



NodeJS

LABA

EXCEPTION HANDLING FEATURES OF AUTO SEMICOLON INSERTION AND TEMPLATE LITERALS

EXCEPTION HANDLING



```
try {  
  // lines of code that may cause an error  
} catch (e) {  
  // error handling  
}
```



```
try {  
  // ...  
} catch {  
  // ...  
} finally {  
  // ...  
}
```

SELF THROWING EXCEPTIONS



```
const value = "error!";  
throw value;  
// Uncaught error!
```


SEMICOLONS

Automatic Semicolon Insertion (ASI) system is the system where javascript parser automatically adds semicolons when parsing the program in the following situations:

- * When the next line starts with code that breaks the current.
- * When the next line starts with a } which closes the block.
- * When the end of the file is reached.
- * In a line with a return, break, throw or continue statement

QUOTES

There are three types of quotes used in javascript. Single quotes, double quotes and backticks



```
// Single quotes usage
const test1 = 'test';
const test2 = 'te\'st';
const test3 = 'te"st';

// Double quotes usage
const test4 = "te\'st";
const test5 = "te'st";
const test6 = "test";

// Backticks usage
const test7 = `test`;
const test8 = `te"s't`;
const test9 = `tes${variable}`;
```

BACKTICKS

Backticks can be used to represent multiline strings:



```
const multiline = 'A string\nnon multiple lines';  
const multiline2 = `A string  
on multiple lines`;
```


BACKTICKS

Or to interpolate values within a string. Giving access to formatted variables!



```
const name = "Jhon";  
const interpolated = `Hello, ${jhon}`;
```


TEMPLATE LITERALS

Template literals have the following features:

- * They support multiline text
- * They make it possible to interpolate strings, allowing you to write expressions in them.
- * They allow you to work with tagged templates, making it easier to create your own domain specific languages (DSL).

This presentation is property of Solvd, Inc. It is intended for internal use only and may not be copied, distributed, or disclosed.