**Array Literal Syntax**

To avoid potential errors when creating dynamic arrays at runtime, it's much safer to stick with the array literal notation.

**Array Constructor Curiousness**

When you pass a single number to the Array() constructor, it doesn't become the value of the first array element.

[复制代码](javascript:void(0);)

// an array of one element

var a = [3];

console.log(a.length); // 1

console.log(a[0]); // 3

// an array of three elements

var a = new Array(3);

console.log(a.length); // 3

console.log(typeof a[0]); // "undefined"

// using array literal

var a = [3.14];

console.log(a[0]); // 3.14

var a = new Array(3.14); // RangeError: invalid array length

console.log(typeof a); // "undefined"

[复制代码](javascript:void(0);)

**Clever uses of the Array() constructor**

var white = new Array(256).join(' ');

**Check for Array-ness**

[复制代码](javascript:void(0);)

Array.isArray([]); // true

// trying to fool the check with an array-like object

Array.isArray({

length: 1,

"0": 1,

slice: function () {}

}); // false

if (typeof Array.isArray === "undefined") {

Array.isArray = function (arg) {

return Object.prototype.toString.call(arg) === "[object Array]";

};

}

[复制代码](javascript:void(0);)