error handling in JavaScript with try and catch

The JavaScript statements try and catch come in pairs:

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.

```
try {
  Block of code to try
}
catch(err) {
  Block of code to handle errors
}
```

1. First, the code in try {...} is executed.
2. If there were no errors, then catch (err) is ignored: the execution reaches the end of try and goes on, skipping catch.
3. If an error occurs, then the try execution is stopped, and control flows to the beginning of catch (err). The err variable (we can use any name for it) will contain an error object with details about what happened.

```
try {
        alert('Start of try runs'); // (1) <--
        lalala; // error, variable is not defined!
        alert('End of try (never reached)'); // (2)
}
catch (err) {
        alert(` Error has occurred!` ); // (3) <--
}
```

what is a library in software development?

Libraries in programming languages are collections of prewritten code that users can use to optimize tasks. library is not an independently executable unit, but an additional module that is requested by a program.

libraries are collections of classes and functions, which is why they are sometimes also called component or class libraries. With the help of a programme interface (API / Application Programming Interface), access to the necessary functions from the library is made possible. However, access is only limited to the *“public”* functions; libraries also have *“private”* functions that work in the background but remain hidden from the outside world.

Libraries help developers format code more efficiently and quickly than writing it from scratch