


 This repository

[Pull requests](#) [Issues](#) [Gist](#)


 

 [Propulsion-Academy / fswd-2017-01](#) Private




[Watch](#) 5 [Star](#) 0 [Fork](#) 1

[Code](#) [Issues](#) 0 [Pull requests](#) 0 [Projects](#) 0 [Wiki](#) [Pulse](#) [Graphs](#)

Branch: master ▾ **fswd-2017-01** / js-fundamentals / day3 / objects.md [Find file](#) [Copy path](#)

 Llorenç Muntaner first week 08934d7 7 days ago

0 contributors

44 lines (30 sloc) 1.55 KB [Raw](#) [Blame](#) [History](#)   

Javascript Objects and classes

We have learned to create classes and instances of classes using constructors, the keyword `new` and adding methods to the prototype of the constructor.

However, it is **important that you understand that there is no such thing as classes in Javascript. The steps we learned are just syntactic sugar for developers coming from OOP programming languages.**

The instances created with the `new` keyword are still objects. They are nothing fancy. Just plain literal objects. We could have created them without the `new` keyword and it everything could have worked.

In order to add methods and properties to an object it can be done directly.

```
var writer = {
  name: 'Fante',
  sayHello: function() {
    console.log(this.name + ' says hello');
  }
};

writer.sayHello();
// this prints 'Fante says hello'
```

the previous is almost identical to:

```
var Writer = function(name) {
  this.name = name;
}

Writer.prototype.sayHello = function() {
  console.log(this.name + ' says hello');
}

var fante = new Writer('Fante');
fante.sayHello();
// this also prints 'Fante says hello'
```

We will not go into the details of why the keyword `new` is used and what it does for us now. For now learn how to use it. Most of the projects written in object oriented javascript use this pattern to create classes and instances.

In the Advanced Javascript week we will go into the details and the reasons to use the Classical Inheritance Pattern.

Lesson: There is no such thing as instance and classes in JS. Everything are just objects

