The finally Block

This lesson explains ‘finally,’ which is an optional block of the try-catch statement.

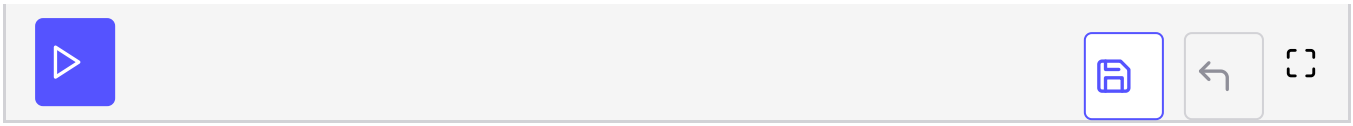
WE'LL COVER THE FOLLOWING

The finally block

`finally` is an optional block of the try-catch statement. It includes expressions that should be executed regardless of whether an exception is thrown or not. To see how finally works, let’s look at a program that throws an exception 50% of the time:

```
import std.stdio;
import std.random;

void throwsHalfTheTime() {
    if (uniform(0, 2) == 1) {
        throw new Exception("the error message");
    }
}

void foo() {
    writeln("the first line of foo()");

    try {
        writeln("the first line of the try block");
        throwsHalfTheTime();
        writeln("the last line of the try block");

    // ... there may be one or more catch blocks here ...

    } finally {
        writeln("the body of the finally block");
    }

    writeln("the last line of foo()");
}

void main() {
```

```
    foo();
}
```



try-catch statement with the finally block

The output of the program is the following when the function does not throw:

```
the first line of foo()
the first line of the try block
the last line of the try block
the body of the finally block
the last line of foo()
```

The output of the program is the following when the function does throw:

```
the first line of foo()
the first line of the try block
the body of the finally block
object.Exception@deneme.d: the error message
```

Although `the last line of the try block` and `the last line of foo()` are not displayed, the content of the `finally` block is still executed when an exception is thrown, as seen above.

When to use the try-catch statement

The try-catch is usually used when thrown exceptions have to be dealt with in a specific manner, for e.g., displaying a specific error message.
Do not catch exceptions otherwise, and leave them to higher-level functions that may want to catch them.

In the next lesson, we will explore the properties of exceptions.