Importing Calculator Functions

You can use Megaladata Calculator functions in your JavaScript code by importing the `"builtIn/Calc"` module. This provides access to all **Calculator** functions, including those implemented in plugins, with the following exceptions:

- `IF` and `IFF`: Use a standard JavaScript function `if (<condition>) {...}`.
- Data access functions (`Data`, `RowNum`, `RowCount`, `DisplayName`, or `CumulativeSum`): Use the equivalent methods provided by the API, such as `RowCount` or `DisplayName`.

Examples of using the functions:

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

// Module import by the require function
const calcModule = require("builtIn/Calc");
OutputTable.Append();
let currentDate = new Date();
OutputTable.Set(0, calcModule.AddWeek(currentDate, 1));
OutputTable.Set(1, calcModule.AddQuarter(currentDate, -1));

// Import using the default value
import calcDefault from "builtIn/Calc";
OutputTable.Append();
OutputTable.Set(0, calcDefault.Val("1"));
OutputTable.Set(1, calcDefault.Str(1e6));

// Importing entire module's content
import * as calc from "builtIn/Calc";
OutputTable.Append();
OutputTable.Set(0, calc.Count("AAA"));
OutputTable.Set(1, calc.Lower("AAA"));

// Importing specific functions
import { RegExMatchCount, RegExMatchedExp } from "builtIn/Calc";
OutputTable.Append();
OutputTable.Set(0, RegExMatchCount("a+?", "aaa"));
OutputTable.Set(1, RegExMatchedExp("a+", "aaa"));

// Dynamic import
import("builtIn/Calc").then(calc => {
    OutputTable.Append();
    OutputTable.Set(0, calc.Repeat(1, 10));
});
```

Read on: Fetch API