|  |  |
| --- | --- |
| **Http 1.1** | **Http 2** |
| **Multiplexing**:  It loads resources one after the other, so if one resource cannot be loaded, it blocks all the other resources behind it. | HTTP/2 is able to use a single TCP connection to send multiple streams of data at once so that no one resource blocks any other resource. |
| **Server push**: Typically, 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 | HTTP/2 solves this problem by allowing a server to "push" content to a client before the client asks for it. |
| **Header compression**: 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. | HTTP/2 uses a more advanced compression method called HPACK that eliminates redundant information in HTTP header packets. This eliminates a few bytes from every HTTP packet. Resulting in faster loading. |
| **Prioritization:** Prioritization affects a webpage's load time. For example, certain resources, like large JavaScript files, may block the rest of the page from loading if they have to load first  . | In HTTP/2, developers have hands-on, detailed control over prioritization. This allows them to maximize perceived and actual page load speed to a degree that was not possible in HTTP/1.1. HTTP/2 offers a feature called weighted prioritization. This allows developers to decide which page resources will load first, every time |
| **Reasource Inlining:** A major drawback of resource inlining, then, is that the client cannot separate the resource and the document. A finer level of control is needed to optimize the connection, a need that HTTP/2 seeks to meet with server push. | Since HTTP/2 enables multiple concurrent responses to a client’s initial GET request, a server can send a resource to a client along with the requested HTML page, providing the resource before the client asks for it. This process is called server push. |

Question1: Write a blog on Difference between HTTP1.1 vs HTTP2

**HTTP:** The Hypertext Transfer Protocol (HTTP) is designed to enable communications between clients and servers. HTTP works as a request-response protocol between a client and server.

**Advantages:**

* HTTP/2 supports full multiplexing for requests as well as responses over a single TCP connection. Due to these capabilities, lower page load times are achieved by removing needless latency and improving the overall capacity of network alongside its availability.
* Resource usage has increased dramatically for machines processing requests to deliver media-rich content and complex web designs. Developers worked hard around optimization hacks, and as a result, the robust solution of HTTP/2 was obtained. Features, such as server push, stream dependency and prioritizing, header compression, and binary format layer, have improved network utilization as the core advantage.
* The HTTP/2’s ability to transmit more data per client-server communication cycle greatly improves web performance. As a result, increased user satisfaction, better SEO, greater productivity, growing userbase, and improved sales figures can be achieved.
* All modern browsers support HTTP/2 over HTTPS with the SSL certificate installation. To open HTTPS capable invisible proxy ports on every relevant port, OWASP ZAP or its alternatives could be used.
* The use of the HPACK algorithm enables HTTP/2 to overcome the common API security threats. This protocol has commands in binary format and compresses the HTTP header metadata to protect sensitive data shared between both machines.

**Disadvantages:**

* While HTTP/2 mitigated the effects of HOL blocking in its predecessor, TCP-level block still causes problems.
* For client machines operating on a slow network, data packets drop bit by bit, and the network quality gets degraded to a single HTTP/2 connection. Due to this, the entire process slows down, thereby blocking a lot of data transfer.
* The cookie security failure is still not addressed in HTTP/2 like its precursor. Cookies are .txt files containing client data obtained by the server and website. However, these cookies may get stolen or tampered with by hackers, who can access personal user data, even without passwords.

------------------------------------------------------------------------------------------------------------------------------

‍

**Question 2:**

Write a blog about objects and its representation in javascript.

**What is Object?**

 “A JavaScript object is a collection of named values having state and behavior (properties and method)”.

For example: Person, car, pen etc .

Lets take a example as “CAR”.

All cars have the same properties, but the property values differ from car to car. All cars have the same methods, but the methods are performed at different times.

List of properties in car.

1. Make: Mercedes
2. Model: C-Class
3. Color: White
4. Fuel: Diesel
5. Weight: 850kg
6. Mileage: 8Kmpl
7. Rating: 4.5

Syntax:

|  |
| --- |
| var <object-name> = {key1: value1, key2: value2,... keyN: valueN}; |

The following code assigns **many values** (Mercedes, C-class, White and soo on) to a **variable** named Car:

|  |
| --- |
| var car = {Make: “Mercedes”, Model: “C-Class”, Color: “White”, Fuel: Diesel, Weight: “850kg”, Mileage: “8Kmpl”, Rating: 4.5}; |

**2)Object Properties**

**The syntax for adding a property to an object is :**

|  |
| --- |
| ObjectName.ObjectProperty = propertyValue; |

**The syntax for deleting a property from an object is:**

|  |
| --- |
| delete ObjectName.ObjectProperty; |

**The syntax to access a property from an object is:**

|  |
| --- |
| objectName.property        // Car.Make  //or  objectName["property”]    // Car["Make"] |

The name:values pairs (in JavaScript objects) are called **properties**.

|  |
| --- |
| var car = {Make: “Mercedes”, Model: “C-Class”, Color: “White”, Fuel: Diesel, Weight: “850kg”,Mileage: “8Kmpl”, Rating: 4.5}; |

## ****3)Object Methods****

An object method is an object property containing a function definition.

i.e.,

Let’s assume to start the car there will be a mechanical functionality.

|  |
| --- |
| function(){return ignition.on} |

simple definition for Java Script Object methods is “Methods are actions that can be performed on objects.”.

-----------------------------------------End-------------------------------------------------