

Getter and Setter

```
// "Class"
function Point() {
    this.X = 0;
    this.Y = 0;
}

// Define getter
Object.__defineGetter__.call(Point.prototype,
"getCoords", function() {
    return this.X + ", " + this.Y;
});

// Define setter
Object.__defineSetter__.call(Point.prototype,
"setCoords", function(coords) {
    var part = coords.toString().split(", ");
    this.X = part[0] || "";
    this.Y = part[1] || "";
});

// Usage
var testPoint = new Point();
testPoint.setCoords = "10, 20";
document.write(testPoint.getCoords);
```

Defining setter and getter

Inheritance

```
function Animal() {
    this.name = "Animal";
    this.toString = function() {
        return "This is " + this.name;
    };
}

function Canine() {
    this.name = "Canine";
}

function Wolf() {
    this.name = "Wolf";
}

// Inheriting
Canine.prototype = new Animal();
```

Inheritance (cont)

```
Wolf.prototype = new Canine();

// There is need to re-reference constructors
Canine.prototype.constructor = Canine;
Wolf.prototype.constructor = Wolf;

// Function for quick inheriting
function extend(child, parent) {
    var temp = function() {};
    temp.prototype = parent.prototype;
    child.prototype = new temp();
    child.prototype.constructor = child;
}
```

Defining get and set properties

```
function Point() {
    this.X = 0;
    this.Y = 0;
}

Object.defineProperty(Point.prototype, 'pointPos', {
    get: function() {
        return "X: " + this.X + " Y: " + this.Y;
    },
    set: function(point) {
        var parts = point.toString().split(", ");
        this.X = parts[0] || "";
        this.Y = parts[1] || "";
    }
});

// Usage
var testPoint = new Point();
testPoint.pointPos = "10, 20";
document.write(testPoint.pointPos);
```



Define get and set prototype

```
function Circle(radius){
    this._radius = radius;
}
// Define prototype
Circle.prototype = {
    set radius(radius){ this._radius = radius; }
    get radius(){ return this._radius; }
    get area(){ return Math.PI *
Math.pow(this._radius, 2); }
};
// Usage
var circle = new Circle(10);
circle.radius = 5;
document.write(circle.area);
// Usage
```

Call parent method

```
// Call parent method
Wolf.prototype.height = function(){
    var height = Canide.prototype.height.apply(this);
    return height * 1.2;
};
```



By **these**
cheatography.com/these/

Not published yet.
Last updated 8th January, 2017.
Page 2 of 2.

Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish Yours!
<https://apollopod.com>