



Scope

Tips

javascript in one pic

Operators

casting

++*/%

string > number > boolean

number > string > boolean

1'+2

3+'4'

'5'+false

6+true

1'-2

3-'4'

'5'-false

6-true

++a a-- a+=1 a-=1

>, >=, <, <=

==, !=

===, !==

casting

a > b;

// a - b > 0

!true;

// false

! && ||

true && false; // false

false || true; // true

typeof

'function'

properties

f.length;

as value (callback)

var api = function(){

return {

name: 'rainy',

age: 24

};

var handler = function(d){

console.log('Name: '+d.name+', Age: '+d.age);

};

var request = function(api, callback){

callback(api());

};

request(api, handler); // Name: rainy, Age: 24

methods

apply/call/bind

// func.apply(thisObj, [arg1, arg2, ...]);

// func.call(thisObj, arg1, arg2, ...);

// func.bind(thisObj, arg1, arg2, ...);

// about thisObj, see 'Scope'

var sayHi = function(name){

console.log('Hello, ' + name + '!');

};

sayHi.call(this, 'rainy');

// Hello, rainy!

sayHi.apply(this, ['rainy!']);

// Hello, rainy!

sayHi.bind(this, 'rainy')();

// Hello, rainy!

Reference type

Array

typeof

'object'

var a = [1, 2, '3', [4, true]];

var a = new Array(1, 2, '3', [4, true]);

properties

a.length;

// 4

a[0] == 1;

// true

a[3][1] == true;

// true

a.slice(0, 2);

// [1, 2]

a.indexOf(1);

// 0

a.push({});

// return a.length (Mod)

a.pop();

// return popped element (Mod)

a.join('-');

// ?

var s = 'a,b,c,d';

s.split(',');

// ['a', 'b', 'c', 'd']

a.concat(['a', 'b']);

// VS push()

methods

map/reduce

// arr.map(callback, thisObj)

/* callback = function(element, index, arr){

* return element to the same pos of (returned)arr;

* };

*/

[55, 44, 33, 22, 11].map(function(e, i, arr){

return e/(arr.length-i);

});

// [11, 11, 11, 11, 11]

// arr.reduce(callback, init);

/* callback = function(prev, curr, index, arr){

* curr walk through

* arr.slice(init != undefined ? 0 : 1, arr.length);

* prev cache last returned value start with:

* (init != undefined ? init : arr[0]);

* };

*/

[55, 44, 33, 22, 11].reduce(function(p, c, i, arr){

return p + c/(arr.length-i);

}, 55);

// 0;

iteration

for(var i = 0; i < a.length; i++){

console.log(a[i]);

}

a.forEach(function(ele){

console.log(ele);

});

Object (OOP)

root of everything.

Object.prototype

properties

methods

if (cond){

statel;

}else if (cond2){

state2;

}else{

state3;

}

false ? a : b

switch (day) {

case MON:

break;

case TUE:

break;

case WEN:

break;

default:

}

do {

statel;

} while (cond)

while (cond){

statel;

}

break; continue

for (var i = 0; i < len; i++){

statel;

}

for (var k in Obj){

console.log('Obj[' + k + '] = ' + Obj[k]);

}

Flow control

if

if (cond){

statel;

}else if (cond2){

state2;

}else{

state3;

}

false ? a : b

switch

switch (day) {

case MON:

break;

case TUE:

break;

case WEN:

break;

default:

}

while

do {

statel;

} while (cond)

while (cond){

statel;

}

break; continue

for

for (var i = 0; i < len; i++){

statel;

}

for (var k in Obj){

console.log('Obj[' + k + '] = ' + Obj[k]);

}