JS Debug Console

By default, the debug console in the Preview window is hidden. To change its visibility, use the following controls:

- **Expand**: opens the console
- **Collapse**: hides the console

The JavaScript node includes a built-in `Console` object. This global object implements the basic console functionality.

Console methods

When the node runs, a console message is written to the server log if its type matches the server's configured logging level— `error` , `warning` , or `info` .

Logging level `error` :

▼ assert

`assert([assertion, msg1 , ..., msgN])`

This function logs an error message to the console if a specified condition is not met.

- `assertion` : a boolean expression. If this expression evaluates to false, the error message is logged to the console.
- `msg1 , ..., msgN` : a list of JavaScript objects to display. Their string representations are concatenated in the order they appear in the list and printed to the console.

If the first parameter is `false` , the function logs an error message. Otherwise, the function does nothing.

▼ error

`error([msg1 , ..., msgN])`

- `msg1 , ..., msgN` : a list of JavaScript objects to display. Their string representations are concatenated in the order they appear in the list and printed to the console.

An error message will be shown.

Logging level `warning` :

▼ warn

```
warn([msg1 , ..., msgN])
```

- `msg1 , ..., msgN` : a list of JavaScript objects to display. Their string representations are concatenated in the order they appear in the list and printed to the console.

Shows a warning message.

Logging level `info` :

▼ info

```
info([msg1 , ..., msgN])
```

- `msg1 , ..., msgN` : a list of JavaScript objects to display. Their string representations are concatenated in the order they appear in the list and printed to the console.

Shows an information message.

▼ log

```
log([msg1 , ..., msgN])
```

- `msg1 , ..., msgN` : a list of JavaScript objects to display. Their string representations are concatenated in the order they appear in the list and printed to the console.

Shows an information message.

Other methods:

▼ clear

```
clear()
```

Clears the debug console in the **Preview** window. No arguments are required.

Examples:

The code for outputting messages to the console:

```
import { OutputTable, InputTables } from "builtIn/Data";

function f() { return "something"; }

console.info("Information");
console.warn("Warning!");
console.error("Error!!!");
console.log(true);
console.log(undefined);
console.log(null);
console.log(1, 2, 3);
console.log(Math);
console.log(OutputTable);
console.log(f);
console.log(OutputTable.Append);
console.log(new RegExp("a+"));
console.log(new Error("error"));
console.log(new String("Str"));
console.log(new Boolean(true));
console.log(new Date());
console.log(new ArrayBuffer(10));
console.log(Symbol());
console.log(Symbol("aaa"));
console.log(Object(Symbol()));
console.log(new Proxy({}, {}));
console.log([1, InputTables, /^[0..9]+$/]);
console.assert(0 == 0);
console.assert(1 == 0);
console.assert(1 == 0, "1 != 0");
console.assert(1 == 0, "lie", "1 != 0");
```

Outputting to the console:

```
Information
Warning!
Error!!!
true
undefined
null
1 2 3
[object Math]
[object Object]
function f() { return "something"; }
function Append() { [native code] }
/a+/
Error: error
[object String]
[object Boolean]
Mon Dec 17 2018 18:40:56 GMT+0300 (RTZ 2 (winter))
[object ArrayBuffer]
Symbol()
Symbol(aaa)
[object Symbol]
[object Proxy]
1,[object Object],/^[0..9]+$/
Assertion failed:
Assertion failed: 1 != 0
Assertion failed: lie 1 != 0
```

Read on: JS Full API Description