Day 1 – Task1

Question :1

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

|  |  |
| --- | --- |
| **HTTP 1.1**  * Loads web content one thing at a time.      * Sends lots of unnecessary data back and forth. * Doesn't tell the browser which parts of a page are most important. * Can get congested on the internet. * Wastes space with repeated information. * Uses multiple connections for different things. * Works with or without encryption. * Might be slower on slow networks. * Must download the entire file even if we just need a part. * Longer waiting time for web pages to load. | **HTTP 2**  * Loads many things at once, making web pages faster. * Sends less data, making communication quicker. * Let’s websites say which parts to load first for a better experience. * Controls data flow to avoid congestion. * Saves space by being more efficient. * Uses a single connection for multiple tasks. * Encourages encryption for better security. * Works well on slow networks. * Allows requesting only the needed part of a big file. * Reduces waiting time for web pages to appear. |

Question 2:

Write a blog about objects and its internal representation in JavaScript.

**Objects and Internal representation in JavaScript:**

* Objects are like containers that can hold different types of information as key-value pairs.
* Information inside objects is stored as properties, which are labelled with names (keys).
* Objects can also contain functions, called methods, which perform actions related to the data.
* Objects are used to structure data and code in a convenient way.

Example:

Var car = {

Colour: red,

Mileage:15,

Type: petrol

}

* Each object has a link to another object called its prototype. Protypes are as a blueprints or guides for objects. This prototype serves as a reference for additional properties and methods that an object can use.
* When user want something from an object, JavaScript first checks if it's in the object itself. If it's not there, it looks in the object's prototype. This process can continue through a chain of prototypes until the item is found or until it reaches the end of the chain.
* Objects can borrow properties and methods from their prototypes, which simplifies code and allow users to create relationships between objects easily.

.