1. Write a blog on Difference between HTTP1.1 vs HTTP2

**Ans:**

**HTTP**

HTTP stands for hypertext transfer protocol, and it is the basis for almost all web applications. More specifically, HTTP is the method computers and servers use to request and send information.

**HTTP 1.1:**

* The first usable version of HTTP was created in 1997. Because it went through several stages of development, this first version of HTTP was called HTTP 1.1. This version is still in use on the web.
* HTTP 1.1 is slower when compared to HTTP 2.
* HTTP 1.1 loads resources one after the other, so if one resource cannot be loaded, it blocks all the other resources behind it.
* HTTP 1.1 serves content to a client device if the client asks for it. However, this approach is not always practical for modern webpages, which often involve several dozen separate resources that the client must request.
* It works on the textual format.
* It compresses data by itself.

**HTTP 2:**

* In 2015, a new version of HTTP called HTTP 2 was created.
* HTTP 2 is much faster and more efficient than HTTP 1.1.
* HTTP 2 is able to use a single TCP connection to send multiple streams of data at once so that no one resource blocks any other resource. HTTP/2 does this by splitting data into binary-code messages and numbering these messages so that the client knows which stream each binary message belongs to.
* HTTP 2 allows a server to "push" content to a client before the client asks for it. The server also sends a message letting the client know what pushed content to expect.
* It works on the textual format.
* It compresses data by itself.

1. Write a blog about objects and its internal representation in Javascript.

**Ans:**

Objects, in JavaScript, is it’s most important data-type and forms the building blocks for modern JavaScript. These objects are quite different from JavaScript’s primitive data-types (Number, String, Boolean, null, undefined and symbol) in the sense that while these primitive data-types all store a single value each (depending on their types).

Objects in JavaScript may be defined as an unordered collection of related data, of primitive or reference types, in the form of “key: value” pairs. These keys can be variables or functions and are called properties and methods, respectively, in the context of an object. For Eg. If student is an object, it will have properties like name, age, address, id, etc.

**Objects and Properties:**

A JavaScript object has properties associated with it. A property of an object can be explained as a variable that is attached to the object. Object properties are basically the same as ordinary JavaScript variables, except for the attachment to objects. The properties of an object define the characteristics of the object. We canaccess the properties of an object with a simple dot-notation: objectName.propertyName

Like all JavaScript variables, both the object name (which could be a normal variable) and property name are case sensitive. We can define a property by assigning it a value.

Eg:

var student = new Object();

student.name = “Kavitha”;

student.gender = “Female”;

student.age = 10;

**Creating Objects in JavaScript:**

**Create Javascript Object with Object Literal:**

One of easiest way to create a javascript object is object literal, simply define the property and values inside curly braces.

Eg: let student = {name: “Kavitha”, gender: “Female”, age: 10}

**Create JavaScript Object with Constructor:**

Constructor is nothing but a function and with help of new keyword, constructor function allows to create multiple objects of same flavor.

Eg:

function student(name, age){

this.name = name;

this.age = age;}

let first = new student(“Kavitha”, 10);

let second = new student(“Kavin”, 10);

console.log(first.name) //Output: Kavitha

console.log(second.name) //Output: Kavin

**Create JavaScript Object Using the object.create() Method:**

Objects can also be created using the object.create() method. This method can be very useful, because it allows the user to choose the prototype object for the object the user want to create, without having to define a constructor function.

Eg:

// Animal properties and method encapsulation

var Animal = {

type: 'Invertebrates', // Default value of properties

displayType: function() { // Method which will display type of Animal

console.log(this.type);

}

};

// Create new animal type called animal1

var animal1 = Object.create(Animal);

animal1.displayType(); // Output:Invertebrates

// Create new animal type called Fishes

var fish = Object.create(Animal);

fish.type = 'Fishes';

fish.displayType();

// Output:Fishes