1.

(function(){

return typeof arguments;

})();

"object"

"array"

"arguments"

"undefined"

2.

var f = function g(){ return 23; };

typeof g();

"number"

"undefined"

"function"

Error

3.

(function(x){

delete x;

return x;

})(1);

1

null

undefined

Error

4.

var y = 1, x = y = typeof x;

x;

1

"number"

undefined

"undefined"

5.

(function f(f){

return typeof f();

})(function(){ return 1; });

"number"

"undefined"

"function"

Error

6.

var foo = {

bar: function() { return this.baz; },

baz: 1

};

(function(){

return typeof arguments[0]();

})(foo.bar);

"undefined"

"object"

"number"

"function"

7.

var foo = {

bar: function(){ return this.baz; },

baz: 1

}

typeof (f = foo.bar)();

"undefined"

"object"

"number"

"function"

8.

var f = (function f(){ return "1"; }, function g(){ return 2; })();

typeof f;

"string"

"number"

"function"

"undefined"

9.

var x = 1;

if (function f(){}) {

x += typeof f;

}

x;

1

"1function"

"1undefined"

NaN

10.

var x = [typeof x, typeof y][1];

typeof typeof x;

"number"

"string"

"undefined"

"object"

11.

(function(foo){

return typeof foo.bar;

})({ foo: { bar: 1 } });

"undefined"

"object"

"number"

Error

12.

(function f(){

function f(){ return 1; }

return f();

function f(){ return 2; }

})();

1

2

Error (e.g. "Too much recursion")

undefined

13.

function f(){ return f; }

new f() instanceof f;

true

false

14.

with (function(x, undefined){}) length;

* + 1
  + 2
  + undefined
  + Error

Answers:

1-1

2-4

3-1

4-4

5-1

6-1

7-1

8-2

9-3

10-2

11-1

12-2

13-1

14-2