Getting Started with JavaScript

Why JavaScript?

Earlier, you learned how to create HTML elements and position or style them with CSS. But until now, you could not manipulate HTML elements or make the page dynamic. This is where javascript comes into play. Javascript provides you with broad functionality which you can use to create your page dynamic.
Here are a few reasons why JavaScript is so popular and what it offers over HTML and CSS:

- **Interactivity:** HTML and CSS are static languages that can create beautiful and well-structured web pages, but they are limited in terms of interactivity. JavaScript offers the ability to add interactivity to web pages, which can help engage users and create a more dynamic experience.
- **Dynamic content:** JavaScript allows for creating dynamic content on web pages, which can help make websites more engaging and responsive. For example, JavaScript can be used to create live updating news feeds, interactive maps, or real-time chat applications.
- **Event handling:** JavaScript offers powerful event handling capabilities, which allow web developers to respond to user interactions and other events on the web page. This can be used to trigger animations, update content, or perform other actions in response to user input.
- **Third-party libraries and frameworks:** JavaScript has a large and vibrant ecosystem of third-party libraries and frameworks, which can help developers build applications more efficiently. These libraries and frameworks can provide pre-built components, tools, and utilities that can save time and effort.

What is Javascript?

Javascript is a High Level, Light-weighted, Interpreted JIT compiled Multi-Paradigm Prototype Based Synchronous, Single Threaded Dynamic Language.
To understand the above definition, you should know what each term signifies within this context.
Here is a definition of each term used above:

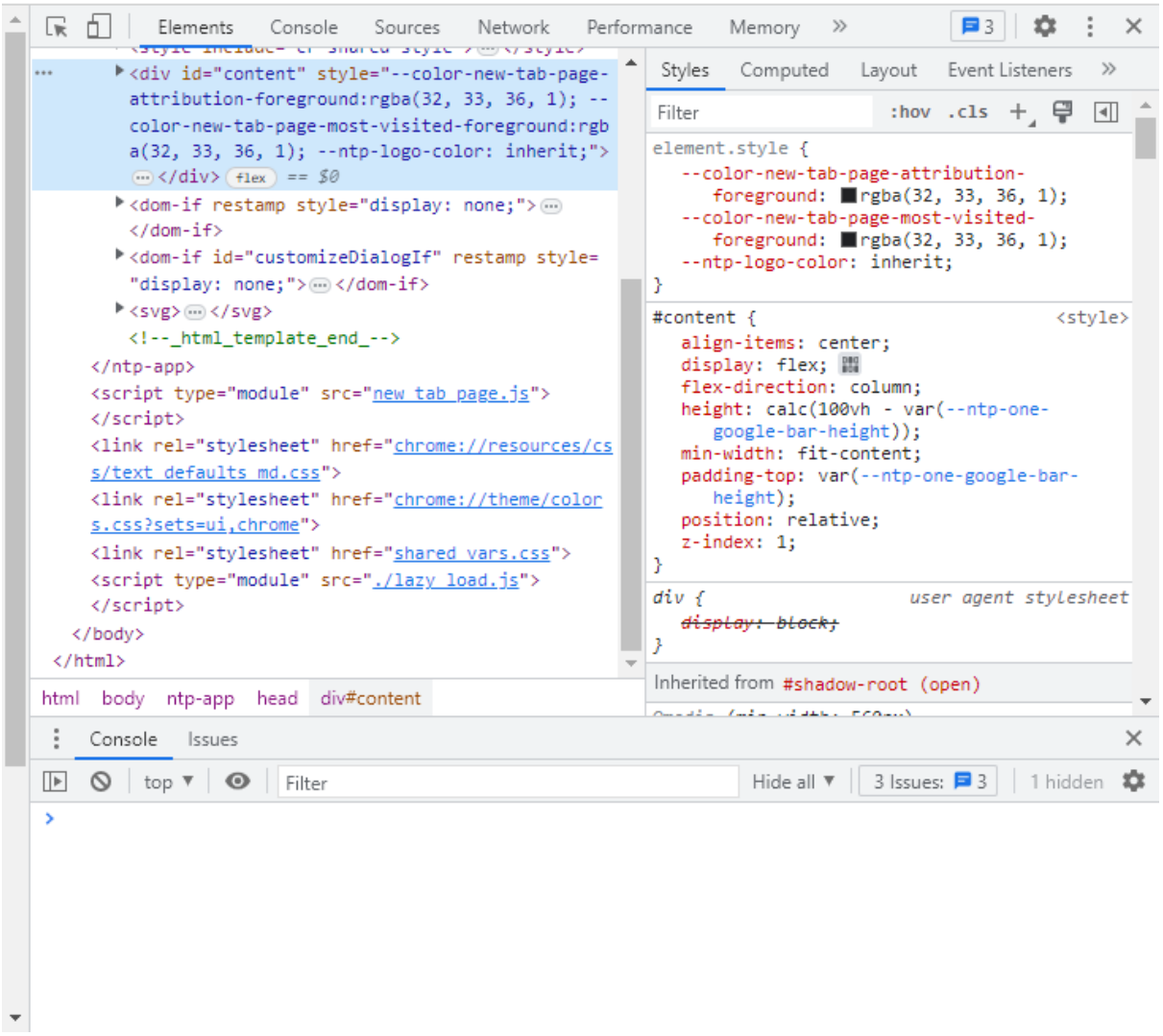
- **High Level**
  - High-level languages are those languages that are independent of system architecture.
- **Light-weighted**
  - The lightweight application is a computer program designed to have low system resource usage.
- **Interpreted or JIT-compiled**
  - An interpreter in the browser reads over the JavaScript to JS code, interprets each line, and runs it. More modern browsers use a JIT compilation which compiles JS to executable bytecode just as it is about to run.
- **Multi-Paradigm**
  - It supports different styles of writing the same code.
- **Prototype Based**
  - This type of style allows the creation of an object without first defining its class.
- **Synchronous**
  - Synchronous means the code runs in a particular sequence of instructions as given in the program.
- **Single-Threaded**
  - Single-threaded means it executes one line of code at a time. So it executes the current line of code before going to the next line.
- **Dynamic Language**
  - Dynamic refers to the value which is capable of change at run time.

History of JS

- In Dec 1991, the Internet was invented, and the first web browser created was Mosaic by Marc Andreessen and Eric Bina at the University of Illinois.
- In 1993 a new web browser called “Netscape” was co-founded by Marc Andreessen.
- But until then, only static websites were created and rendered using HTML. There was no responsiveness on the web pages.
- The developer felt the need for dynamic websites which would change with user interaction.
- In 1995 Brendan Eich was hired by Netscape to develop a scripting language for the browsers.
- Within 10 days, Brendan Eich created the first draft of Javascript and named it MOCHA.
- Later it was renamed to LiveScript and finally to Javascript over the years.
- At that time, Java was a hot language, so naming MOCHA to JavaScript was purely a marketing strategy, and JAVA is not related to JavaScript anyhow.
- At this time, to compete with Netscape, Microsoft decided to include scripting technologies in their browser. In 1996 they released Internet Explorer 3 with their own implementation of JavaScript called JScript.
- With the availability of two scripting languages for different browsers, it became a nuisance for developers to learn two languages.
- To solve this problem, Netscape submitted JavaScript to Ecma International to “standardize the syntax and semantics of a general purpose, cross-platform, vendor-neutral scripting language.”
- In 1997 the first version of ECMA 262 or ECMAScript (ES), now commonly known as JavaScript, was released.
- To this date, javaScript has been evolving, and new versions are added to it now and then.

First JavaScript Code

- To start the journey of your JavaScript, you should first understand where you can see the output of your javascript code in the browser and where you can write your first code of js.
- In every browser, you have the feature of developer tools which look something like this.



To open the DevTools or developer tools in your browser, you can follow any of these steps:

- Right-click anywhere on a page and select **Inspect**.

  or

- Click the three-dot button to the right of the address bar and select **More Tools** > **Developer Tools**.

  or

- Using Shortcut
    - For mac : Cmd + Option + **J**
    - For Windows : Ctrl + Shift + **J**

The DevTools provide many features to debug and understand programs. One of the many features of DevTools is the console window. The console displays error messages and other information related to the code running on the page.
You can write the single-line js code in the console window and perform simple operations. Just click on the console button from the devTools menu and start writing your code.

Dialog Boxes in JS

We have covered the following dialog boxes in this video:
i. Prompt box - instructs the browser to display a dialog with an optional message prompting the user to input some text and to wait until the user either submits the text or cancels the dialog.
ii. Alert box - instructs the browser to display a dialog with an optional message and to wait until the user dismisses the dialog.
iii. Confirm box - instructs the browser to display an optional message and wait until the user confirms or cancels the dialog.
These are actually web APIs and not JavaScript functions. It works on the window object, which you will learn about in the upcoming session.

Summarizing it

Let's summarize what we have learned in this Lecture:

- Why JavaScript was needed?
- What JavaScript is?
- History of Javascript.
- Where to write code in a browser
- Dialog Boxes in JS

Some References

- Definition of JavaScript: link
- Different ECMA versions available: link
- Different ways to open DevTools in Chrome: link