**Road Map Day 1 Task**

1. ***Difference between HTTP1 vs HTTP2?***

* HTTP stands for hypertext transfer protocol & it is used in client-server communication.
* By using HTTP, user sends the request to the server & the server sends the response to the user.

| **HTTP1** | **HTTP2** |
| --- | --- |
| HTTP/1 was known to have poor response time. With websites becoming more resource-intensive, the protocol was losing its efficiency. It progressively became essential to minimise latency and boost page load speeds. | The core features of HTTP/2 provide greater levels of control that can be used to optimise the web application performance. |
| The client (browser) has to send a request to the server using the method (GET/POST). | HTTP/2 developers introduced a binary framing layer. |
| Server responds with the requested resource, for example – image, alongside the status of what it did to the client’s request. | This layer partitions requests and responses in tiny data packets and encodes them. |
| If a request at the queue head cannot retrieve its required resources, it can block all requests behind it. | Due to this, multiple requests and responses become able to run parallelly with HTTP/2 |
|  |  |

1. ***Objects and its internal representation in Javascript:***

* A JavaScript object is a collection of named values having state and behaviour (properties and method).
* For example: Person, car, pen, bike, Personal Computer , Washing Machine etc.
* Take the case of cars.
* All cars have the same properties, but the property values differ from car to car. All cars have the same methods, but the methods are performed at different times.
* Let’s have an example of my favourite merc car and list out its properties(Features):

Make: Mercedes

Model: C-Class

Colour: White

Fuel: Diesel

Weight: 850kg

Mileage: 8 Kmpl

Rating: 4.5

1. **Objects:**

* The following code assigns a simple value (Mercedes) to a variable named car:

**var car = "Mercedes";**

* Objects are variables too. But objects can contain many values.
* The following code assigns many values (Mercedes, C-class, White and so on) to a variable named Car:
* var car = {Make: “Mercedes”, Model: “C-Class”, Color: “White”, Fuel: Diesel, Weight: “850kg”, Mileage: “8 Kmpl”, Rating: 4.5};

1. **Object Properties**

* The name:value pairs (in JavaScript objects) are called properties.

**var car = {Make: “Mercedes”, Model: “C-Class”, Color: “White”, Fuel: Diesel, Weight: “850kg”,Mileage: “8 Kmpl”, Rating: 4.5};**

* From the above snippet, let’s have a look what falls under property and property value:
* 
* The object properties can be different primitive values, other objects and functions.
* Properties can usually be changed, added, and deleted, but some are read only.
* The syntax for adding a property to an object is :

**ObjectName.ObjectProperty = propertyValue;**

1. **Object Method:**

* An object method is an object property containing a function definition.
* Let’s assume that to start the car there will be mechanical functionality.

**function()**

**{**

**return ignition.on**

**}**

* and so similar is to stop/brake/headlights on & off, etc.

**Internal representation of JavaScript:**

* When you type a web address into your browser. For eg: [www.guvi.in/](http://www.guvi.in/)
* The browser goes to the DNS server, and finds the real address of the server that the website lives on.
* The browser sends an HTTP request message to the server, asking it to send a copy of the website to the client.
* This message, and all other data sent between the client and the server, is sent across your internet connection using TCP/IP.
* If the server approves the client's request, the server sends the client a "200 OK" message, which means "Of course you can look at that website! Here it is", and then starts sending the website's files to the browser as a series of small chunks called data packets.
* The browser assembles the small chunks into a complete web page and displays it to you.

**Road Map Day 2 Task**

***1, Difference between JS Document and JS window objects:***

| **JS Document** | **JS Window Objects** |
| --- | --- |
| It represents the document loaded inside the window or browser. | It represents the browser window in which you are seeing the content. |
| The properties related to it are stored in the document object. | The properties related to it are stored in the window object. |
| It is loaded after the loading window because the window contains a document. | It is loaded before the document because of the window container document. |
| It is the root element of the document object model. | The window is the global element for all objects, functions, etc. |
| It is an object of the window. | It is an object of the browser. |
| We can not access windows objects properties inside the document. | We can access document object properties inside the window. |
| logically:  document:{ properties} | logically:  window:{  document:{properties}  } |
| Example: document.title will return the title of the document | Example: window.document.title will return the title of the document. |