDog() {

return new Dog('default', 0);

}

}

let c = new Dog.defaultDog();

In this example we’re calling **Dog.defaultDog**, which is returning an **instance** of the class. The static method **defaultDog** doesn’t have context for variables like name, age and pawSize because it is a static member.

## Destructuring

Useful syntax for grabbing properties from an object and assigning them to variables.

It be applied to:

* Parameters
* Arrays
* Any object

// New Way

const list = [1, 2, 3];

let [a, , b] = list;

[b, a] = [a, b];

// Old Way

const list = [1, 2, 3];

let a = list[0], b = list[2];

let tmp = a; a = b; b = tmp;

### Spread Operator

Using the spread operator you can easily access the remaining properties or array items.

Spreading can be used to merge things too!

* Combine two or more objects together with **{ …objA, …objB}.**
* Combine two or more arrays together with **[…arrayA, …arrayB]**

// Accessing the rest of an array

const arr = [1, 2, 3, 4, 5];

const [first, ...rest] = arr;

// first = 1, rest = [2, 3, 4, 5]

// Merging two objects together

const objA = { name: 'Adam', age: 20 };

const objB = { age: 21, gender: 'male'};

const merged = {...objA, ...objB};

/\* merged = {

name: 'Adam',

age: 21,

gender: 'male' }

\*/

The spread operator can also be used with parameters if you have a function where you don’t know how many inputs to expect:

function arrayToString(...args) {

    return args.join(', ');

}

const x = arrayToString('a', 'b', 'c');

// x = ‘a, b, c’