// #1

var test = 'foo';

function testScope() {

      var test = 'baz';

      console.log(test);

}

testScope(); // Виводить спочатку «baz» тому, що виконується присвоєння в функції.

console.log(test); // Виводе «foo» після «baz» тому, що test присвоюється раніше.

// #2

var test = 'foo';

function testScope() {

      test = 'baz';

      console.log(test);

}

console.log(test); // Виводить «foo» бо присвоєно за межами функції

testScope(); // Виводить «baz» бо присвоєно у scope функції

console.log(test); // Виводить «baz» бо присвоєно у scope функції змінило test усюди

// #3

var test;

test = 'foo';

var test = 'baz';

console.log(test); // Виводемо «baz» бо перезаписуємо test з «foo» у «baz».

// #4

function foo() {

  console.log(1);

}

var foo;

foo = function() {

  console.log(2);

};

foo(); // Виводе «2» бо проходить перезапис функції

// #5

var test = 'foo';

function foo() {

    return test;

}

function baz(callback) {

    return callback();

}

baz(foo); // Виводе «foo» бо test проходить через функції foo() і baz(foo);

// #6

var test = 'foo';

function foo() {

    return test;

}

function baz(callback) {

    var test = 'baz';

    return callback();

}

baz(foo); // Виводе «foo» бо var test = 'baz'; не впливає на прийняте значення callback

// #7

var test = 'foo';

function foo() {

    return test;

}

function baz(callback) {

    this.test = 'baz';

    var binded = callback.bind(this);

    return binded();

}

baz(foo); // Виводе «baz» бо this.test не має скоупа звичайного обєкту і прозводить присвоєння значені, а  var binded = callback.bind(this) робить нову функцію.

// #8

var test = 'foo';

var foo = function () {

    return test;

};

var baz = function (callback) {

    var test = 'baz';

    return callback();

};

baz(foo); // Виводе «foo» бо функції baz = function (callback) передається функція foo = function () і не проходить перезапис test.

// #9

var test = 'qwerty';

var a = 4;

function foo() {

    if (a === 4) {

          var test = 'bazinga';

    }

      console.log(test);

}

foo(); // Виводе «bazinga» бо проходить умову а===4 і перезаписвує test.

// #10

var x = 5;

(function () {

    console.log(x);

    var x = 10;

    console.log(x);

})(); // Виведе undefined і 10 бо спочатку х= undefined ,а потім було пере присвоєння х=10.

// #11

(function() {

   var a = b = 5;

})();

console.log(b); // Виведе «5» бо було присвоєння b=5 в середині функції.

// #12

function test() {

   console.log(a); // Виведе «Undifined» бо змінна а ще небула присвоєна функцію.

   console.log(foo()); // Виведе «2» функція провила повернення числа 2

   var a = 1;

   function foo() {

      return 2;

   }

}

test();