**1. Difference Between HTTP1.1 vs HTTP2**

HTTP (Hypertext Transfer Protocol) is the foundation of data communication on the World Wide Web. HTTP1.1 and HTTP2 are two versions of this protocol that differ in several key aspects. Here's a breakdown of the differences between HTTP1.1 & HTTP2.

* **Request-Response Model:** HTTP1.1 uses a request-response model where a single request is sent, and a single response is received. In HTTP2, multiple requests can be sent in parallel and multiple responses can be received at the same time, which can result in faster page load times.
* **Binary Protocol:** HTTP1.1 uses text-based format for data transfer, while HTTP2 uses a binary format. The binary format is more efficient and easier to parse, leading to faster data transfer.
* **Header Compression:** In HTTP1.1, headers are sent in plain text, which can result in slow page load times as the size of the headers increases. HTTP2 uses header compression to reduce the size of headers, which leads to faster page load times.
* **Server Push:** HTTP2 introduces the concept of server push, which allows the server to send resources to the client before they are requested. This can lead to faster page load times, as the resources are already in the client's cache when they are needed.
* **Prioritization:** In HTTP1.1, it's up to the client to decide which requests are made first. In HTTP2, the server can specify the priority of requests, which allows for faster page load times.

HTTP2 offers significant improvements over HTTP1.1, including multiplexed request-response, header compression, server push, binary format, and prioritization. These improvements lead to faster and more efficient data transfer, making HTTP2 a preferred choice for modern web applications.

**2. Objects and their Internal Representation in JavaScript**

1. Objects in JavaScript are a collection of key-value pairs that represent a single entity, such as a person, place, or thing. They allow you to store and manipulate data in a way that is easy to understand and manage.
2. The internal representation of an object in JavaScript is called an object wrapper. It is an instance of the Object constructor and is automatically created when you create a new object.
3. The object wrapper is used to manage the object's properties and methods and provides several built-in functions that can be used to interact with the object.
4. One of the most important things to understand about objects in JavaScript is that they are always passed by reference. This means that when you pass an object to a function, you are not passing a copy of the object, but instead you are passing a reference to the original object. This means that any changes you make to the object inside the function will affect the original object as well.
5. Another important aspect of objects in JavaScript is their prototype chain. Every object in JavaScript is linked to a prototype object, which is essentially a blueprint for the object. This prototype object can contain properties and methods that are automatically inherited by the object. This allows you to reuse code and make it easier to manage complex data structures.
6. When you access a property of an object in JavaScript, the engine first looks for the property on the object itself. If it doesn't find the property, it will then look in the object's prototype. This continues up the prototype chain until the property is found or the chain reaches the end.
7. Finally, it's important to understand that objects in JavaScript are dynamic, meaning that you can add or remove properties and methods from an object at any time. This gives you a lot of flexibility and makes it easy to work with data that is constantly changing.
8. Objects in JavaScript are a powerful data structure that allows you to manage and manipulate data in a way that is easy to understand and maintain. Their internal representation as object wrappers, combined with the ability to access properties through the prototype chain and the dynamic nature of objects, make them a fundamental part of the language.

Top of Form