Javascript Scope Exercises

1. **Determine what this Javascript code will print out (without running it):**

x = 1;

var a = 5;

var b = 10;

var c = function(a, b, c) {

var x = 10;

document.write(x);**//10,** x is local var x = 10;

document.write(a);**//8** a: argument c(**8**,9,10)

var f = function(a, b, c) {

b = a;

document.write(b);**//8 b = a = 8**

b = c;

var x = 5;

}

f(a,b,c);

document.write(b);**//9** b: argument c(8,**9**,10)

}

c(8,9,10);

document.write(b);**//10** b in var b = 10;

document.write(x);**//1** x in x = 1;

}

**Result:**

10

8

8

9

10

1

1. **What is the difference between a method and function?**

//Slide 37

var name = {

... methodName: function(parameters)

{ statements; }

};

var pt = {

x: 4, y: 3,

distanceFromOrigin: function() {

return Math.sqrt(this.x \* this.x + this.y \* this.y); }

};

alert(pt.distanceFromOrigin()); // 5

//like in Java, objects' methods run "inside" that object

In most respects it is identical to a function except for two key differences:

* A method is implicitly passed the object on which it was called.
* A method is able to operate on data that is contained within the object (class in Java).

1. **What does 'this' refer to when used in a Java method?**

* The ‘this’ keyword refers to that object
* The this keyword is option in Java

1. **What does 'this' refer to when used in a JavaScript method?**

//Slide 37, 38

* The ‘this’ keyword refers to that object
* Global variables/functions you declare become part of it – they use the global object as this when you call them
* Unlike in Java, the this keyword is mandatory in JS

1. **What does 'this' refer to when used in a JavaScript constructor function?**

//Slide 52

The keyword this inside the constructor function points to the newly created object.

function Person(name, age, sex) {

this.name = name;

this.age = age;

this.sex = sex;

}

var george = new Person("George Smith", 33, "M");

var ken = new Person("Ken Jones", 39, "M");

* New operator creates a new object and calls the constructor function to initialize the fields.
* The keyword this inside the constructor function points to the newly created object.

1. **Assume object x is the prototype for object y in Javascript. Object x has a method f( ) containing keyword 'this'. When f is called by x.f( ), what does 'this' refer to?**

In this case, keyword this refers to x

|  |
| --- |
| Refer to slide 51:  var o = {  a: 2,  m: function(b){  return this.a + 1;  }  };  console.log(o.m()); // 3  // When calling o.m in this case, 'this' refers to o  var p = Object.create(o);  // p is an object that inherits from o  p.a = 4; // creates an own property 'a' on p console.log(p.m()); // 5  // when p.m is called, 'this' refers to p.  // So when p inherits the function m of o,  // 'this.a' means p.a, the own property 'a' of p |

1. **What is a free variable in JavaScript?**

free variable: A variable referred to by a function that is not one of its parameters or local variables.

* bound variable: A free variable that is given a fixed value when "closed over" by a function's environment.

1. **Create an object that has properties with name = "fred" and major="music" and a property that is a function that takes 2 numbers and returns the smallest of the two, or the square of the two if they are equal.**

var myObject = {

name: “fred”,

major: “music”,

myFunc: function(a,b){

if(a===b)

return a\*a\*b\*b;

return Math.mix(a,b);

}

};

myObject;// {name: "fred", major: "music", myFunc: ƒ}

myObject.myFunc(1,2);//1

myObject.myFunc(2,2);//2\*2\*2\*2 = 16

1. **Write Javascript code for creating three Employee objects using the "new" keyword and a constructor function. Employee objects have the following fields: name, salary, position.**

function Employee(name,salary,position){

this.name = name;

this.salary=salary;

this.position = position;

}

var emp1 = new Employee(“Quy”,10000, “Java Full-stack Developer”);

var emp2 = new Employee(“Huy”,9000, “Back-end Developer”);

var emp3 = new Employee(“Phu”,8000, “Front-End Developer”);

1. **Write a Javascript function that takes any number of input arguments and returns the product of the arguments.**

function myFunc(){

let pro = 1;

for(let i=0;i<arguments.length;i++)

pro=pro\*arguments[i];

return pro;

}

myFunc(2);

myFunc(2,4);

myFunc(2,4,6);

1. **Write an arrow function that returns the maximum of its three input arguments.**

var f = (a,b,c) =>{

return Math.max(a,Math.max(b,c));

};

f(1,2,3);//3

f(4,5,6);//6

f(1,1,2);//2