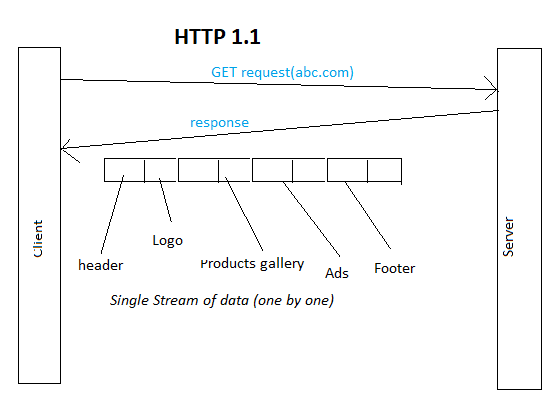
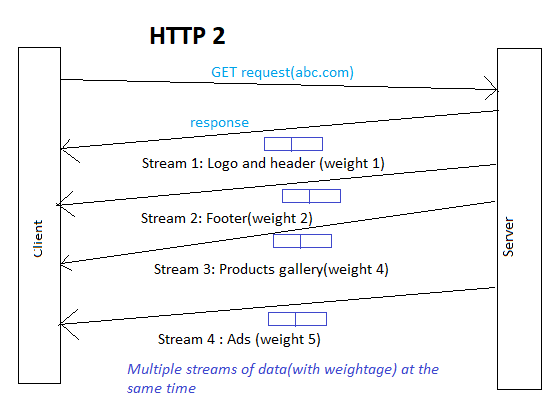
**Day 1– 23Jan 2024 Task**

|  |  |  |
| --- | --- | --- |
| **Sno** | **HTTP 1.1** | **HTTP 2** |
| 1**.Multiplexing** | HTTP/1.1 loads resources one after the other. | HTTP/2 is able to use a single TCP connection to send multiple streams of data at once |
| 2. | So if one resource cannot be loaded, it blocks all the other resources behind it. | No one resource blocks any other resource. |
| **3. Header Compression** | Basic compression | Advanced Compression method called **HPACK** that eliminates redundant information in HTTP header packets. |
| 4.**Server push** | A server only serves content to a client device if the client asks for it. | A server can "push" content to a client before the client asks for it. The server also sends the client what to expect. |
| **5.Prioritization** | Not available. | Developers can assign each of these data streams a different weighted value, and the value tells the client which data stream to render first. |

1.Difference between HTTP1.1 vs HTTP2





2) Javascript Objects

Objects in javascript is a set of properties.

Property is a pair containing:

1. Key – can be a

* String
* Number
* Identifier

*\*\*\*\*Note:All keys are represented as strings internally\*\*\*\**

1. Value

* Any datatype like number, strings or other objects,or functions
* If the value is a function ,the property is called a method.

When the value is a function, the property is called a *method*.

Syntax:

objectName = {

member1Name: member1Value,

member2Name: member2Value,

member3Name: member3Value,

};

|  |  |
| --- | --- |
| **JS** | **Real world** |
| Object | A “thing”  Ex: “Movie Avatar” |
| Property(key : value) | A trait(name of the trait : value of the trait)  ex: (name=”Avatar 3D”)  (genre=”Animation”)  (director=”James Cameron”)  (release date=”22 dec 2022”) |
| Method | Actions that the thing can do  Ex:  (playmovie())  Resume()  Stop()  Pause() |

*When an object is created, a block of memory is created for it in the heap memory.*

*A reference to the object is created in the stack memory.*

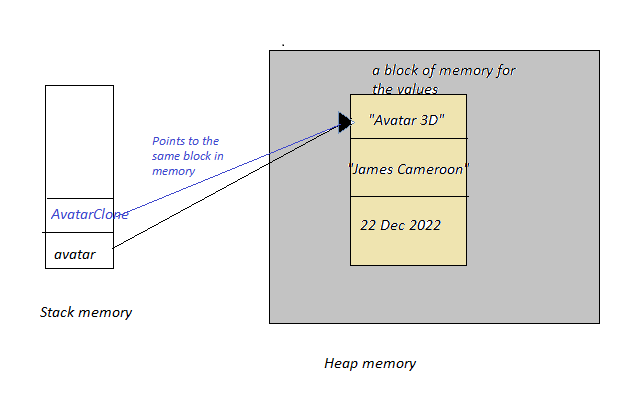
Ex:

Var avatar={

Name: “Avatar 3D”,

Director:”James cameroon”,

ReleaseDate:”22Dec2022”}



* Whenever a value is changed through one object, it affects all the references pointing to the same object*. Objects are copied by reference.*

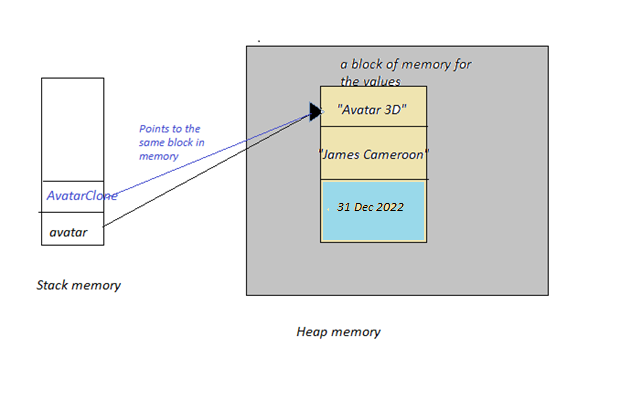
var avatarClone=avatar;

avatar.releaseDate=”31Dec2022”

Console.log(avatarClone.releaseDate)

OUTPUT:

31Dec2022



Arrays, functions, dates are also special types of objects.

var obj={

1:”one”,

2:”two”,

1.34:”value 2”,

name:”object”,

size:34.56,

method1: ()=>{//do somehting}

}

Accessing object properties:

|  |  |  |
| --- | --- | --- |
| **Key type** | **[] Notation** | **. Notation** |
| number   * 1 * 2 * 1.34 | * obj[1] * obj[2] * obj[1.34] | NA |
| string without spaces   * name * size | * obj[name] * obj[size] | obj.name  obj.size |
| string with spaces   * “release date * “1 key” | * obj[“release date”] * obj[“1 key”] | NA |
| Methods   * method1 | NA | obj.method1() |

\*\*obj is the name of the object