Day-1

1. Write a blog on difference between HTTP 1.1 vs HTTP 2.

Answer:

HTTP/2 improved on HTTP/1.1 in a number of ways that allowed for speedier content delivery and improved user experience.

**Binary protocols:**

Binary protocols consume less bandwidth, are more efficiently parsed and are less error-prone than the textual protocols used by HTTP/1.1. Additionally, they can better handle elements such as whitespace, capitalization and line endings.

**Multiplexing:**

HTTP/2 is multiplexed, i.e. it can initiate multiple requests in parallel over a single TCP connection. As a result, web pages containing several elements are delivered over one TCP connection. These capabilities solve the head-of-line blocking problem in HTTP/1.1 in which a packet at the front of the line blocks others from being transmitted.

**Header compression:**

HTTP/2 uses header compression to reduce the overhead caused by TCP's slow start mechanism.

**Server push:**

HTTP/2 servers push likely-to-be-used resources into a browser's cache, even before they are requested. This allows browsers to display content without additional cycles.

**Increased security:**

Web browsers only support HTTP/2 via encrypted connections, increasing user and application security.

**Main goals of developing HTTP/2 was:**

* Protocol negotiation mechanism- protocol electing, eg. HTTP/1.1, HTTP/2 or other.
* High-level compatibility with HTTP/1.1- methods, status codes, URIs and header fields.
* Page load speed improvements trough
* Compression of request headers
* Binary protocol
* HTTP/2 Server Push
* Request multiplexing over a single TCP connection
* Request pipelining
* HOL blocking(Head-of-line)-Package blocking

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

Answer:

**Object:**

* In JavaScript, an object is a standalone entity, with properties and type.
* For example. A cup is an object, with properties. A cup has a color, a design, weight, a material it is made of, etc.
* The same way, JavaScript objects can have properties, which define their characteristics.

**Creating Objects in JavaScript:**

1. By object literal
2. By creating instance of Object directly (using new keyword)

* JavaScript object is a collection of named values having state and behaviour (properties and method).

For example: Person, car, pen, bike, Personal Computer etc.

In case of cars, 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.

* For example, car and list out its properties:

Make: Mercedes

Model: C-Class

Colour: White

Fuel: Diesel

Weight: 850kg

Mileage: 8Kmpl

Rating: 4.5

**1)Objects:**

Objects are variables too. But objects can contain many values.

* The following code assigns a simple value (Mercedes) to a variable named car:

var car = "Mercedes";

* 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};

* The values are written as name:value pairs (name and value separated by a colon).

Syntax:

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

So, conclusion and definition for JS objects is “JavaScript objects are containers for named values”.

**2)Object Properties:**

Properties can usually be changed, added, and deleted, but some are read only.

* 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};

The object properties can be different primitive values, other objects and functions.

* 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 ex: Car.Make

objectName["property”] ex: Car["Make"]

Properties are the values associated with a JavaScript object.

**3)Object Methods:**

* Methods are actions that can be performed on objects.
* An object method is an object property containing a function definition.

function() {return ignition. On} and so similar is to stop/brake/headlights on & off, etc.