Day – 1

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

|  |  |
| --- | --- |
| **HTTP 1.1** | **HTTP 2.0** |
| * Sends request in a text based format and receives a HTML page as response | * Sends request in a binary format and receives a HTML page as response |
| * When client makes a request to the server TCP connection is established and reused when the second request is raised. Consequent requests would be send only after the response for previous request is received. | * When client makes a request to the server TCP connection is established along with streams, which allows the user to make multiple requests at the same time. |
| * Header Repetition . For every requests raised header will be carried along with response. | * HPACK is used where the header and the requested data is separated. HPACK automatically caches the header the consequent requests. |
| * Data Compression is done by itself. | * Data Compression and Header Compression is done by HPACK. |
| * It uses request resource pipelining to get multiple responses. | * Uses PUSH framework to get all the files required for one request. |

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

Object:

An Object is an standalone entity with its properties and values. It is also known Key:Value pairs, which means every variable in a object will have value assigned for it.

Eg:

Let Student ={

Name: “Pradeepa”,

Department : “Physics”

College : “LDC”

}

Here Student is a object created along with keys Name, Department and College which has value as Pradeepa, Physics and LDC respectively.

To get the details of the object use:

Console.log(Student);

Which inturn returns the key value pairs as [Name:”Pradeepa”,Department: “Physics”,College:”LDC”]

To access the particular property in a object we should use the dot operator. If the user wants to get the name of a student then the user can use:

Console.log(Student.Name);

Where the output would be Pradeepa.

In some cases the properties of objects are not kown. At those times . operators wont work and we use [] instead of them.

Eg:

Let Student ={

Name: “Pradeepa”,

Department : “Physics”

College : “LDC”

}

Function getobject(O,P){

Console.log(O,P);

Let result = O[P];

Console.log(result);

}

getobject(Student,Department);

Here when the function getobject is called the object Student gets stored in the Paramenter O and the property Department gets stored in the paramaeter P. The variable result fetches the exact data that is student’s department which is Physics. If insteadof [] we use dot operator then the function will search for the property P in the object which do not exists and gives the output as undefined.

Creation of Array of Objects is also possible in Javascript.

Eg:

Let Student =[{

Name: “Pradeepa”,

Department : “Physics”

College : “LDC”

}

{

Name: “Bharathi”,

Department : “Computer Science”

College : “LDC”

}

{

Name: “Priya”,

Department : “Maths”

College : “LDC”

}];

Here we have created a array consisting of 3 objects. To access the array we use indexing. To access the properties of the objects in a array we use indexing along with dot operator.

Console.log(Student[2.Name]);------ will give output as Bharathi.

Similarly sub objects can also be created.

Let Student ={

Name: “Pradeepa”,

Department : “Physics”

College : “LDC”

Address: {

Home Address: { No:2,Area:KVP};

College Address:{No:3,Area:Narimedu};

}

Student is a object which has Address as sub object which in turn had Home Address and College Address as Sub Object.

To print the area of the of the college

Console.log(Student.Address.CollegeAddress.Area);-------Output-------------------Narimedu.