1. **Write a blog on Difference between HTTP1.1 vs HTTP2**

**HTTP/1.1**

introduced in 1999, was the standard for web communication for many years. It served its purpose well but had some limitations that became more apparent as web content became more complex. Here are some key characteristics of HTTP/1.1:

Multiplexing: In HTTP/1.1, browsers open multiple connections to a server to retrieve different assets. This can result in a phenomenon called "head-of-line blocking," where a slow-loading asset delays the loading of subsequent assets on the same connection.Compression: HTTP/1.1 lacks native header compression. Each request and response contains verbose headers, which can lead to unnecessary overhead.Request prioritization: Prioritizing resource loading was difficult in HTTP/1.1. Browsers had to make educated guesses about which resources to load first.No server push: In HTTP/1.1, the server can't proactively push resources to the client; the client has to request them individually.

**HTTP/2**

HTTP/2, introduced in 2015, was developed to address the limitations of HTTP/1.1 and improve web performance. Here are some key differences and advantages of HTTP/2

Multiplexing: HTTP/2 introduced multiplexing, allowing multiple requests and responses to be sent and received concurrently over a single connection. This eliminates head-of-line blocking issues and speeds up page loading.Header compression: HTTP/2 uses HPACK to compress headers, reducing the overhead associated with header data, which was a major drawback of HTTP/1.1.Request prioritization: HTTP/2 allows for the prioritization of resources, ensuring that more important assets are loaded first. This improves the perceived page loading speed.Server push: In HTTP/2, servers can proactively push resources to the client before they are requested. This further reduces latency and speeds up page loading, as the server can anticipate the client's needs.Binary framing: HTTP/2 uses binary framing, which is more efficient than the text-based protocol of HTTP/1.1, making it easier to parse and reducing the risk of errors.Connection multiplexing: In HTTP/1.1, multiple connections were needed for parallel requests, while HTTP/2 uses a single connection, reducing the overhead of setting up multiple connections.:

1. **Write a blog about objects and its internal representation in Javascript**

IntroductionIn the world of programming, objects are fundamental data structures used to store and manipulate data. JavaScript, being a versatile and popular language, relies heavily on objects for various tasks. In this blog, we will dive into the internal representation of objects in JavaScript, exploring how they work and some key concepts associated with them.

In JavaScript, an object is a composite data type that can hold various types of data, including properties and methods. Objects are an essential part of the language and are used to represent and model real-world entities, making them a powerful tool for developers.

**Internal Representation of Objects**Under the hood, JavaScript engines use various data structures to implement objects efficiently. One common approach is using a hash table (or dictionary) data structure. The object's properties and methods are stored as key-value pairs in the hash table. This allows for quick access and manipulation of object properties.

**JavaScript provides various ways to create objects**:Object Literal: As shown in the example earlier, you can create objects using object literals.Constructor Functions: You can use constructor functions to create objects. This is a common pattern for creating multiple objects of the same type.Object.create(): You can create objects with a specific prototype using Object.create().Class Syntax (ES6): With the introduction of ES6, JavaScript has class syntax that provides a more structured way to create objects.

**Object Methods and Functions**JavaScript objects can have methods, which are functions associated with the object. These methods can access and manipulate the object's properties and perform various tasks. When a method is called, it often references the object using the this keyword.

1. **codekata practice**

CodeKata is a term often used to describe a set of coding exercises or challenges that help developers improve their programming skills. These exercises typically cover a wide range of topics and can be a great way to practice and learn different programming concepts and algorithms. While I can't provide specific CodeKata exercises, I can guide you on how to create and practice with coding challenges:

**1 Choose a Programming Language**:Select a programming language you are comfortable with or want to learn better. Common choices include Python, Java, C++, JavaScript, etc.**2** **Start with Basic Challenges:**Begin with simpler challenges to get the hang of solving coding problems. Websites like LeetCode, HackerRank, CodeSignal, and Project Euler provide a variety of problems categorized by difficulty.**3 Understand the Problem:**Read and understand the problem statement or prompt carefully. Identify the input and output requirements, constraints, and any specific rules.**4 Plan Your Solution:**Before coding, take some time to plan your solution. This might involve designing an algorithm, pseudocode, or outlining the steps you'll take.**5 Write Code:**Implement your solution in your chosen programming language. Follow best practices and use proper coding style.**6 Test Your Code:**Create test cases to verify that your solution works correctly. Check for edge cases and handle potential errors.**7 Debug and Optimize:**If your code doesn't work as expected, debug and refine your solution. Look for ways to optimize your code if necessary.**8 Submit and Compare:**Submit your solution to the platform you're using and compare it to others' solutions to see different approaches.**9 Learn from Others:**Review other people's solutions to gain insight into different coding styles and more efficient approaches.**10 Keep Practicing:**Regular practice is key to improving your coding skills. Work on a variety of problems to build your problem-solving abilities.**11 Track Progress:**Maintain a log of the problems you've solved and your solutions. This can help you track your progress and identify areas for improvement.**12 Explore Advanced Topics:**As you become more confident, challenge yourself with more complex problems involving data structures, algorithms, and other advanced concepts.

1. **Read about IP address, port, HTTP methods, MAC address**

**IP Address (Internet Protocol Address):**An IP address is a unique numerical label assigned to each device connected to a computer network that uses the Internet Protocol for communication.IP addresses are used to identify and locate devices on a network. They can be either IPv4 (e.g., 192.168.1.1) or IPv6 (e.g., 2001:0db8:85a3:0000:0000:8a2e:0370:7334).

**PORT**

A port is a 16-bit number used to identify a specific process or service running on a device within a network.Ports are used in combination with IP addresses to direct data to the appropriate application or service. For example, web traffic is typically directed to port 80 (HTTP) or port 443 (HTTPS).

**HTTP Methods (Hypertext Transfer Protocol Methods):**HTTP methods are a set of request methods used in the Hypertext Transfer Protocol (HTTP) for interacting with web servers.Common HTTP methods include:GET: Retrieve data from a specified resource.POST: Submit data to be processed to a specified resource.PUT: Update a resource or create a new one if it doesn't exist.DELETE: Remove a specified resource.PATCH: Apply partial modifications to a resource.HEAD: Retrieve headers of a resource without the actual data.OPTIONS: Request information about the communication options for the target resource.

**MAC Address (Media Access Control Address):**A MAC address is a unique identifier assigned to network interfaces, such as network interface cards (NICs) or Wi-Fi adapters, to distinguish devices on a local network.MAC addresses are expressed as a sequence of six pairs of hexadecimal digits, separated by colons or hyphens (e.g., 00:1A:2B:3C:4D:5E).Unlike IP addresses, MAC addresses are typically hardwired into the network hardware and remain constant for a device's lifetime.MAC addresses are used at the data link layer of the OSI model to facilitate local network communication.