[1 Classes 2](#_Toc451945821)

[1.1 As a function 2](#_Toc451945822)

[1.1.1 Header 2 2](#_Toc451945823)

[1.1.1.1 Header 3 2](#_Toc451945824)

[1.1.1.1.1 Header 4 2](#_Toc451945825)

1. Classes

# As a function

Create a function, which acts as an object constructor, using ‘this’ within the function, which acts as a class. ‘this’ keyword ensures uniqueness of functions/properties within the object/class. Keeps them out of the ‘global’ namespace.



Functions can typically be added ‘after-the-fact’ via prototyping (adding functionality to the ‘prototype’ of the object/class), which makes the function itself sort of ‘static’ to the class – being shared across all instances of the object, and, ergo, more inexpensive.

function Apple (type) {

this.type = type;

this.color = "red";

}

Apple.prototype.getInfo = function() {

return this.color + ' ' + this.type + ' apple';

};

# Singletons:

One offs.

var apple = new function() {

this.type = "macintosh";

this.color = "red";

this.getInfo = function () {

return this.color + ' ' + this.type + ' apple';

};

}}