Лабораторна робота №2

// #1

var test = 'foo';

function testScope() {

var test = 'baz';

console.log(test);

}

testScope(); // Виводе 'baz' тому що виконується функція.

console.log(test); // Виводе 'foo' тому що 'baz' має свій скоуп.

// #2

var test = 'foo';

function testScope() {

test = 'baz';

console.log(test);

}

console.log(test); // Виводе 'foo' тому що функція ще не виконувалась.

testScope(); // Виводе 'baz' тому що функція виконалась і зміній присвоїли нове значення.

console.log(test); // Виводе 'baz' тому що в функції змінній присвоїли нове значення.

// #3

var test;

test = 'foo';

var test = 'baz';

console.log(test); // Виводе 'baz' тому що ми перезаписали зміну test закинувши в неї нове значення.

// #4

function foo() {

console.log(1);

}

var foo;

foo = function() {

console.log(2);

};

foo(); // Виводе 2 тому що ми перезаписали функцію та надали їй другий зміст.

// #5

var test = 'foo';

function foo() {

return test;

}

function baz(callback) {

return callback();

}

baz(foo); // Виводе 'foo' тому що ми передаємо функцію foo() в функцію baz(callback).

// #6

var test = 'foo';

function foo() {

return test;

}

function baz(callback) {

var test = 'baz';

return callback();

}

baz(foo); // Виводе 'foo' тому що перезапис var test = 'baz' працює в деякій області видимості.

// #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' тому що метод callback.bind(this) робить нову функцию та передає туди this.test = 'baz'.

// #8

var test = 'foo';

var foo = function () {

return test;

};

var baz = function (callback) {

var test = 'baz';

return callback();

};

baz(foo); // Виводе 'foo' тому що в функцію baz = function (callback) передається функція foo = function () і ця функція не перезаписується.

// #9

var test = 'qwerty';

var a = 4;

function foo() {

if (a === 4) {

var test = 'bazinga';

}

console.log(test);

}

foo(); // Виводе 'bazinga' тому що перевірила зміну a === 4 та перезаписала var test = 'bazinga'.

// #10

var x = 5;

(function () {

console.log(x);

var x = 10;

console.log(x);

})(); // Виводе undefined і 10 тому що var x; піднявся вгору в функції а потім х=10 тому відповідь undefined і 10.

// #11

(function() {

var a = b = 5;

})();

console.log(); // Виводе 5 тому що зміна а=b=5 може виконуватись в будь якій області, потім виконується функція і виводиться 5. a локальна зміна а b будь-яка.

// #12

function test() {

console.log(a); // Виводе undefined тому що зміна а поки-що не має значення.

console.log(foo()); // Виводе 2 тому що виконується функція function foo() яка возращає 2.

var a = 1;

function foo() {

return 2;

}

}