**Exception Handling**

Exception handling in JavaScript is a mechanism used to gracefully deal with **unexpected errors** or **exceptional situations** that may occur during the execution of a JavaScript program. These exceptional situations can include things like **invalid input, network errors, or attempting to perform an operation on an undefined variable**. When such errors occur, they can **disrupt the normal flow** of a program and potentially cause it to **crash**.

JavaScript allows you to handle exceptions using try/catch blocks:

The try statement allows you to define a block of code to be **tested for errors** while it is being executed.

The catch statement allows you to define a block of code to be executed, **if an error occurs in the try block**

Throw, and Try...Catch...Finally

The try statement defines a code block to run (to try).

The catch statement defines a code block to handle any error.

The finally statement defines a code block to **run regardless of the result.**

The throw statement defines a custom error.

javascript

try {

// Code that might throw an error

let result = 10 / 0;

console.log(result);

} catch (error) {

console.error("An error occurred: " + error.message);

}

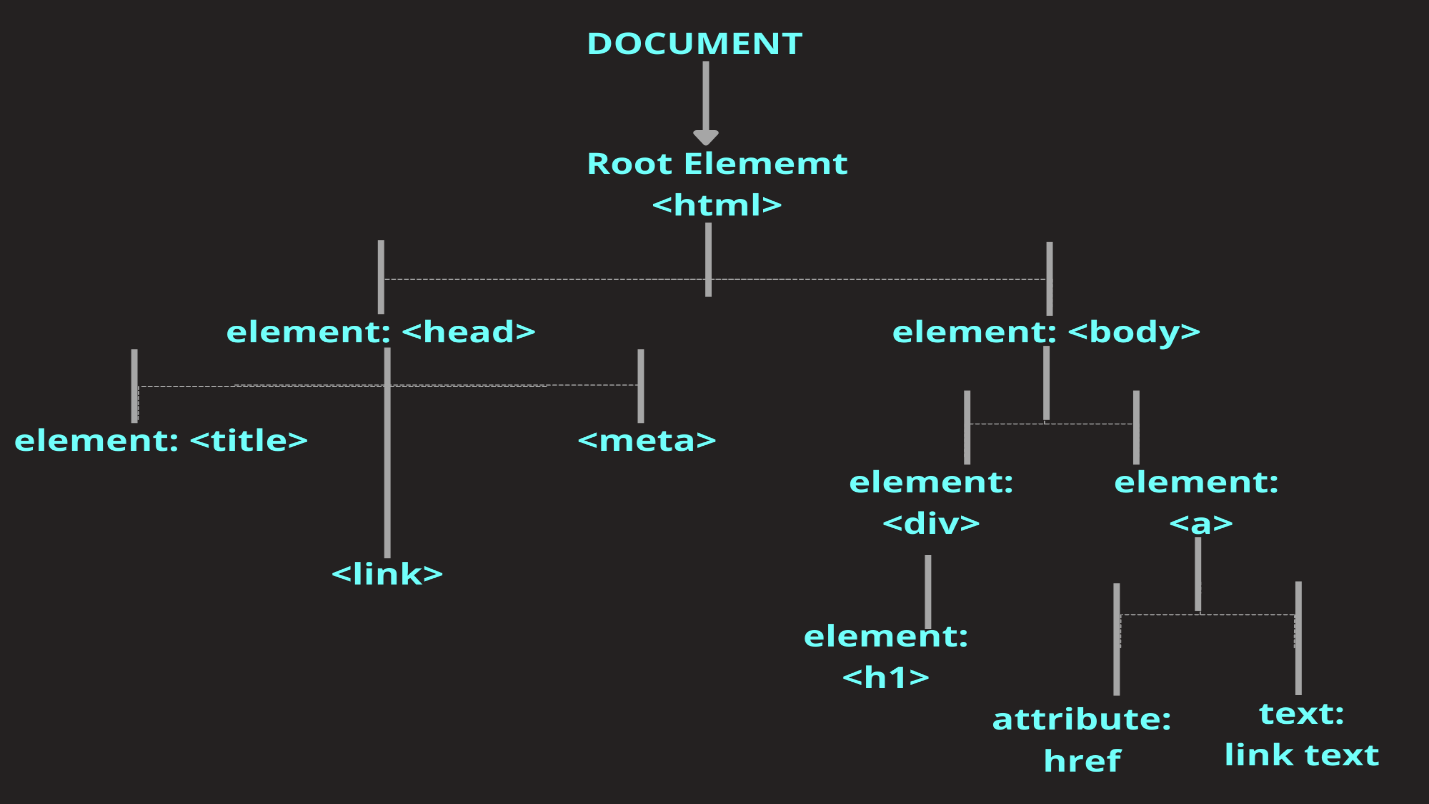
DOM

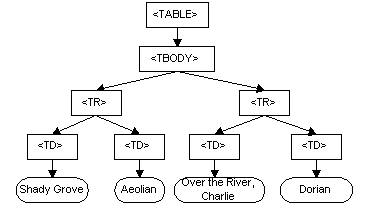
The Document Object Model (DOM) is the **data representation** of the objects that comprise the structure and content of a **document** on the web. This guide will introduce the DOM, look at how the DOM represents an HTML document in memory and how to use APIs to create web content and applications

The DOM is a W3C (World Wide Web Consortium) standard.

The DOM defines a standard for accessing documents:

*"The W3C Document Object Model (DOM) is a platform and* **language-neutral interface that allows programs** *and scripts to dynamically* ***access*** *and* ***update*** *the content, structure, and style of a document."*





<p id="demo"></p>  
  
<script>  
document.getElementById("demo").innerHTML = "Hello World!";  
</script>