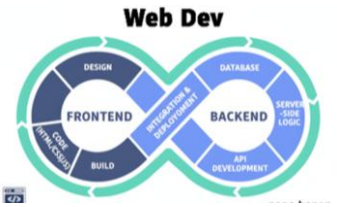
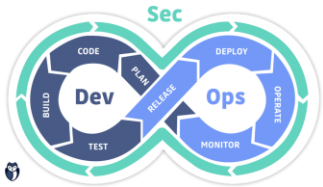




Day 39

"Web Development + Security"

JavaScript try catch & Error Handling:

What is Error Handling in JavaScript?

Error handling means detecting and managing errors gracefully so your program doesn't crash and users see a meaningful message instead of breaking the code. In JavaScript, we mainly use the try...catch block to handle runtime errors (errors that happen while code is running).

Syntax:

try {

// 🟢 Code that may throw an error

} catch (error) {

// 🔴 Code to handle the error

}

Example: a very basic code taking two numbers as input and adding them, without any try..catch

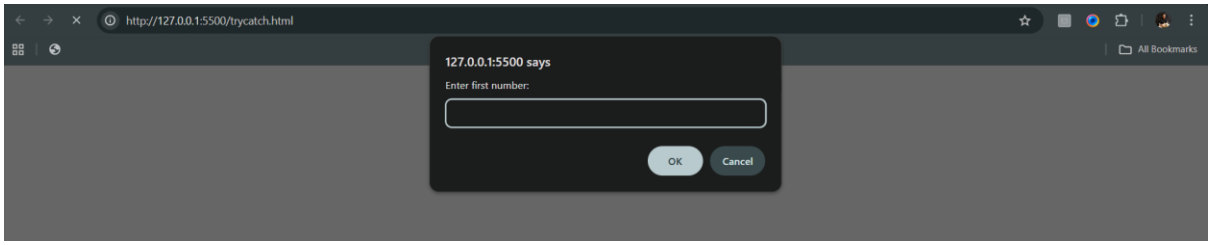
trycatch.html:

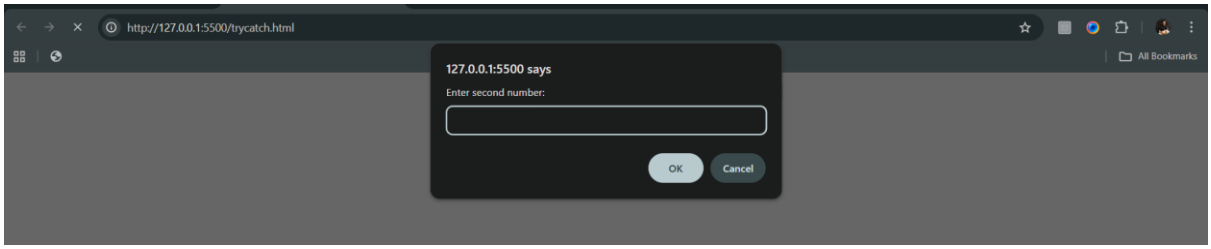
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <script src="trycatch.js"></script>
</body>
</html>
```

Trycatch.js:

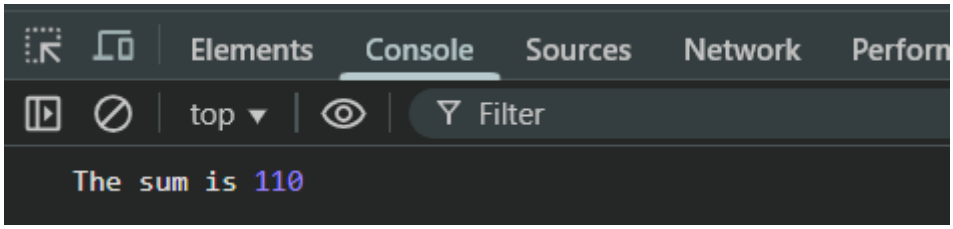
```javascript
let a = prompt("Enter first number:");

let b = prompt("Enter second number:");

let sum = parseInt(a)+parseInt(b);

console.log("The sum is", sum);
```

Output:

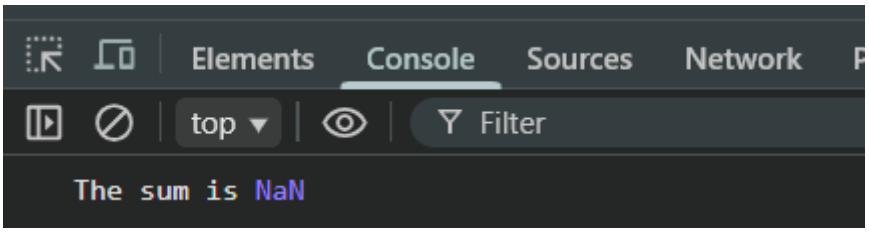




Console:



For above example, we can enter anything in the prompt, such as text as well. Let's try: for this

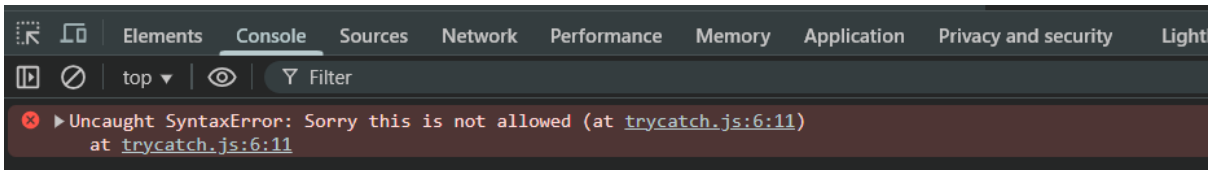


Since, JS has forgiving nature, we will see this. Now, let's say that we don't want this forgiving nature, I mean we want to have check on the input, then we will use try..catch:

Trycatch.js:

```
JS trycatch.js > ...
1   let a = prompt("Enter first number:");
2
3   let b = prompt("Enter second number:");
4
5   if(isNaN(a) || isNaN(b)){
6       throw SyntaxError("Sorry this is not allowed");
7   }
8   let sum = parseInt(a)+parseInt(b);
9
10  console.log("The sum is", sum);
```

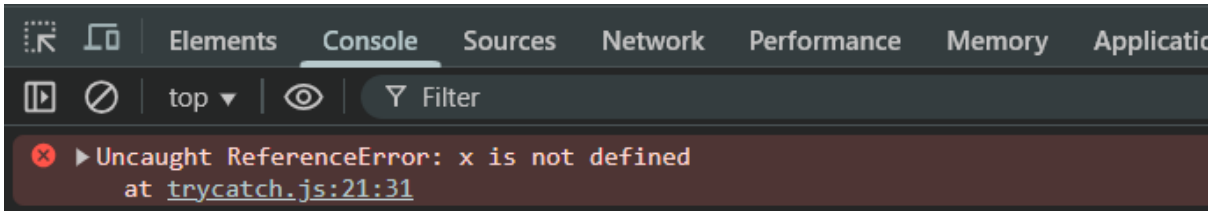
Output: console shows this when we enter alphabets in the appeared prompt.



Now, suppose we have introduced something which we have not defined earlier, then we will get error for sure. For example: we wrote *x at line 21

```
12  let a = prompt("Enter first number:");
13
14  let b = prompt("Enter second number:");
15
16  if(isNaN(a) || isNaN(b)){
17      throw SyntaxError("Sorry this is not allowed");
18  }
19  let sum = parseInt(a)+parseInt(b);
20
21  console.log("The sum is", sum*x);
```

Console:

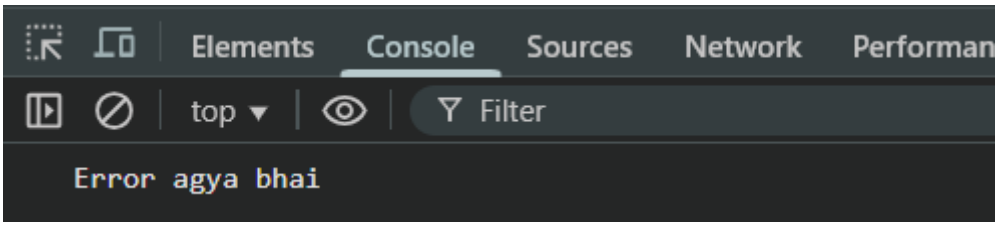


Now, suppose we want that this error should not come, I mean any java built in error should not come, we will handle this using try..catch:

Trycatch.js:

```
let a = prompt("Enter first number:");

let b = prompt("Enter second number:");

if (isNaN(a) || isNaN(b)) {
    throw SyntaxError("Sorry this is not allowed");
}
let sum = parseInt(a) + parseInt(b);

try {
    console.log("The sum is", sum * x);
} catch (error) {
    console.log("Error agya bhai");
}
```

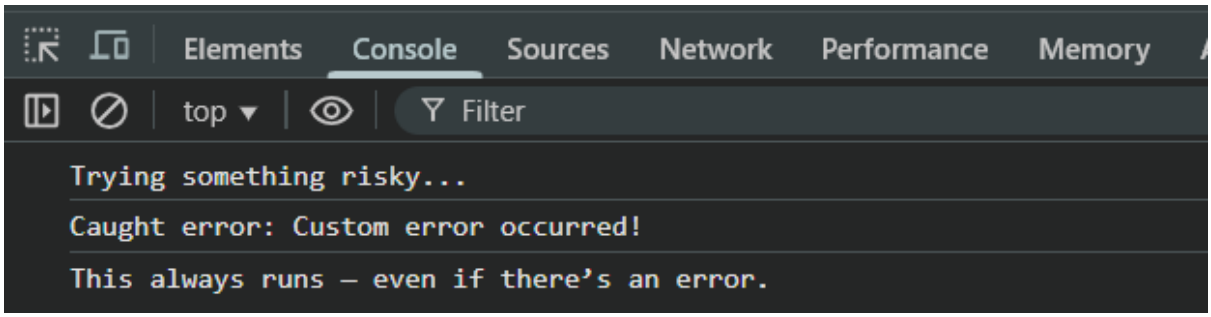
Console: when we enter 2 numbers



Now, introducing finally in the try..catch:

```
//introducing finally in try ..catch
let a = prompt("Enter first number:");

let b = prompt("Enter second number:");
try {
  console.log("Trying something risky...");
  throw new Error("Custom error occurred!");
}
catch (error) {
  console.log("Caught error:", error.message);
}
finally {
  console.log("This always runs — even if there's an error.");
}
```

Console:



--The End--