**HTTP1.1 Vs HTTP2**

HTTP/1.1 and HTTP/2 are both protocols for transferring hypertext requests and information on the web, but they have some key differences in terms of performance, features, and how they handle data. Here are some of the main differences between HTTP/1.1 and HTTP/2:

* **Multiplexing:**
  + **HTTP/1.1:** Uses a single connection per request, which means that each resource (e.g., image, script, stylesheet) is requested and delivered one at a time.
  + **HTTP/2:** Supports multiplexing, allowing multiple requests and responses to be sent in parallel over a single connection. This can significantly improve page load times, especially for complex web pages with many resources.
* **Header Compression:**
  + **HTTP/1.1:** Headers are not compressed, leading to higher overhead due to redundant information being sent with each request and response.
  + **HTTP/2:** Uses header compression, reducing the amount of data transferred and improving efficiency.
* **Binary Protocol:**
  + **HTTP/1.1:** Uses plain text for its protocol, which is human-readable but can be less efficient.
  + **HTTP/2:** Employs a binary protocol, which is more efficient for both machines and humans. While it's not as readable as plain text, the parsing is faster, and the compact representation reduces overhead.
* **Prioritization:**
  + **HTTP/1.1:** Does not have built-in support for prioritizing requests, so all resources are loaded based on their order in the HTML document.
  + **HTTP/2:** Introduces stream prioritization, allowing the client to specify the priority of each resource. This enables the browser to load critical resources first, improving page rendering times.
* **Server Push:**
  + **HTTP/1.1:** Relies on the browser to request all resources, even if they are needed to render the page.
  + **HTTP/2:** Introduces server push, which allows the server to push resources to the client before they are explicitly requested. This can reduce latency by anticipating the client's needs.
* **Connection Handling:**
  + **HTTP/1.1:** Requires multiple connections for parallelism, leading to higher resource usage.
  + **HTTP/2:** Uses a single connection, reducing latency and resource usage.
* **Backward Compatibility:**
  + **HTTP/1.1:** Widely supported and used, but lacks some of the optimizations introduced by HTTP/2.
  + **HTTP/2:** Designed to be fully backward-compatible with HTTP/1.1, allowing for a smooth transition.

**Objects and Its Internal Representation**

### **Internal Structure:**

**Properties:**

* + Objects in JavaScript have properties, which are the key-value pairs that define the characteristics and attributes of the object.
  + Properties can be added, modified, or deleted dynamically.

**Methods:**

* Objects can have methods, which are functions associated with the object.
* Methods can be invoked to perform actions related to the object.

**Prototype:**

* Each object in JavaScript is linked to a prototype object from which it inherits properties and methods.
* The prototype chain is a mechanism for object inheritance.

**Hidden Classes and Optimizations:**

* JavaScript engines use various optimizations to improve the performance of object manipulation.
* Hidden classes are a concept in the V8 JavaScript engine (used in Chrome and Node.js) that helps optimize property access.

### **Object Representation in Memory:**

Internally, JavaScript engines manage objects in memory, and the details can vary between different engines. In general, objects may be represented using a combination of dictionaries (for properties) and additional structures to handle methods and prototype relationships efficiently.

Understanding the internal representation is not typically necessary for everyday JavaScript programming, but having a conceptual grasp of properties, methods, prototypes, and how they relate to each other can help you write more efficient and effective code.