Decoding

Error

Messages

Understanding and Fixing Common Error Messages

1 Syntax Errors

A syntax error means there’s a mistake in the structure of your code, such as a missing character or misplaced operator.

Code Example:

```
const greet = (name => {
  console.log("Hello, " + name);
}; // SyntaxError: Unexpected token '>'
```

Fix:

```
const greet = (name) => {
  console.log("Hello, " + name);
}; // Corrected syntax
```

2

Reference Errors

Reference error occurs when your code attempts to use a variable that has not been declared or is out of scope.

Code Example:

```
function printValue() {
  console.log(myVar); // ReferenceError:
myVar is not defined
}
printValue();
```

Fix:

```
function printValue() {
  let myVar = 'Hello!';
  console.log(myVar);
}
printValue();
```

3 Type Errors

Type errors happen when you perform operations on a variable of an unexpected type, such as trying to access a property on undefined or null.

Code Example:

```
function getLength(arr) {
  return arr.length; // TypeError: Cannot
read property 'length' of undefined
}
getLength();
```

Fix:

```
function getLength(arr) {
  if (Array.isArray(arr)) {
    return arr.length;
  }
  return 0; // Default to 0 if not an
array
}
getLength([]);
```

4 Compilation Error

Compilation errors occur when your code cannot be compiled due to syntax issues or misconfigurations in the build process.

Code Example:

```
// ES6+ syntax in an environment that does
not support modules
import { readFile } from 'fs'; //
SyntaxError: Unexpected token 'import'
```

Fix:

```
// CommonJS syntax compatible with
environments that do not support ES6
modules
const { readFile } = require('fs');
```

5 Range Errors

Range errors occur when a value is out of the permissible range, such as specifying a negative number for array length.

Code Example:

```
let arr = new Array(-1); // RangeError:
Invalid array length
```

Explanation: Array lengths must be non-negative integers.

Fix:

```
let arr = new Array(10); // Valid array
length
```

6 Network Errors

Network errors occur when a network request fails due to issues such as incorrect URLs or server problems.

Code Example:

```
fetch('https://api.example.com/data')
  .then(response => response.json())
  .then(data => console.log(data))
  .catch(error =>
console.error('NetworkError:', error));
```

Fix: Ensure the URL is correct and the server is operational. Handle errors gracefully.

7

Deprication Warnings

Deprecation warnings indicate that a feature or API is outdated and will be removed in future releases.

Code Example:

```
const os = require('os');
console.log(os.tmpdir()); //
DeprecationWarning: 'os.tmpdir' is
deprecated
```

Fix:

```
const { tmpdir } = require('os');
console.log(tmpdir());
```

Follow

For more insights and tips to **Debug like a Pro!**

Muhammad Ishaq