**DAY 1 TASK**

1.WRITE A BLOG ON DIFFERENCE BETWEEN HTTP 1.1 VS HTTP 2.

|  |  |
| --- | --- |
| HTTP1.1 | HTTP2 |
| * 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/2 is able to use a single [TCP](https://www.cloudflare.com/learning/ddos/glossary/tcp-ip/) connection to send multiple streams of data at once so that no one resource blocks any other resource |
| * It uses request resource inlining for use getting multiple pages. | * It uses **PUSH** frame by server that collects all multiple pages. |
| * It compresses data by itself. | * It uses **HPACK** for data compression. |
| * It works on the textual formate. | * It works on the binary protocol. |
| * This first version of HTTP was called HTTP/1.1. | * a new version of HTTP called HTTP/2 was created. |
| * It went through several stages of development. | * HTTP/2 is much faster and more efficient than HTTP/1.1. |
| * In HTTP/1.1, flow control relies on the underlying TCP connection. | * HTTP/2 multiplexes streams of data within a single TCP connection |

2.WRITE A BLOG ABOUT OBJECTS AND IT’S INTERNAL REPRESENTATION IN JAVA SCRIPTS.

1. 2. Write a blog about objects and its internal representation in 2.WRITE A BLOG ABOUT OBJECTS AND IT’S INTERNAL REPRESENTATION IN JAVA SCRIPTS.

OBJECTS:

* 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)
* Objects are more complex and each object may contain any combination of these primitive data-types as well as reference data-types.object, is a reference data type.
* The variables don’t actually store the value.
* Loosely speaking, 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 your object is a student, it will have properties like name, age, address, id, etc and methods like updateAddress, updateNam, 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. You access the properties of an object with a simple dot-notation:

 For example:

let’s create an object named myCar and give it properties named make, model, and year as follows:

var myCar = new Object();  
myCar.make = 'Ford';  
myCar.model = 'Mustang';  
myCar.year = 1969;

**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 as shown below

let bike = {name: 'SuperSport', maker:'Ducati', engine:'937cc'};

# 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 as shown below

function Vehicle(name, maker) {  
 this.name = name;  
 this.maker = maker;  
}  
let car1 = new Vehicle(’Fiesta’, 'Ford’);  
let car2 = new Vehicle(’Santa Fe’, 'Hyundai’)  
console.log(car1.name); //Output: Fiesta  
console.log(car2.name); //Output: Santa Fe