1. **Different ways to create objects in javascript?**

Ans: JavaScript Variables can contain single values. Objects are variables too.

*Using Object Literal:*

var customer = {firstName: “Naveen” , lastName: “Kurakula” , salary: 5000 , company: “EkthaSolutions”};

*Using JavaScript new keyword:*

var customer = new Object();

customer.firstName = “Naveen”;

customer.lastName = “Kurakula”;

customer.salary = 6000;

customer.company = “Ektha”;

*Using Object Constructor:*

function customer(firstName, lastName, salary, company) {  
    this.firstName = firstName;  
    this.lastName = lastName;  
    this.salary = salary;  
    this. company = company;  
}  
var bofaEmp = new customer ("Yeshes", "Meka", 10000, "BankOfAmerica");  
var citiEmp = new customer ("Rahul", "Kaur", 12000, "CitiBank");

All Js objects are mutable(i.e will not create a copy of object, addressed by reference) while variables are immutable.

1. **How many ways we can create the arrays?**

Ans: *Using literal way*

var points = new Array();  //Not preferred way to create an array this might lead to probl

*Using JavaScript new keyword*

var points = [];

1. **What are arguments in javascript functions?**

Ans: Function parameters are real values passed to the function during the function call.

*ParameterDefaults:*

If the parameter are less than the declared in the function, then the parameters will be set to ‘undefined’.

If the parameters are more than declared in the function they can be accessed through in built ‘*arguments*’ object defined for javascript.

1. **What is prototypal inheritance in javascript?**

Ans: By default, every function has prototype property and it is null.  Its main purpose is to allow multiple instances of an object to share a common property. Thus, object properties which are defined using the prototype object are inherited by all instances which reference it.

1. **What are enumerators in javascript.**

Ans: An enumerated type is a data type consisting of a set of named values called elements, members or enumerators of the type. The enumerator names are usually identifiers that behave as constants in the language.

Example:

var SizeEnum = {

SMALL: 1,

MEDIUM: 2,

LARGE: 3,

properties: {

1: {name: "small", value: 1, code: "S"},

2: {name: "medium", value: 2, code: "M"},

3: {name: "large", value: 3, code: "L"}

}

};

Then use it like so:

var mySize = SizeEnum.MEDIUM;

var myCode = SizeEnum.properties[mySize].code;

1. **Callbacks and closures.**

**Ans**: A callback function, also known as a higher-order function, is a function that is passed to another function (let’s call this other function “otherFunction”) as a parameter, and the callback function is called (or executed) inside the otherFunction.

Example: $("#btn\_1").click(function() {

alert("Btn 1 Clicked");

});

Principles to implement Callback functions:

* Use Named OR Anonymous Functions as Callbacks.
* Pass Parameters to Callback Functions.
* Make Sure Callback is a Function Before Executing It.

function getInput(options, callback) {

allUserData.push(options);

​

// Make sure the callback is a function​

if (typeof callback === "function") {

// Call it, since we have confirmed it is callable​

callback(options);

}

}

Without the check in place, if the getInput function is called either without the callback function as a parameter or in place of a function a non-function is passed, our code will result in a runtime error.

1. **Module based programming in javascript.**

Ans: *Modules* divide programs into clusters of code that, by *some* criterion, belong together. If we do not use module based approach which isn’t structured as a set of modules, it is not apparent which parts of the code are needed to use a particular function.

1. **What is strict mode in javascript**

Ans: Strict mode converts previously used bad syntax into real errors.

* Using a variable, without declaring it, is not allowed.
* Using an object, without declaring it, is not allowed.
* Deleting a function is not allowed.
* Duplicating a parameter name is not allowed
* Escape characters are not allowed.

1. **What is the difference between == and ===**

Ans: If x=5(given)

Then, x==5 returns true

X==”5” returns true

X === 5 returns true

X===”5” returns false

From this we can understand that === is strict comparison for which both the operand type and content have to match.

Whereas, == is abstract comparison for which operand type is converted to same type and then the comparison is done.

1. **Double equals checks the value where as triple equals checks the value and datatype.**

Ans Explained above.

1. **Ternary operator.**

Ans: It contains a condition where in it assigns a values based on the result of the condition.

var grade = (score>70)? “Pass”: “Fail”

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

Ans: A static member is shared by all instances of the class as well as the class itself but it is only stored in one place. By declaring a variable within a function, it is only available from within there. A privileged method is a method having access to private properties, but at the same time publicly exposing itself. Public variables are those which are in global scope and available for all the functions.