Statically typed language:

Statically typed programming languages do type checking (i.e. the process of verifying and enforcing the constraints of types) at *compile-time* as opposed to *run-time*.

Examples of statically typed languages are: Java, C, C++

Dynamically typed language:

Dynamically typed programming languages do type checking at *run-time* as opposed to *compile-time*.

Examples of dynamically typed languages are: Perl, Ruby, Python, PHP, JavaScript

Scripted languages:

Scripting languages are interpreted within another program. JavaScript is embedded within a browser and interpreted by that browser.

Examples of scripting languages

1. JavaScript
2. Perl
3. Python

Advantages of Scripting languages:

1. Simple – Scripting languages are easier to write than a programming language.
2. Fewer Lines of Code (LOC)

Programmed languages:

Programming languages like Java are compiled and not interpreted by another application in the same way.

Examples of programming languages:

1. C
2. C++ and
3. Java

HTTP1.1

Year: 1997

Key Feature:

It supports connection reuse i.e. for every TCP connection there could be multiple requests and responses, and pipelining where the client can request several resources from the server at once. However, pipelining was hard to implement due to issues such as head-of-line blocking and was not a feasible solution.

Status Code:

Introduces a warning header field to carry additional information about the status of a message. Can define 24 status codes, error reporting is quicker and more efficient.

Authentication Mechanism:

It is relatively secure since it uses digest authentication and NTLM authentication.

Caching:

Expands on the caching support by using additional headers like cache-control, conditional headers like If-Match, and entity tags.

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

Year: 2015

Key Feature:

Uses multiplexing, where over a single TCP connection resources to be delivered are interleaved and arrive at the client almost at the same time. It is done using streams that can be prioritized and can have dependencies and individual flow control. It also provides a feature called server push that allows the server to send data that the client will need but has not yet requested.

Status Code:

The underlying semantics of HTTP such as headers, and status codes remain the same.

Authentication Mechanism:

Security concerns from previous versions will continue to be seen in HTTP/2. However, it is better equipped to deal with them due to new TLS features like connection error of type Inadequate_Security

Caching:

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/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

Objects, in JavaScript, are its most important data type and form the building blocks for modern JavaScript. These objects are quite different from JavaScript's primitive data-types(Number, String, Boolean, null, undefined, and symbol) in the sense that while these primitive data-types all store a single value each (depending on their types).

Objects are more complex and 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 the memory where the object is stored. The variables don't actually 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 Eg. If your object is a student, it will have properties like name, age, address, id, etc and methods like updateAddress, updateNam, etc.