1. What is HTML?

HTML stands for **HyperText Markup Language**. It's the basic building blocks for making websites. Imagine you're writing a story, but instead of using regular words, you use special codes to tell the computer how to show your story on a webpage.

HTML uses tags, which are like little instructions, to tell the web browser what to do. For example, you use a "<p>" tag to start a new paragraph, and you use "<h1>" to make a big heading at the top of your page.

Here's a simple example:

```
HTML
<!DOCTYPE html>
<html>
<head>
   <title>My First Webpage</title>
</head>
<body>
   <h1>Welcome to My Webpage</h1>
   <p>This is a paragraph of text.</p>
   <p>And here's another paragraph.</p>
</body>
</html>
```

In this code, you're telling the browser that it's an HTML document, giving it a title, and then adding some content like headings and paragraphs. When you open this code in a web browser, it will display a webpage with the title "My First Webpage" and the text you provided in the paragraphs.

So, **HTML** is like a **language that web browsers understand to create webpages.** It's the foundation of almost every website you visit on the internet.

2. What is CSS?

CSS, which stands for **Cascading Style Sheets,** is like the decorator for a webpage. Think of your webpage as a plain room - that's your HTML content, which is just the walls, doors, and furniture. CSS is what you use to add the cool and stylish touches.

So, while HTML is all about the structure and content (like saying, "Put a chair here" in your room), CSS is about the appearance (like saying, "Paint that chair blue and add some fancy designs").

With CSS, you can change colors, fonts, sizes, and how things are arranged on a webpage. It's like selecting the perfect clothes and arranging them to create a fashionable outfit. CSS ensures your webpage looks impressive and well-organized, just like making sure your room is ready to impress anyone who walks in!

3. Why did Javascript come into the picture?

JavaScript came into the picture because there was a need for more interactivity and dynamic behavior on the web. Here's a simplified explanation:

1. *Static Webpages:*
In the early days of the internet, webpages were mostly static. They displayed text and images, but there was very little interactivity. HTML was used to structure the content,

and CSS was used for styling, but there was no way to make webpages respond to user actions or change without reloading the entire page.

2. Desire for Interactivity:
As the internet evolved, there was a growing desire for webpages to be more interactive, like desktop applications. People wanted to do things like click buttons, submit forms, and see changes on a webpage without having to reload the entire page.

3. JavaScript Emerged:
JavaScript was created to meet this need. It's a programming language that runs in the web browser and allows developers to add interactivity to web pages. With JavaScript, you can create functions and scripts that respond to user clicks, input, and other events. This made it possible to build web applications and websites that feel more like software applications.

4. History of Javascript and Ecmascript?

Imagine that there was once a town called JavaScript. This town had its own language, and everyone in the town spoke it. But there were other towns around that spoke different languages. It was difficult for people from different towns to communicate with each other.

One day, the leaders of JavaScript decided that they wanted to create a standard language so that everyone could communicate with each other, even if they were from different towns. They called this new language ECMAScript.

ECMAScript is like the Esperanto of programming languages. It is a standard language that can be used by different browsers and programming languages. This means that if you write code in ECMAScript, it will work in any browser or programming language that supports ECMAScript.

JavaScript is one of the most popular implementations of ECMAScript. It is the language that is used to add interactivity to web pages. But there are other implementations of ECMAScript as well, such as ActionScript and JScript.

Here is a simple analogy to help you understand the difference between JavaScript and ECMAScript

ECMAScript is the language specification, which is like the car engine. It defines the core features of the language, such as variables, functions, objects, and control flow statements.

JavaScript is a specific implementation of ECMAScript, which is like a car model. It adds its own features and functionality to the core ECMAScript language.

Here is a timeline of the history of JavaScript and ECMAScript:

- 1995: JavaScript is created by Brendan Eich at Netscape.
- 1996: JavaScript is submitted to Ecma International for standardization.
- 1997: The first version of the ECMAScript standard is released.
- 1998: JavaScript is supported by all major browsers.
- 1999: ECMAScript 3 is released, which adds many new features to the language.
- 2009: ECMAScript 5 is released, which adds even more new features to the language.
- 2015: ECMAScript 6 is released, which is a major revision of the language.
- 2016: ECMAScript 7 is released.
- 2017: ECMAScript 8 is released.
- 2018: ECMAScript 9 is released.
- 2019: ECMAScript 10 is released.
- 2020: ECMAScript 11 is released.
- 2021: ECMAScript 12 is released.
- 2022: ECMAScript 13 is released.

ECMAScript is constantly evolving, and new versions are released every year. This means that JavaScript is always getting better and more powerful.