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

* HTTP2 is much faster and more reliable than HTTP1.
* HTTP1 loads a single request for every TCP connection, while HTTP2 avoids network delay by using multiplexing.
* HTTP2 is binary, instead of textual
* HTP2 uses header compression to reduce overhead
* HTTP2 allows servers to “push” responses proactively into client caches

Example for HTTP1.1 vs HTTP2

* In HTTP/2 : When you’re typing something in a Google Doc, as I am typing this article right now, every keystroke sends data to Google’s servers. Google’s servers process that data, and then send updates back to your browser with the text you’ve typed, along with other helpful information like suggestions, the last edit status of the document, and much more.
* In HTTP/1.1: Each of your keypresses would initiate a new connection to the server, to send each character you typed over the wire. Then, your browser would have to constantly “ping” Google’s server to see if the status of the document changed, to add the character to the screen you’re looking at.

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

* Objects are quite different from JavaScript’s primitive data-types(Number, String, Boolean) since these primitive data-types all store a single value each (depending on their types).
* Where an object, is a reference data type and may contain any combination of primitive data-types.
* It may be defined as an unordered collection of related data, of primitive or reference types, in the form of “key: value” pairs.

Syntax : {key : ”value”}

For eg. If your object is a student, it will have properties like name, subject. It will be declared as,

Let student = {

Name : “Elavarasi”,

Subject : “Eng”

}

Where, Name and Subject are key, “Elavarasi” and”Eng” are value.

* To access the student details, the syntax is (object name) : For eg: console.log(student)

To access any details from the object we use

* Dot notation – syntax (object name.key name) : For eg: console.log(student.name)
* Bracket notation –syntax(oject name[“key name”])

We can also add any additional details in the object by creating the sub-object.

|  |
| --- |
| ***EXAMPLE:***  Let student ={  Name : “Elavarasi”,  Subject : “Eng”  Sibilings : {  Brother : raj,  Sister : devi  }}  Console.log(student.sibilings) |

We can access the particular detail in the array of the object.

|  |
| --- |
| **Example :**  let school = {  name: "Vivekananda School",  location : "Delhi",  established : "1971",  attendance :["mon","tue", "wed"]  }  console.log(school.attendance[1]);  **Output:**  Tue |

Push() – to add any data into the object.

|  |
| --- |
| **Example :**  let school = {  name: "Vivekananda School",  location : "Delhi",  established : "1971",  attendance :["mon","tue","wed"]  }  school.attendance.push("thru");  console.log(school.attendance)  **Output:**  [ 'mon', 'tue', 'wed', 'thru' ] |

Splice() – to add any details inbetween without deleting any other details.

Syntax :splice(start, delete count,item1,item2…);

|  |
| --- |
| **Example:** let school = {  name: "Vivekananda School",  location : "Delhi",  established : "1971",  attendance :["mon","wed"]  }  school.attendance.splice(1,0,"tue");  console.log(school.attendance)  **Output :**  [ 'mon', 'tue', 'wed' ] |

Note : If you don’t need to delete anything then use 0 in delete count place.Delete count is optional.

To add details before the data:

|  |
| --- |
| **Example:**  let school = {  name: "Vivekananda School",  location : "Delhi",  established : "1971",  attendance :["mon","wed"]  }  school.attendance.splice(0,0,"sun");  console.log(school.attendance)  **Output :**  [ 'sun', 'mon', 'wed' ] |

Delete the detail and add other detail instead of that:

|  |
| --- |
| **Example:**  let school = {  name: "Vivekananda School",  location : "Delhi",  established : "1971",  attendance :["mon","wed"]  }  school.attendance.splice(0,1,"tue");  console.log(school.attendance)  **Output:**  [ 'tue', 'wed' ] |