1. **Что такое Lexical Environment?** a) Внутренний объект, создающийся при запуске функции и хранящий переменные, созданные при этом запуске.   
    b) Глобальный объект.   
    c) Область видимости функции.
2. **Что выведется на экран при выполнении данного кода?**

var a = 5;   
function showNum() {  
 var a = 10;   
 };   
alert(a);

a) 10

b) 5

c) 15

1. **Как вызвать вложенную функцию в данном коде?**

function sayHi() {

alert(‘Hello there!’);

return function() {

alert(‘Hello world!’)

}

}

a) sayHi();   
 b) sayHi()();   
 c) sayHi()(function(){});

1. **Когда создается объект [[Scope]]?** a) При загрузке страницыb) При создании функции   
    c) При запуске функции
2. **Что выведется на экран при выполнении данного кода?**

var a = 1;

function myFunc1() {

var a = 2;

function myFunc2() {

return function() {

alert(a);

}

}

myFunc2()();

}

myFunc1();

a) 1

b) 2

c) Ошибка

1. **Что выведется на экран при выполнении данного кода?**

var a = 5

(function() {

alert(a)

})()

a) 5   
 b) undefined  
 c) Ошибка

1. **Каким будет значение [[Scope]] для функции myFunc2?**

function myFunc1() {

return function() {

(function(){

function myFunc2() {

alert(‘Hello!’);

}

var a = myFunc2;

a();

}())

}

}

myFunc1()();

a) анонимная функция

b) window

c) myFunc1

1. **Что выведет на экран alert?**

function() {

if(true) {

var a = 5;

}

alert(a);

};

a) 5

b) undefined

c) Ошибку

1. **Что выведет на экран alert?**

var a = 5;

function first() {

a = 6;

}

function second() {

alert(a);

}

first();

second();

a) 5

b) 6

c) undefined

1. **Что выведет на экран alert?**

function first() {

window.a = 3;

}

function second() {

alert(a);

}

first();

second();

a) 0

b) 3

c) undefined

1. **Для чего используются модули через замыкания?**

a) Чтобы скрыть внутренние детали реализации скрипта.

b) Чтобы избежать использования глобальных переменных.

c) Чтобы избежать повторного использования блоков кода.

1. **Выберите вариант кода, который является способом экспорта значения переменной message из модуля:**

(function () {

var message = "Привет";

function showMessage() {

alert( message );

}

showMessage();

... код для экспорта переменной message здесь ...

}())

a) window.message = message;

b) return message;

c) return this.message;

1. **Что выведет на экран следующий код?**

(function () {

var a = 1,

b = 100;

function addNum(a,b) {

window.addNum = function(a,b) {

return a + b;

}

}

addNum();

}());

alert(addNum(5,10));

a) 15

b) 101

c) Ошибка

1. **Какой из предложенных вариантов вызова функции на месте не сработает?**

a) +function() {

alert('Вызов на месте');

}();

b) (function() { alert('Вызов на месте')}());

c) function() { alert('Вызов на месте')}();

1. **Что выведет на экран следующий код?**

var c = 10;

(function () {

var a = 1,

b = 100;

return function() {

var a = 2,

b = 200;

return a + c;

}

}())

a) 11;

b) 12;

c) Ошибка;

1. **Что выведет на экран следующий код?**

var f = function() {

f.prop1 = true;

return function(){

alert(f.prop1)};

}

var f1 = f()();

f1();

a) true;

b) undefined;

c) Ошибка;

1. **В каком порядке выведет числа 6, 7 и 8 на экран следующий код?**

var a = 6;

function test() {

var a = 7;

function again() {

var a = 8;

alert(a); // Первый alert

}

again();

alert(a); // Второй alert

}

test();

alert(a); // Третий alert

a) 7, 6, 8

b) 8, 7, 6

c) 6, 7, 8

1. **Что выведет на экран следующий код?**

function getFunc() {

var a = 7;

return function(b) {

alert(a+b);

}

}

var f = getFunc();

f(5);

a) 5

b) 7

c) 12

d) null

e) undefined

1. **Что выведет на экран следующий код?**

function makeCounter() {

var currentCount = 0;

return {

nextNumber: function() {

return currentCount++;

}

}

prevNumber: function() {

return currentCount--;

}

};

var counter = makeCounter();

counter.getNext(5);

alert(counter.getPrev(1));

a) 4

b) 0

c) Ошибка

1. **Что выведет на экран следующий код?**

(function () {

var a = 100;

return (function() {

var b = 10;

alert(a + b)

}())

}())

a) 110;

b) 100;

c) null;

d) undefined;