

object.prototype.method = function()	object.prototype = Object.create(object); object.prototype.constructor = object	object	object = Object.create(object)	object.prototype.method = function()	object.prototype = object.prototype; object.prototype.constructor = object
<pre>function K(x, y) { this.x = x; this.y = y; } Ka = {}; K.prototype = function() { return 'K' + x; }; K.prototype.add = function() { return this.x + this.y; }; K.prototype.sub = function() { return this.x - this.y; }; console.log('K: Ka, K: a'); console.log('K: Str("K")', K.prototype); // KStr("K") console.log('K: K'); [Object Function] Str [Object Function] a = "K" name "K" prototype [Object Object] add [Object Function] sub [Object Function] constructor [Object Function] Str [Object Function] a = "K" name "K" prototype [Object Object] _proto_ [Object Object] constructor [Object Function] add [Object Function] sub [Object Function] console.log('K: _proto_, K: _proto_'); console.log('K: prototype, K: prototype'); [Object Object] add [Object Function] sub [Object Function] constructor [Object Function] Str [Object Function] a = "K" name "K" prototype [Object Object] _proto_ [Object Object] constructor [Object Function] add [Object Function] sub [Object Function] var k = new K(1, 2); // console.log('k: a, k: a'); // "K" undefined // console.log('k: Str("K")', k.constructor); // TypeError: k.constructor is not a function console.log('k: a'); [Object Object] _proto_ [Object Object] constructor [Object Function] Str [Object Function] a = "K" name "K" prototype [Object Object] _proto_ [Object Object] console.log('K: _proto_, k: _proto_'); console.log('K: prototype, k: prototype'); [Object Function] Str [Object Function] a = "K" name "K" prototype [Object Object] _proto_ [Object Function]</pre>	<pre>function H(x, y) { this.x = x; this.y = y; } H.prototype = Object.create(K.prototype); H.prototype.constructor = H; H.prototype.add = function() { return this.x + this.y; }; H.prototype.sub = function() { return this.x - this.y; }; // console.log('H: Ka, H: a'); // "H" undefined // console.log('H: Str("H")', H.prototype); // TypeError: H.prototype is not a function console.log('H: H'); [Object Function] Str [Object Function] name "H" prototype [Object Object] constructor [Object Function] pre [Object Function] name "H" prototype [Object Object] _proto_ [Object Object] _proto_ [Object Object] _proto_ [Object Object] _proto_ [Object Object] console.log('H: _proto_, H: _proto_'); console.log('H: prototype, H: prototype'); [Object Function] Str [Object Function] name "H" prototype [Object Object] constructor [Object Function] pre [Object Function] name "H" prototype [Object Object] _proto_ [Object Object] _proto_ [Object Object] var h = new H(1, 2); // console.log('h: a, h: a'); // "H" undefined // console.log('h: Str("H")', h.constructor); // TypeError: h.constructor is not a function console.log('h: a'); [Object Object] _proto_ [Object Object] constructor [Object Function] pre [Object Function] name "H" prototype [Object Object] _proto_ [Object Object] _proto_ [Object Object] console.log('H: _proto_, h: _proto_'); console.log('H: prototype, h: prototype'); [Object Function] Str [Object Function] a = "H" name "H" prototype [Object Object] _proto_ [Object Function]</pre>	<pre>var o = {}; o.x = 1; o.y = 2; o._proto_ [Object Object] constructor [Object Function] a = "K" name "K" prototype [Object Object] _proto_ [Object Object] console.log('o: a'); [Object Object] _proto_ [Object Object] constructor [Object Function] a = "K" name "K" prototype [Object Object] _proto_ [Object Object] var o = Object.create(a); o.x = 1; // console.log('o: a, o: a'); // "a" undefined // console.log('o: Str("a")', o.constructor); // TypeError: o.constructor is not a function console.log('o: a'); [Object Object] _proto_ [Object Object] constructor [Object Function] a = "a" name "a" prototype [Object Object] _proto_ [Object Object] var o = {}; o.x = 1; o.y = 2; o._proto_ [Object Object] constructor [Object Function] a = "a" name "a" prototype [Object Object] _proto_ [Object Object]</pre>	<pre>function F(x, y) { this.x = x; this.y = y; } Fa = {}; F.prototype = function() { return 'F' + x; }; F.prototype.add = function() { return this.x + this.y; }; F.prototype.sub = function() { return this.x - this.y; }; console.log('F: Fa, F: a'); console.log('F: Str("F")', F.prototype); // FStr("F") console.log('F: F'); [Object Function] Str [Object Function] a = "F" name "F" prototype [Object Object] add [Object Function] sub [Object Function] constructor [Object Function] Str [Object Function] a = "F" name "F" prototype [Object Object] _proto_ [Object Object] _proto_ [Object Object] console.log('F: _proto_, F: _proto_'); console.log('F: prototype, F: prototype'); [Object Object] add [Object Function] sub [Object Function] constructor [Object Function] Str [Object Function] a = "F" name "F" prototype [Object Object] _proto_ [Object Object] _proto_ [Object Object] var f = new F(1, 2); // console.log('f: a, f: a'); // undefined // console.log('f: Str("F")', f.constructor); // TypeError: f.constructor is not a function console.log('f: a'); [Object Object] _proto_ [Object Object] constructor [Object Function] Str [Object Function] a = "F" name "F" prototype [Object Object] _proto_ [Object Object] _proto_ [Object Object]</pre>	<pre>function G(x, y) { this.x = x; this.y = y; } Ga = {}; G.prototype = function() { return 'G' + x; }; G.prototype.add = function() { return this.x + this.y; }; G.prototype.sub = function() { return this.x - this.y; }; // console.log('G: Ga, G: a'); // undefined // console.log('G: Str("G")', G.prototype); // TypeError: G.prototype is not a function console.log('G: G'); [Object Function] Str [Object Function] name "G" prototype [Object Object] add [Object Function] sub [Object Function] constructor [Object Function] Str [Object Function] a = "G" name "G" prototype [Object Object] _proto_ [Object Object] _proto_ [Object Object] console.log('G: _proto_, G: _proto_'); console.log('G: prototype, G: prototype'); [Object Function] Str [Object Function] name "G" prototype [Object Object] constructor [Object Function] Str [Object Function] a = "G" name "G" prototype [Object Object] _proto_ [Object Object] _proto_ [Object Object] var g = new G(1, 2); // console.log('g: a, g: a'); // "G" undefined // console.log('g: Str("G")', g.constructor); // TypeError: g.constructor is not a function console.log('g: a'); [Object Object] _proto_ [Object Object] constructor [Object Function] Str [Object Function] a = "G" name "G" prototype [Object Object] _proto_ [Object Object] _proto_ [Object Object]</pre>	<pre>function Y(x, y) { this.x = x; this.y = y; } Ya = {}; Y.prototype = function() { return 'Y' + x; }; Y.prototype.add = function() { return this.x + this.y; }; Y.prototype.sub = function() { return this.x - this.y; }; // console.log('Y: Ya, Y: a'); // undefined // console.log('Y: Str("Y")', Y.prototype); // TypeError: Y.prototype is not a function console.log('Y: Y'); [Object Function] Str [Object Function] name "Y" prototype [Object Object] add [Object Function] sub [Object Function] constructor [Object Function] Str [Object Function] a = "Y" name "Y" prototype [Object Object] _proto_ [Object Object] _proto_ [Object Object] console.log('Y: _proto_, Y: _proto_'); console.log('Y: prototype, Y: prototype'); [Object Object] add [Object Function] sub [Object Function] constructor [Object Function] Str [Object Function] a = "Y" name "Y" prototype [Object Object] _proto_ [Object Object] _proto_ [Object Object]</pre>