

JavaScript Profession

Volodya Serobyen



Lesson 1

Errors in Js



Error

Error objects are thrown when runtime errors occur. The Error object can also be used as a base object for user-defined exceptions.

Runtime errors result in new Error objects being created and thrown.



Error Types

An error is an action which is inaccurate or incorrect. There are three types of error in programming which are discussed below :

- Syntax Error
- Logical Error
- Runtime Error



Syntax Error

Syntax error: According to computer science, a syntax error is an error in the syntax of a sequence of characters or tokens that is intended to be written in a particular programming language or it is also the compile-time error if the syntax is not correct then it will give an error message.

An exception caused by the incorrect use of a pre-defined syntax. Syntax errors are detected while compiling or parsing source code.

For example, if you leave off a closing brace (}) when defining a JavaScript function, you trigger a syntax error. Browser development tools display JavaScript and CSS syntax errors in the console.



Logical Error

Logical error: It is the most difficult error to be traced as it is the error on the logical part of the coding or logical error is a bug in a program that causes it to operate incorrectly and terminate abnormally (or crash).

You cannot catch those errors, because it depends on your business requirement what type of logic you want to put in your program.



Runtime Error

Runtime Error: A runtime error is an error that occurs during the running of the program, also known as the exceptions. In the example that is given below the syntax is correct, but at runtime, it is trying to call a method that does not exist.

```
<script type = "text/javascript">  
    <!--  
        window.printme();  
    //-->  
</script>
```

Error

Usually you create an Error object with the intention of raising it using the throw keyword. You can handle the error using the try...catch construct:

```
try {  
    throw new Error( 'Whoops!' )  
} catch (e) {  
    console.error(e.name + ': ' + e.message)  
}
```




Range Error

The RangeError object indicates an error when a value is not in the set or range of allowed values.

A RangeError is thrown when trying to pass a value as an argument to a function that does not allow a range that includes the value.

- attempting to create an array of an illegal length with the Array constructor
- when passing bad values to the numeric methods which is out of range

Range Error Example

```
function check(n)
{
    if( !(n >= -500 && n <= 500) )
    {
        throw new RangeError("The argument must be between -500 and 500.")
    }
}

try
{
    check(2000)
}
catch(error)
{
    if (error instanceof RangeError)
    {
        // Handle the error
    }
}
```

A decorative border on the left side of the slide, consisting of a grid of small white geometric shapes (circles, squares, and triangles) on a dark blue background.

Try Catch block

The try...catch statement marks a try block and a catch block. If the code in the try block throws an exception then the code in the catch block will be executed.

```
try {  
    try_statements  
}catch (exception_var) {  
    catch_statements  
}finally {  
    finally_statements  
}
```

Reference Error

The ReferenceError object represents an error when a variable that doesn't exist (or hasn't yet been initialized) in the current scope is referenced.

```
try {  
    let a = undefinedVariable  
} catch (e) {  
    console.log(e instanceof ReferenceError)  
    console.log(e.message)  
    console.log(e.name)  
    console.log(e.fileName)  
    console.log(e.lineNumber)  
    console.log(e.columnNumber)  
    console.log(e.stack)  
}
```



Syntax Error

The `SyntaxError` object represents an error when trying to interpret syntactically invalid code. It is thrown when the JavaScript engine encounters tokens or token order that does not conform to the syntax of the language when parsing code.

Syntax errors in Javascript cannot be handled by using try-catch blocks as they are thrown while the code is being parsed. The `window.onerror()` function can be used instead to figure out that there is a syntax error.

Syntax Error

```
try {  
    throw new SyntaxError('Hello');  
} catch (e) {  
    console.error(e instanceof SyntaxError);  
    console.error(e.message);  
    console.error(e.name);  
}
```



TypeError

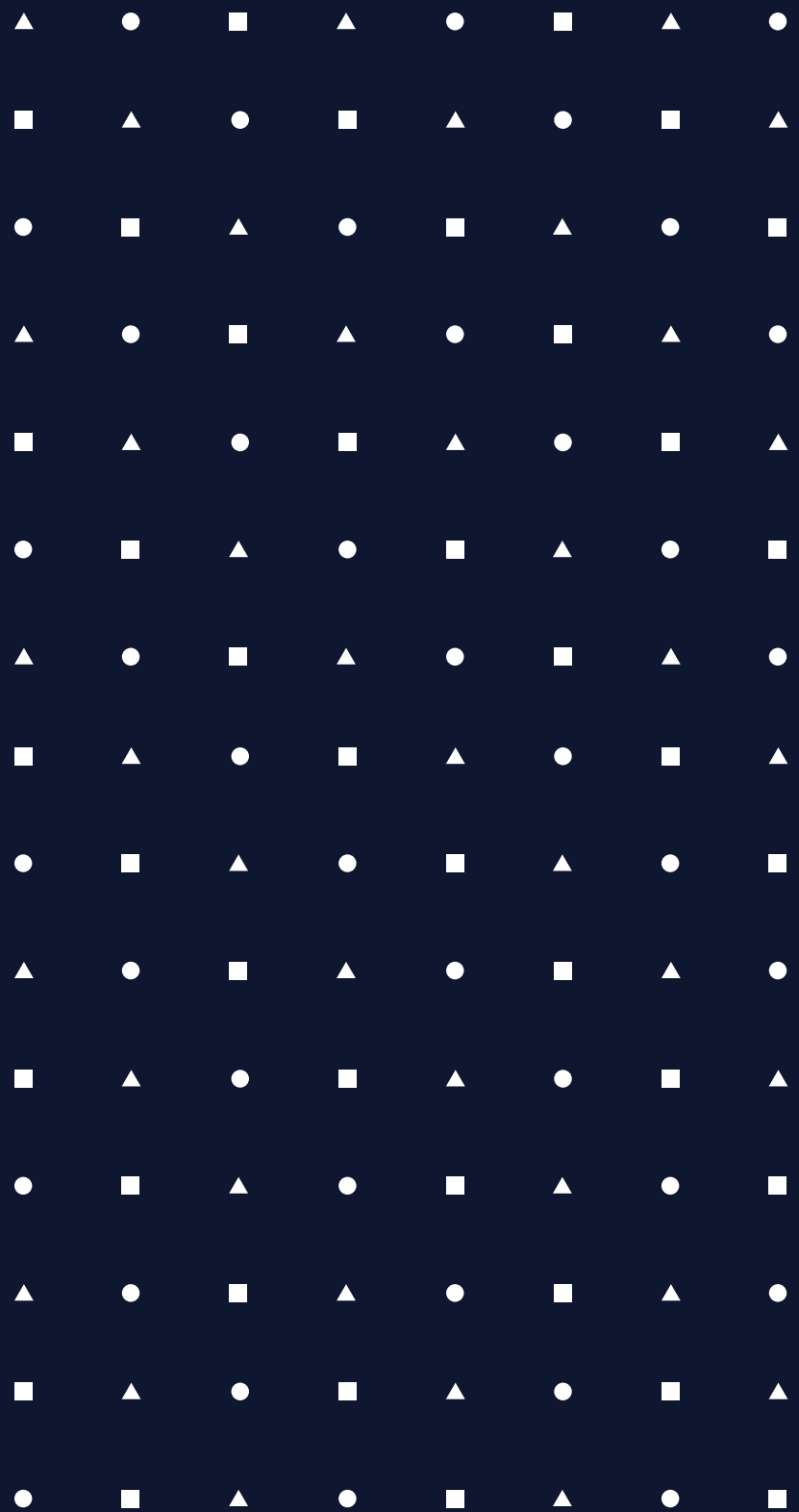
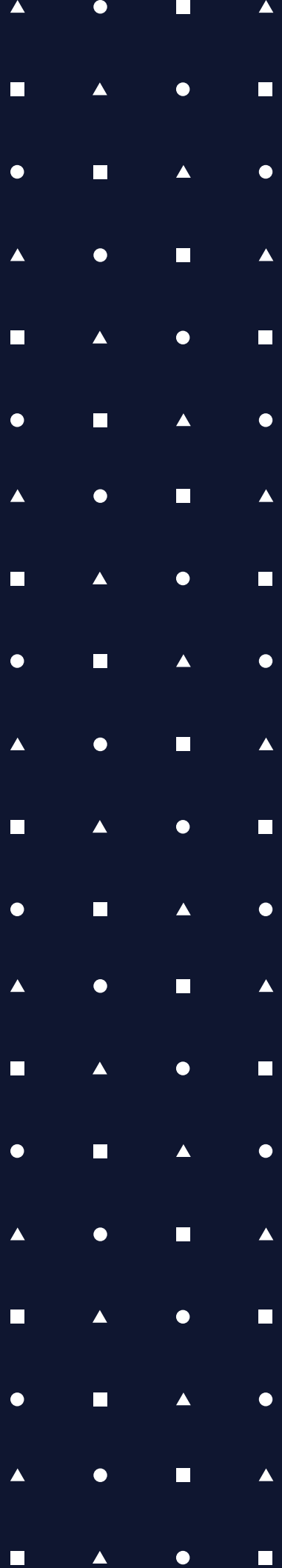
The TypeError object represents an error when an operation could not be performed, typically when a value is not of the expected type.

A TypeError may be thrown when:

- argument passed to a function is incompatible with the type expected by that operator or function
- when attempting to modify a value that cannot be changed
- when attempting to use a value in an inappropriate way.

TypeError

```
try {  
    null.f()  
} catch (e) {  
    console.log(e instanceof TypeError)  
    console.log(e.message)  
    console.log(e.name)  
}
```

Thank you