1.**Difference between HTTP1.1 vs HTTP2**

|  |  |
| --- | --- |
| **HTTP1.1** | **HTTP2** |
| * Less efficient | * More efficient and faster |
| * Server send data one after the other and will take more time to load all content. If one resource cannot be loaded it will block the rest of the resources. | * Server send all of the data at once so no resource blocks the other ones. This is called multiplexing. |
| * Prioritization is less likely to be done in HTTP1.1 | * Uses weighted prioritization where the developer decides which page resources will load first . |
| * Small files load quicker than larger ones. So HTTP1.1 compresses HTTP messages. | * HTTP2 uses advanced compression called HPACK. This eliminates redundant information in every Header packets and thus so many bytes are eliminated. |
| * Server serves the client only if the client asks for it. | * HTTP allows server to push content before the client asks for it. |

2. **Objects:**

Objects are datatypes and they are the representation of real-world entities by defining the properties along with their values. In Javascript they are defined as unordered collection of data in the form “key:value” pairs.

**Ways to create objects:**

* **Object literal**

It is a comma separated list of key-value pairs in curly braces.

Eg: var student={id:1, name:’John’,DOB-‘1/1/2001’}

* **Object.create()**

This creates a new object using an existing object as the prototype of the new object.

Eg: var student1= Object.create(student)

student.id=2

student.name=’Jack’

* **Object constructor**

This can be used when we have to create multiple objects of similar type. Here a blueprint is created and multiple objects can be created using the constructor as the wrapper for the new objects.

Eg: construction function

function person(name,age,DOB){

this.name=name;

this.age=age;

this.DOB=DOB;

}

Creating objects from the above construction

Var p1= new person(“John”, “17” ,”1/1/2001”)

Var p2= new person(“Jack”, ”16” ,”1/1/200”)

* **Object.assign()**

This is used to copy all data from one or more objects to another object.

Eg:

Var obj1={a:1}

Var obj2={b:2}

Var obj3={c:3}

Var newobj= object.assign(obj1,obj2,obj3}

Console.log(newobj)

o/p= object {a:1,b:2,c:3}

* **Object.fromEntries**

This transforms key value pairs to an object.

var student={[

[‘id’,1],

[‘name’,’Jack’]

]}

Var student1=object.fromEntries(student)

Console.log(student1)

o/p= object{id:1,name:Jack}