1. What are the ways to create the objects

We can define and create a single object, using an object literal, with the keyword new and by using object constructor, and then create objects of the constructed type.

Ex:

//Single Object

var car = {company:"Maruthi", model:"800"};

//New Keyword

var car = new Object();

//Object Constructor

function car(company, model) {

this.company = first;

this.model = last;

}

2. How many ways we can create the arrays.

Using an array literal and new Keyword

Ex:

var j2ee = ["servlets", "jsp", "jsf"];

var j2ee = new Array("servlets", "jsp", "jsf");

3. What are arguments in javascript functions.

Arguments in functions are the values passed to the function.

Ex:

function sum(3,5) {

return 3 + 5;

}

4. What is prototypal inheritance in javascript?

A prototype is a property of functions and of objects that are created by constructor functions. Object is the prototype of a function. Its main use is when a function is used as a constructor.

Ex:

function Vehicle(company, type) {

this.company = company;

this.type = type; }

var bike = new Vehicle("Maruthi", "car");

var car = new Vehicle("Yamaha", "bike");

5. What are enumerators in javascript?

An enumerator is an object used to iterate through a large collection of data. Iterating through a large collection with the enumerator object is faster than any other way in javascript. It allows using a walk- through mechanism to iterate through the items.

Ex:

var Clothes = {

SHIRT: 1,

PANT: 2,

TSHIRT: 3,

FLIPONS:4,

properties: {

1: {name: "shirt", value: 1, code: "Sh"},

2: {name: "pant", value: 2, code: "P"},

3: {name: "tshirt", value: 3, code: "TS"},

4: {name: "flipons", value: 3, code: "FP"}

}

};

6. Callbacks and closures.

A callback function, also known as a higher-order function, is a function that is passed to another function as a parameter, and the callback function is called (or executed) inside the other function.

Ex:

Function curry(param1, param2, callback) {

alert(‘Started making curry \n’ + param1 + ‘, ‘ + param2);

callback();

}

curry(‘’,’’,function()) {

alert(‘Finished Curry’);

});

A closure is an inner function that has access to the outer function’s variables. The closure has access to its own, it has access to the outer function’s variables, and it has access to the global variables.

Ex:

function sum() {

var a = 5;

var b = 6;

document.getElementById("demo").innerHTML = a + b;

}

7.Module based programming in javascript.

Modules divide programs into clusters of code that, by some criterion, belong together.

Ex:

var symbols = [“foo”, “bar”];

function foo() {

return “foo”;

}

var bar = {

name: “bar”,

size:3 };

var dummy = “dummy”;

8.What is strict mode in javascript

The "use strict" directive is a literal expression, ignored by earlier versions of JavaScript. "use strict" is to indicate that the code should be executed in "strict mode". With strict mode, you cannot, use undeclared variable.

Ex:

"use strict";

math();

function math() {

y = 5;

}

9.what is the difference between == and ===?

== is just comparing the two values, and if they are of different types, type conversion is done.

=== compares the values and well as their types - so no type conversion will be done here.

Ex:

var a = [1,2];

var b = [1,2];

console.log(a == b);

console.log(a === b);

10.Ternary operator.

The conditional (ternary) operator is the only JavaScript operator that takes three operands. This operator is frequently used as a shortcut for the if statement.

condition ? expr1 : expr2;

Ex:

access = firstCheck ? "Access denied" : secondCheck ? "Access

11.Difference between public, private and static variables and their use cases.

A variable which can be used outside a function, becomes GLOBAL. A global variable has global scope.

Private members are made by the constructor. Ordinary vars and parameters of the constructor becomes the private members.

A function is an object. We can create static variables which allow to put a static keyword before a variable, and then such variable is not cleared in next calls.