Day 1 : FSD TASK 1:

1.Difference between HTTP1.1 vs HTTP2?

HTTP: Hypertext Transfer Protocol

* HTTP is used for passing a request from the client-server (which can be our laptop) to the web-server and again the response from the web-server back to the client-server.
* HTTP works in the 7th layer of the OSI model and uses port 80 for communication.
* When you open a web browser, you are using HTTP(s) indirectly which works on port 443 .
* HTTPS is an application protocol that runs on top of the [TCP/IP](https://medium.com/javarevisited/5-best-books-and-courses-to-learn-computer-networking-tcp-ip-and-udp-protocols-5a0e4dce75fa) that forms the foundation of the internet.
* This protocol facilitates the exchange of resources between client and server and the resources can be anything like text, image, video, etc.
* The web browser / mobile apps are the HTTP clients that [sends HTTP request](https://javarevisited.blogspot.com/2015/10/how-to-send-http-request-from-unix-or-linux-curl-wget-example.html) to the server’s URL address which in turn converts to an IP after the DNS resolution.
* In the web server, an HTTP daemon process keeps running to handle the [HTTP requests](https://javarevisited.blogspot.com/2013/07/how-ssl-https-and-certificates-works-in-java-web-application.html) as they arrive. This will process the request and provides the response to the clients.

|  |  |
| --- | --- |
| HTTP1 | HTTP2 |
| HTTP1 uses a single connection for each request which can lead to slower the performance due to head-of-line blocking. | Whereas HTTP2 uses multiplexing, allowing multiple requests and responeses to be sent in parallel over the single connection and improving efficiency. |
| Headers are not compressed, leading to increased overhead. | Whereas utilizes header compression, reducing the amount of data transmitted and improving speed |
| It is text based protocol. | It is binary protocol for no loss in data tranmission |
|  |  |

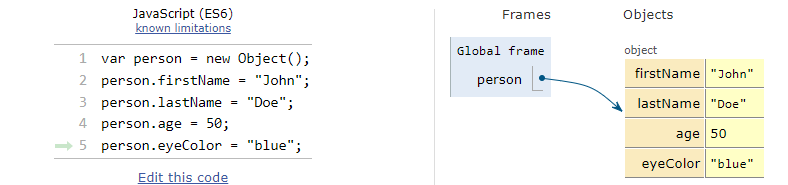
2. Objects and its internal Representation in Javascript

* **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.



In the above code , var person = new object();

* person is the object, in which firstName, lastname, age, eyecolour are the properties of

the Object.

* Every property in the object is stored by a key:value format.
* Properties in the object are assigned a reference value and are given a reference or a

pointer to that value. That reference or pointer points to the location in memory where

the object and its properties are stored.

Object method: Methods are actions that can be performed on objects.

**Creating Objects In JavaScript :**

**In JavaScript objects can be created by the following ways:**

## Create JavaScript Object with Object Literal

## Create JavaScript Object with Constructor

## Using the JavaScript Keyword new

## Using the Object.create method

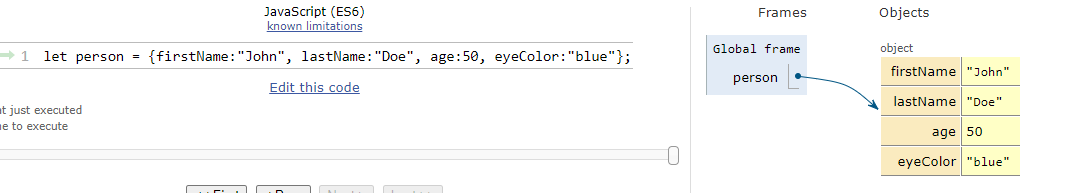
.

## Create JavaScript Object with Object Literal

* One of easiest way to create a javascript object is object literal, i.e by simply

define the property and values inside curly braces as shown below.

* Ex: **let person = {firstName: “SuperSport” , lastName: “Doe” , eyecolour: “937cc”};**



## Create JavaScript Object with Constructor

* Constructor is nothing but a function and with help of new keyword,

constructor function allows to create multiple objects of same flavor as shown

below:

* function vehicle(name, maker) {

this.name = name;

this.maker = maker;

}

let car1 = new Vehicle(‘Fiesta’ , ‘Ford’ );

let car2 = new Vehicle(‘Santa Fe’ , ‘Hyundai’ );

console.log(car1.name);

console.log(car1.maker);

## Using the JavaScript Keyword new

* The following example also creates a new JavaScript object with four properties:
* Ex: var person = new Object();  
   person.firstName = “John”;  
   person.lastName = “Doe”;  
   person.age = 50;  
   person.eyeColor = “blue”;

# 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