



CLASSES AND OBJECTS

MET CS
Andrew Sheehan

PRIMITIVE & REFERENCE TYPES

Almost everything in the Javascript language deals with Objects.

PRIMITIVE TYPES

Boolean, Number, String, Null
and Undefined

REFERENCE TYPES

Otherwise known as Objects.

Objects are an list of
properties (name and a value)

Just like a Hashtable

OBJECT PROPERTIES

Data attributes can be dynamically added & removed.

```
const Car = {  
  year = 1900  
};
```

```
Car.color = "not indicated"; // Adding a property
```

```
delete Car.year // Deleting a property
```

USING FUNCTIONS TO CREATE OBJECTS

// “Constructor Functions” are one way to create
// user-defined objects.

```
const Car = function(color) {  
  this.color = color,  
}
```

FUNCTIONS

ADDING METHODS/ATTRIBUTES

```
const Car = function(color) {  
  this.color = color,  
}
```

```
Car.prototype.run = function() {  
  // logic ..  
}  
Car.prototype.currentSpeed = 0;
```

USING CLASS TO CREATE OBJECTS

// behind the scenes, its using functions...

```
class Car {
```

```
    // No defined constructor? A default exists.
```

```
    constructor(color) {  
        this.color = color;
```

```
    }
```

```
}
```


USING CLASS

CAN YOU USE? (YES)

ES6 classes - OTHER

Usage

% of all users

Global

92.83% + 0.16% = 92.99%

ES6 classes are syntactical sugar to provide a much simpler and clearer syntax to create objects and deal with inheritance.

Current aligned

Usage relative

Date relative

Apply filters

Show all

?

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *
			4-41		10-28			
		2-44	42-48	3.1-8	29-35	3.2-8.4		
6-10	12-17	45-68	49-75	9-12	36-60	9-12.1		2.1-4.4.4
11	18	69	76	12.1	62	12.3	all	67
	76	70-71	77-79	13-TP		13		

USING CLASS EXAMPLE

// instantiating the object (“An instance of Car”)

```
const myCar = new Car(“red”);
```

CLASS

METHOD DECLARATIONS



```
class Food {  
  
    constructor (name, protein, carbs, fat) {  
        this.name = name;  
        this.protein = protein;  
        this.carbs = carbs;  
        this.fat = fat;  
    }  
  
    toString () {  
        return `${this.name} | ${this.protein}`;  
    }  
  
    print () {  
        console.log( this.toString() );  
    }  
}
```

CLASS

STATIC METHOD DECLARATIONS

```
1 class ClassWithStaticMethod {  
2     static staticMethod() {  
3         return 'static method has been called.';  
4     }  
5 }  
6  
7 console.log(ClassWithStaticMethod.staticMethod());  
8 // expected output: "static method has been called."  
9  
10
```