

<http://jsfiddle.net/zymsys/nUS7k/>

//Object Literal Demo

```
var o = {
  name: 'Lemon',
  sayHello: function() {
    document.write("<div>I'm a " + this.name + "</div>");
  }
};

o.sayHello();
```

<http://jsfiddle.net/zymsys/PVRyS/>

//Constructor demo

```
var Fruit = function(name) {
  this.name = name;
  this.sayHello = function() {
    document.write("<div>I'm a " + this.name + "</div>");
  }
};

var o = new Fruit('Lemon');
var k = new Fruit('Kiwi');
var c = new Fruit('Banana');

o.sayHello();
k.sayHello();
c.sayHello();
```

<http://jsfiddle.net/zymsys/zVfE4/>

//The importance of 'new'

```
var Fruit = function(name) {
  this.name = name;
  this.sayHello = function() {
    document.write("<div>I'm a " + this.name + "</div>");
  }
};

var o = Fruit('lemon'); //Forgot new!!!
o.sayHello(); //Fails. Also, name and sayHello are in the global namespace.
```

<http://jsfiddle.net/zymsys/KgHhN/>

//The importance of 'new'

```
var Fruit = function(name) {
  if (!(this instanceof Fruit)) {
    return new Fruit(name);
  }
  this.name = name;
  this.sayHello = function() {
    document.write("<div>I'm a " + this.name + "</div>");
  }
};

var o = Fruit('lemon'); //Forgot new!!!
o.sayHello(); //Works anyway thanks to our check above.
```

<http://jsfiddle.net/zymsys/6Dq2W/>

//This and That demo

//Part 1: The problem with this

```
var Fruit = function(name) {
  this.name = name;
  this.sayHello = function() {
    document.write("<div>I'm a " + this.name + "</div>");
  }
};

var o = new Fruit('Lemon');

var k = {
  name: 'Kiwi',
  sayHello: o.sayHello
};

var c = {
  name: 'Banana'
};

o.sayHello();
k.sayHello();
o.sayHello.apply(c);
```

<http://jsfiddle.net/zymsys/nthJA/>

//This and That demo

//Part 2: That solution to this problem

```
var Fruit = function(name) {
  var that = this;
  this.name = name;
  this.sayHello = function() {
    document.write("<div>I'm a " + that.name + "</div>");
  }
};

var o = new Fruit('Lemon');

var k = {
  name: 'Kiwi',
  sayHello: o.sayHello
};

var c = {
  name: 'Banana'
};

o.sayHello();
k.sayHello();
o.sayHello.apply(c);
```

<http://jsfiddle.net/zymsys/E2pUX/>

```
//Private Variables
```

```
//Taking this and that out of the picture
```

```
var Fruit = function(name) {  
    var message = "<div>I'm a " + name + "</div>";  
    this.sayHello = function() {  
        document.write(message);  
    };  
};
```

```
var o = new Fruit('Lemon');
```

```
var k = {  
    name: 'Kiwi',  
    sayHello: o.sayHello  
};
```

```
var c = {  
    name: 'Banana'  
};
```

```
o.sayHello();  
k.sayHello();  
o.sayHello.apply(c);
```

<http://jsfiddle.net/zymsys/NVCyU/>

//Inheritance (pure JavaScript)

//Confusing and error prone.

```
var Fruit = function(name) {
  if (name) {
    this.name = name;
  }
  this.sayHello = function() {
    document.write("<div>I'm a " + this.name + "</div>");
  };
};

var AcidicFruit = function(name, ph) {
  Fruit.call(this, name);
  this.getAcidity = function() {
    return ph;
  };
};
AcidicFruit.prototype = new Fruit();

var o = new AcidicFruit('Lemon', 2);

o.sayHello();
document.write("<div>My pH is " + o.getAcidity() + "</div>");
```

<http://jsfiddle.net/zymsys/6uCgk/>

//Inheritance (via jQuery.extend and no constructors)
//Improvement over <http://jsfiddle.net/zymsys/NVCyU/> ?
//I think so.

```
var Fruit = {
  name: function (newName) {
    if (newName) {
      this._name = newName;
      return this;
    } else {
      return this._name;
    }
  },
  sayHello: function() {
    document.write("<div>I'm a " + this.name() + "</div>");
  }
};

var AcidicFruit = {
  _ph: 7,
  ph: function(newPh) {
    if (newPh) {
      this._ph = newPh;
      return this;
    } else {
      return this._ph;
    }
  }
};

var o = $.extend({}, AcidicFruit, Fruit).name('Lemon').ph(2);

o.sayHello();
document.write("<div>My pH is " + o.ph() + "</div>");
```

<http://jsfiddle.net/zymsys/xpgQA/>

//Parts / Traits

```
var Fruit = function(name) {
  this.sayHello = function() {
    document.write("<div>I'm a " + name + "</div>");
  };
};

function addPh(that) {
  var ph = 7; //Neutral
  that.acidity = function(newValue) {
    if (typeof newValue == 'undefined') {
      return ph;
    } else {
      ph = newValue;
      return that;
    }
  };
  return that;
}

var o = new Fruit('Lemon');
addPh(o).acidity(2);
o.sayHello();
document.write("<div>My acidity is " + o.acidity() + ".</div>");
```