←

Syntax Errors: Syntax errors occur when the code violates the language's syntax rules. These errors prevent the code from being executed and are typically caught by the compiler or interpreter during the compilation or interpretation process. Common causes of syntax errors include misspelt keywords, missing or misplaced punctuation, and incorrect indentation.

```
// Syntax Error: Missing closing parenthesis
function sum(a, b { return a + b }
```

Resolving Syntax Errors: To fix syntax errors, carefully review the code and identify the specific line or lines where the error is occurring. Check for any obvious mistakes, such as missing semicolons, unmatched parentheses, or incorrect function definitions. The error message provided by an interpreter will indicate the location and nature of the syntax error, helping you pinpoint and resolve the issue.

Type Errors: Type errors occur when operations are performed on incompatible data types or when unsupported actions are attempted. For example, trying to invoke a variable as a function. These errors typically occur during runtime and can cause the program to crash or produce unexpected results.

```
// Type Error: a is not a function
let a = 1
console.log(a())
```

Resolving Type Errors: To address type errors, carefully examine the code where the error occurs and identify the variables or expressions involved. Ensure that the data types of these entities match the expected types for the operation, and that correct operations are performed.

Reference Errors: Reference errors occur when a program tries to access a variable or function that is not defined or out of scope. These errors often happen when a variable is referenced before it is declared or when the scope of a variable is not correctly understood. Reference errors can lead to unexpected behaviour or program crashes.

```
// Reference Error: Accessing an undefined variable
console.log(x);
```

Resolving Reference Errors: To resolve reference errors, carefully review the code and identify the location where the error is triggered. Check for any misspelt variable names, misplaced function calls, or undefined variables. Ensure that variables are declared before they are used and that their scope aligns with the intended access points.