HTTP1.1 AND HTTP2

* HTTP/2 employs the binary framing layer to wrap all messages in binary format while still keeping HTTP semantics, such as verbs, methods, and headers, in contrast to HTTP/1.1, which preserves all requests and responses in plain text format.
* 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.

CACHING

* HTTP1.1 Expands on the caching support by using additional headers like cache-control, conditional headers like If-Match and by using entity tags.
* HTTP/2 does not change much in terms of caching. With the server push feature if the client finds the resources are already present in the cache, it can cancel the pushed stream.

WEB TRAFFIC

* HTTP/1.1 provides faster delivery of web pages and reduces web traffic as compared to HTTP/1.0. However, TCP starts slowly and with domain sharding (resources can be downloaded simultaneously by using multiple domains), connection reuse and pipelining, there is an increased risk of network congestion.
* HTTP/2 utilizes multiplexing and server push to effectively reduce the page load time by a greater margin along with being less sensitive to network delays.

# Objects And Its Internal Representation In JavaScript

* Its most significant data type serves as the foundation for contemporary JavaScript. While the primitive data types in JavaScript (Number, String, Boolean, null, undefined, and symbol) all store a single value apiece, these objects store multiple values (depending on their types).
* each object may contain any combination of these primitive data-types as well as reference data-types.
* 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 actually store the value.
* For Eg. If your object is a student, it will have properties like name, age, address, id, etc and methods like updateAddress, updateNam, etc.
* A property of an object can be explained as a variable that is attached to the object