1. 以下代码执行后，a的值为 B

a = 1;

var a = 2;

function a(){

a = 3;

}

a();

A. 1

B. 2

C. 3

D. function a(){a=3;}

2. 以下代码执行后，a的值为 B

a = 1;

var a = function b(){};

function a(){}

b = 123;

A. 1

B. function b(){}

C. function a(){}

D. 123

3. 以下代码执行后，a的值为 D

a = function(a){

return a = 1;

};

a();

var b = a = function a(){

return a = 2;

};

b();

A. function(a){return a = 1;}

B. 1

C. 2

D. function a(){return a = 2;}

4. 以下代码运行后，a的值为 C

a = function(){

a = 1;

}();

a = function(a){

return 2;

}

(function(){

return a = 3;

}());

A. 1

B. function(a){return a;}

C. 2

D. 3

5. 以下代码运行后，a的值为 A

var a = function(){

var a = 1;

return function(){

b = a++;

};

};

var b;

var c = a();

a()();

c();

A. function(){var a = 1;return function(){b = a++;};}

B. function(){b = a++;};

C. 1

D. 2

6. 以下代码运行后，a的值为 D

var a = (0, 1, function(){

return ++arguments[+[]] || arguments.callee(2);

}(a));

A. 0

B. 1

C. 2

D. 3

7. 以下代码运行后，a的值为 A

var a = new function(){

this.a = A();

return {

a : 1

};

};

function A(){

this.a = 2;

a = 3;

return 4;

}

A. {a:1}

B. 2

C. 3

D. {a:4}

8. 以下代码运行后，a的值为D

var a = "asdf123asd12as1a".match(/a\w+?(?=\d??)/g);

A. asdf1

B. ["asdf1"]

C. ["asdf123", "asd12", "as1"]

D. ["as", "as", "as"]

1. 以下代码运行后，a的值为

var a = {

a : function(b){

console.log(b);

return b = this.b;

},

b : 1

},

b = 2,

c = {

b : 3

};

a = a.a.apply(null, [].slice.call(a.b));

A. null

B. 1

C. 2

D. 3

10. 以下代码运行后，点击b标签后，a的值为

function c(){

return document.createElement(arguments[0]);

}

var div = c("div"),

p = c("p"),

b = c("b"),

body = document.body;

b.appendChild(document.createTextNode("0"));

b.onclick = function(){

a = 1;

};

p.addEventListener("click", function(){

a = 2;

}, 0);

p.appendChild(b);

div.onclick = function(){

a = 3;

};

div.appendChild(p);

body.addEventListener("click", function(){

a = 4;

}, 1);

body.appendChild(div);

A. 1

B. 2

C. 3

D. 4

11. 以下代码运行后，a的值为

var a = 1;

(function(b){

var t = setTimeout(function(){

clearTimeout(t);

b(a++);

}, 500);

})(function(a){

a = ++a;

});

a = 2;

A. 1

B. 2

C. 3

D. 4

12. 以下代码运行后，a的值为

var a = {

a : 2

};

function A(){

this.a = 1;

}

A.prototype = a;

a = new A();

b = new A();

a = (a === b) + (a.\_\_proto\_\_ === b.\_\_proto\_\_);

A. {a:2}

B. 0

C. 1

D. 2

13. 以下代码运行后，a的值为

var a = [

[1, 2],

[3, 4],

[5, 6]

].reduce(function(a, b){

return b.concat(a);

});

A. [[2, 1], [4, 3], [6, 5]]

B. [[[5], [6]], [[3], [4]], [[1], [2]]]

C. [2, 1, 4, 3, 6, 5]

D. [5, 6, 3, 4, 1, 2]

14. 以下代码运行后，a的值为

var a = (0).toFixed(2);

!!(function(a){

if(a >> 1){

arguments.callee(++a);

}

for(;a++ < Math.PI; a = Math.ceil(Math.pow(a, 3))){}

}(a));

A. 0.00

B. 8.00

C. 0.09

D. false

15. 以下请求跨域的是 ABDFG

A. http://www.a.com请求http://www.b.com

B. http://www.a.com请求http://www.a.cn

C. http://www.a.com请求http://www.a.com/a

D. http://www.a.com请求http://www.a.com:2000

E. http://www.a.com:80请求http://www.a.com

F. https://www.a.com请求http://www.a.com

G. http://w.a.com请求http://ww.a.com