**Day 01**

1. **Difference between HTTP1.1 vs HTTP2**
2. HTTP1 is slow while HTTP2 is fast since it used prioritization process for data transmission
3. In HTTP1 data is loaded one after other while in HTTP2 stream of data is transmitted at once
4. In HTTP1 server pushes content only if the client asks for it while in HTTP2 the server pushes before the client asks for it
5. Normal header compression is used in HTTP1 while HPACK is used inn HTTP2
6. **Objects and its internal representation in JavaScript**

Objects, in JavaScript, is most important data-type and forms the building blocks for modern JavaScript. These objects are different from JavaScript’s primitive data-types. While primitive data-types all store a single value each.

Objects are more complex and each object may contain any combination of these primitive data-types as well as reference data-types in form of Key: value pair. These keys can be variables or functions and are called properties and methods, respectively, in the context of an object. 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.

1. Adding a property - ObjectName.ObjectProperty = propertyValue;
2. Deleting a property - delete ObjectName.ObjectProperty;
3. Accessing a property - objectName.property

//or

objectName["property”]

//or

objectName[expression]

Simple JS object example - let bike = {name: 'SuperSport', maker:'Ducati', engine:'937cc'};

1. The following example also creates a **new JavaScript object** with four properties:

var person = new Object();

person.firstName = “John”;

person.lastName = “Doe”;

person.age = 50;

person.eyeColor = “blue”;

1. **Using the Object.create method**

Objects can also be created using the Object.create() method. This method can be very useful, because it allows you to choose the prototype object for the object you want to create, without having to define a constructor function.

// Animal properties and method encapsulation

var Animal = {

type: 'Invertebrates', // Default value of properties

displayType: function() { // Method which will display type of Animal

console.log(this.type);

}

};

// Create new animal type called animal1

var animal1 = Object.create(Animal);

animal1.displayType(); // Output:Invertebrates

// Create new animal type called Fishes

var fish = Object.create(Animal);

fish.type = 'Fishes';

fish.displayType();

// Output:Fishes