Q2: Hypertext Transfer Protocol (HTTP) is an application protocol that is, currently, the foundation of data communication for the World Wide Web

**HTTP-1.1**

* It was released in 1997.
* Internet landscapes was constantly changing with websites becoming more dynamic and heavy.
* **Features** – CORS, Keep-alive was introduced.
* **Flaws**: HOL- Head of Line blocking, Repetition of header data. More focus on gzip, minifying css/js, Caching, ..etc.

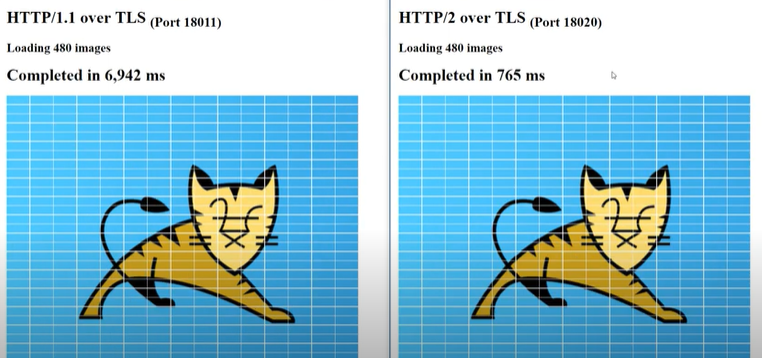
**HTTP-2.0**

* It was released in 2015.
* Single served TCP connection. Streams are created.
* **Features** – HPACK, PUSH.

HPACK: Header data is separate from request data and can be zipped.it reduces HTTP request size. It also enables reuse of header data which is repeated in every request.

PUSH: can be setup on server side. PUSH frames enables us to send mandatory resources in advance along with an HTTP response. It should be used with case as this can lead to increase in size of HTTP response.

* You can keep on using gzip, leverage browser caching, minify CSS/JS etc. to further improve the speed.
* Almost all the modern web server support this.



Q3: In JavaScript, objects are king. If you understand objects, you understand JavaScript.

In JavaScript, almost "everything" is an object.

* Booleans can be objects (if defined with the new keyword)
* Numbers can be objects (if defined with the new keyword)
* Strings can be objects (if defined with the new keyword)
* Dates are always objects
* Maths are always objects
* Regular expressions are always objects
* Arrays are always objects
* Functions are always objects
* Objects are always objects

All JavaScript values, except primitives, are objects.

JavaScript Primitives

A **primitive value** is a value that has no properties or methods.

A **primitive data type** is data that has a primitive value.

JavaScript defines 5 types of primitive data types:

* string
* number
* boolean
* null
* undefined

Primitive values are immutable (they are hardcoded and therefore cannot be changed).

Objects are complex and each object may contain a combination of 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.

Other way around, 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.  
An object can be created with figure brackets {} with an optional list of properties. A property is a “key: value” pair, where a key is the property name value can be anything.

Let’s have an example of my favorite merc car and list out its properties(Features):

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

Taking the above as reference, I'll stress up on objects, Object properties and Methods.

**1) Objects:**

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

var car = "Mercedes";

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

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**

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};



From the above snippet, let’s have a look what falls under property and property value:

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

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

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"]

//or

objectName[expression]   // x = "Make"; Car[x]

So, Conclusion and simple definition for Java Script properties is “Properties are the values associated with a JavaScript object”.

**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}

and so similar is to stop/brake/headlights on & off, etc.

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