**Question 1:**

**Write a blog on Difference between HTTP1.1 vs HTTP2?**

**Answer:**

Starting from basics, HTTP stands for Hyper Text Transfer Protocol.

WWW is about communication between web clients and servers.

Communication between client computers and web servers is done by sending HTTP Requests and receiving HTTP Responses.

Clients are often browsers (Chrome, Edge, Safari), but they can be any type of program or device.

Servers are most often computers in the cloud.

Communication between clients and servers is done by requests and responses which shall be as following cycle:

1. A client (a browser) sends an HTTP request to the web

2. A web server receives the request

3. The server runs an application to process the request

4. The server returns an HTTP response (output) to the browser

5. The client (the browser) receives the response

Difference between HTTP1.1 and HTTP2

HTTP2 is much faster and more reliable than HTTP1. HTTP1 loads a single request for every TCP connection, while HTTP2 avoids network delay by using multiplexing.

HTTP is a network delay sensitive protocol in the sense that if there is less network delay, then the page loads faster. However, an impressive increase in network bandwidth only slightly improves page load time. This is key to understanding the differences in performance efficiencies between the different versions of HTTP. Back in the day when people used dial up modems, web pages were simple and it was the actual data transfer between the server and the client that contributed towards the largest chunk of the page load time. Today the actual downloading of resources from server takes a negligible portion of the total page load time due to the tremendous increase in bandwidth availability. It is the time taken to establish the TCP connection and making requests that impacts performance.

|  |  |  |
| --- | --- | --- |
| Differentiator | HTTP/1.1 | HTTP/2 |
| Year | 1997 | 2015 |
| Key Features | It supports connection reuse i.e. for every TCP connection there could be multiple requests and responses, and pipelining where the client can request several resources from the server at once. | Uses multiplexing, where over a single TCP connection, resources to be delivered are interleaved and arrive at the client almost at the same time. It is done using streams which can be prioritized, can have dependencies and individual flow control. It also provides a feature called server push that allows the server to send data that the client will need but has not yet requested. |

**Question 2:**

**Write a blog about objects and its internal representation in Java script.**

Answer:

Objects are important data types in javascript. Objects are different than primitive datatypes (i.e. number, string, boolean, etc.). Primitive data types contain one value but Objects can hold many values in form of Key: value pair. Where each key-value pair is called a property. The key is the name of the property and the value is the value of the property.

The key of an object is a string while the value can be any [data type](https://www.tutorialstonight.com/js/js-data-types.php), including another object.

**How to create object in JavaScript**

JavaScript objects can be created in many different ways. Some of the ways are:

1. [JavaScript object literal](https://www.tutorialstonight.com/js/js-objects.php#javascript-object-literal)
2. [JavaScript Object.create method](https://www.tutorialstonight.com/js/js-objects.php#javascript-object-create)
3. [JavaScript object constructor](https://www.tutorialstonight.com/js/js-objects.php#javascript-object-constructor)
4. [object using class](https://www.tutorialstonight.com/js/js-objects.php#javascript-object-using-class)

Among all ways to create objects in JavaScript, the most common way is the object literal.

### **1.** **JavaScript Object Literal**

The simplest way to create an object in JavaScript is using the object literal.

It is a comma separated key-value pair enclosed in curly braces ({}) assigned to a variable.

To create an empty object just assign a curly brace to a variable:

let person = {};

The object defined above is an empty object.

The properties are added to an object in form of key-value pair, the key should be a string and the value could be any data type.

key-value pairs are separated by a comma (,).

let person = {

firstName: “Raghu”,

lastName: “Annamdasu”

};

### **2. Using Object.create method**

The Object.create() method is used to create an object from an existing object. It creates a new object with the same properties as the existing object.

It uses an existing object as a prototype of the newly created object

Object.create(object);

we can pass the reference object in the method as the first parameter. This will create a new object whose values can be changed later.

let person = {

firstName: "Raghu",

lastName: "Annamdasu",

fullName: function() {

return "My name is " + this.firstName + " " + this.lastName;

}

}

// create a new object

let newPerson = Object.create(person);

//Change the value of new object

newPerson.firstName = "Ram";

newPerson.lastName = "Kotha";

console.log(newPerson.fullName()); // returns My name is Ram Kotha

We can create an object without a prototype by giving null as a first parameter and later we can update the properties.

// create a new empty object

var person = Object.create(null)

Console.log(person); // { }

person.firstName=”Raghu”;

person.lastName = “Annamdasu”;

console.log(person); // {firstName: “Raghu”, lastName: “Annamdasu”}

### **3. Javascript Object Constructor**

Another way to create an object is using the object constructor.

The object constructor is a function that is used to create an object. It is defined using the “new” keyword.

The object constructor is used to create an object with a specific set of properties and methods.

If you want to create multiple objects of the same type then it is better to use the **object constructor**.

Here are the steps to create an object using the object constructor:

1. Create a [function](https://www.tutorialstonight.com/js/js-function.php) that will be used to create an object.
2. The function should have a parameter that is the object properties.
3. Now use the “new” keyword with the function to create an object.

// creating a function to create an object

function person(firstName, lastName, age) {

this.firstName = firstName;

this.lastName = lastName;

this.age = age;

}

// creating an object using the function

let person1 = new person("Raghu", "Annamdasu", 29);

console.log(person1); // {firstName: “Raghu”, lastName: “Annamdasu”, age: 29}

let person2 = new person("Ram", "Kotha", 26);

console.log(person2); // {firstName: “Ram”, lastName: “Kotha”, age: 26}