

## Difference b/w HTTP1 and HTTP2

1. The Hypertext Transfer Protocol (HTTP) is a data communications protocol and acts as the foundation of the World Wide Web. After the original version was released, the functionality was extended in HTTP/1.0, and overall, it was made more versatile. In 2015, a new version of HTTP called HTTP/2 was created. HTTP/2 solves several problems that the creators of HTTP/1.1 did not anticipate. HTTP/2 is much faster and more efficient than HTTP/1.1. One of the ways in which HTTP/2 is faster is in how it prioritizes content during the loading process.
2. **Server push:** Typically, a server only serves content to a client device if the client asks for it. However, this approach is not always practical for modern webpages, which often involve several dozen separate resources that the client must request. HTTP/2 solves this problem by allowing a server to "push" content to a client before the client asks for it. The server also sends a message letting the client know what pushed content to expect – like if Bob had sent Alice a Table of Contents of his novel before sending the whole thing.
3. **Multiplexing:** HTTP/1.1 loads resources one after the other, so if one resource cannot be loaded, it blocks all the other resources behind it. In contrast, HTTP/2 is able to use a single TCP connection to send multiple streams of data at once so that no one resource blocks any other resource.
4. **Header compression:** Small files load more quickly than large ones. To speed up web performance, both HTTP/1.1 and HTTP/2 compress HTTP messages to make them smaller. However, HTTP/2 uses a more advanced compression method called HPACK that eliminates redundant information in HTTP header packets.

## **Objects :**

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. These keys can be variables or functions and are called properties and methods, respectively, in the context of an object.

Every object has some property associated with some value. These values can be accessed using these properties associated with them.

```
var myCar = new Object();
```

```
myCar.make = 'Suzuki';
```

```
myCar.model = 'Altros';
```

```
myCar.year = 1978;
```

```
myCar.wheels = 2;
```

After creating myCar object, the value inside the object can be accessed using keys.

i.e.

```
myCar.year
```

Output: 1978