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

|  |  |
| --- | --- |
| Http1.1 | Http2 |
| * The first usable version of HTTP was created in 1997. Because it went through several stages of development, this first version of HTTP was called HTTP/1.1. This version is still in use on the web. * HTTP/1.1 loads resources one after the other, so if one resource cannot be loaded, it blocks all the other resources behind it. * A server only serves content to a client device if the client asks for it. However, this approach is not always practical for modern webpages, which often involve several dozen separate resources that the client must request. * Small files load more quickly than large ones. To speed up web performance, both HTTP/1.1 and HTTP/2 compress HTTP messages to make them smaller. | * In 2015, a new version of HTTP called [HTTP/2](https://www.cloudflare.com/website-optimization/http2/what-is-http2/) was created. HTTP/2 solves several problems that the creators of HTTP/1.1 did not anticipate. * HTTP/2 can 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. HTTP/2 does this by splitting data into binary-code messages and numbering these messages so that the client knows which stream each binary message belongs to. * HTTP/2 solves this problem by allowing a server to "push" content to a client before the client asks for it. The server also sends a message letting the client know what pushed content to expect * HTTP/2 uses a more advanced compression method called HPACK that eliminates redundant information in HTTP header packets. Given the volume of HTTP packets involved in loading even a single webpage, those bytes add up quickly, resulting in faster loading. |

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

**Objects And Its Internal Representation in JavaScript**

* Objects, in JavaScript, is its most important datatype and forms the building blocks for modern JavaScript.
* These objects are quite different from JavaScript’s primitive datatypes (Number, String, Boolean, null, undefined and symbol) in the sense that while these primitive datatypes all store a single value each (depending on their types).
* Objects are more complex, and each object may contain any combination of these primitive datatypes as well as reference datatypes.
* An object is a reference data type. Variables that are assigned a reference value are given a reference or a pointer to that value. That reference or pointer points to the location in memory where the object is stored. The variables don’t 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 E.g. 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:

objectName.propertyName