|  |  |
| --- | --- |
| HTTP1.1 | HTTP2 |
| 1.Developed by Timothy Berners-Lee in 1989 as a communication standard for the World Wide Web. | 1.HTTP/2 began as the SPDY protocol, developed primarily at Google with the intention of reducing web page load latency by using techniques such as compression, multiplexing, and prioritization. |
| 2.HTTP is a top-level application protocol that exchanges information between a client computer and a local or remote web server. | 2.This protocol served as a template for HTTP/2 when the Hypertext Transfer Protocol working group httpbis of the IETF (Internet Engineering Task Force) put the standard together, culminating in the publication of HTTP/2 in May 2015. |
| 3.In this process, a client sends a text-based request to a server by calling a method like GET or POST. In response, the server sends a resource like an HTML page back to the client. | 3.From the beginning, many browsers supported this standardization effort, including Chrome, Opera, Internet Explorer, and Safari. |
| 4.It works on the textual format. | 4.Due in part to this browser support, there has been a significant adoption rate of the protocol since 2015, with especially high rates among new sites. |
| 5.There is head of line blocking that blocks all the requests behind it until it doesn’t get its all resources. | 5.In HTTP/2, as we know, it sends multiple concurrent responses to the client’s requests. |
| 6.It uses requests resource Inlining for use getting multiple pages | 6. The target web server can send additional resources along with the HTML document without the client’s request for the resource, and this process is called server push. |
| 7..It compresses data by itself. | 7.HTTP/2 even maintains the separation between pushed resources and HTML documents by using this technique, which resolves the issue of resource inlining as the client can either cache the resource or decline the pushed resources from the HTML document. |
| 8.For example, let’s say you are visiting a website at the domain www.example.com. When you navigate to this URL, the web browser on your computer sends an HTTP request in the form of a text-based message, similar to the one shown here:  GET/index.html HTTP/1.1  Host:www.example.com | 8.The client has complete control over server push that can adjust the priority or even disable the server push; whenever required, it will only send a SETTINGS frame to change the HTTP/2 component. |
| 9.This request uses the GET method, which asks for data from the host server listed after Host:. | 9.The server push has many features but is still not supported by the many web browsers that disable many critical components for the client, such as cancelling a cached resource, allowing duplicate resources, etc. And this technique should be used based on the requirement of the web application. |
| 10.In response to this request, the example.com web server returns an HTML page to the requesting client, in addition to any images, stylesheets, or other resources called for in the HTML. | 10.To read more on web application optimization and server push, you can check out the PRPL pattern developed by Google. |

(ii) Write a blog about objects and its internal representation in Javascript

1.Objects, in JavaScript, is it’s most important data-type and forms the building blocks for modern JavaScript.

2.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).

3.Objects are more complex and each object may contain any combination of these primitive data-types as well as reference data-types.

4.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.

5.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.

6.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.

objectName.propertyName

7.All JavaScript variables, both the object name (which could be a normal variable) and property name are case sensitive. You can define a property by assigning it a value. For example, let’s create an object named myCar and give it properties named make, model, and year as follows:

var myCar = new Object();

myCar.make = 'Ford';

myCar.model = 'Mustang';

myCar.year = 1969;

8.Properties of JavaScript objects can also be accessed or set using a bracket notation (for more details see property accessors). Objects are sometimes called associative arrays, since each property is associated with a string value that can be used to access it. So, for example, you could access the properties of the myCar object as follows:

myCar['make'] = 'Ford';

myCar['model'] = 'Mustang';

myCar['year'] = 1969;