***HTTP/1.1:***

*Request-Response Model*: Each request triggers a new connection, leading to multiple setups and inefficiencies.

*Sequential Processing:* Delays in one resource slow down others due to the "head-of-line blocking" problem.

*Resource Multiplicity:* Multiple connections try to fetch resources simultaneously, taxing servers and networks.

*Optimization Challenges:* HTTP/1.1 uses gzip compression to reduce the size of data transferred between the client and server. Additionally, developers use techniques like image spriting and minification to optimize assets, reducing load times

## ***HTTP/2.0:***

*Multiplexing:* Multiplexing enables concurrent requests and responses within one connection, eliminating delays.

*Binary Brilliance:* Binary protocol speeds up processing and error handling, facilitating faster data transmission.

*Server Push Sorcery:* Servers proactively send assets, reducing load times by milliseconds.

Header Huffman Coding: HPACK compression optimizes bandwidth by reducing repetitive header data.

*Priority:* Unlike the linear processing of HTTP/1.1, HTTP/2 allows for stream prioritization. This means more critical resources can be given higher priority, ensuring faster loading of essential content.

*At the end, we should know one thing that HTTP1.1 has been a pioneer and paved a way in web but HTTP2.0 has redefined and brought a revolutionary in web. It provided the faster loads, decreased the latency issue and made the workflow* *efficient, it has increased the web performance standards to high level.*