Week 05

Monday, May 17, 2021 12:17 PM

Testing and Debugging

Errors and bugs are a fact of life in programming - they will always be there.

Errors

Errors are caused when something goes wrong in a program. They are usually caused by one of the following:

- System error — there's a problem with the system or external devices with which the program is interacting.
- Programmer error — the program contains incorrect syntax or faulty logic; it could even be as simple as a typo.
- User error — the user has entered data incorrectly, which the program is unable to handle

Programmer errors are our responsibility, so we must ensure they are minimized as much as possible and fixed promptly. We also should try to limit user errors by predicting any possible interactions that may throw an error, and ensure they are dealt with in a way that doesn't negatively affect the user experience

An exception is an error that produces a return value that can then be used by the program to deal with the error. For example, trying to call a method that is nonexistent will result in a reference error that raises an exception.

An exception will also produce a stack trace. This is a sequence of functions or method calls that lead to the point where the error occurred.

Warnings

A warning can occur if there's an error in the code that isn't enough to cause the program to crash.

Strict Mode

ECMAScript 5 introduced a strict mode that produces more exceptions and warnings and prohibits the use of some deprecated features. This is due to the fact that strict mode considers coding practices that were previously accepted as just being "poor style" as actual errors.

To start using strict mode you just have to add

```
'use strict';
```

This will be picked up by any JavaScript engine that uses strict mode. If the engine does not support strict mode, this string will simply be ignored.

Error Objects

- RangeError is thrown when a number is outside an allowable range of values.
- ReferenceError is thrown when a reference is made to an item that doesn't exist. For example, calling a function that hasn't been defined.

- SyntaxError is thrown when there's an error in the code's syntax.
- TypeError is thrown when there's an error in the type of value used; for example, a string is used when a number is expected.
- URIError is thrown when there's a problem encoding or decoding the URI.
- InternalError is a non-standard error that is thrown when an error occurs in the JavaScript engine. A common cause of this is too much recursion.

Tests

Testing is an important part of programming that can often be overlooked. Writing good tests means your code will be less brittle as it develops, and any errors will be identified early on.

Test-driven development(TDD) is the process of writing tests before any actual code. Obviously these tests will initially fail, because there is no code to test. The next step is to write some code to make the tests pass. After this, the code is refactored to make it faster, more readable, and remove any repetition. The code is continually tested at each stage to make sure it continues to work. This process should be followed in small piecemeal chunks every time a new feature is implemented, resulting in the following workflow:

1. Write tests (that initially fail)
2. Write code to pass the tests
3. Refactor the code
4. Test refactored code
5. Write more tests for new features

Jest is a framework to run test, found on page 369.