JavaScript Specifics

General Specifics

The following general specifics apply to JavaScript for the Designer:

- In the script, input and output parameters are known by their name.
- If you want your function to return a value
    - add the return parameter to the operation signature
    - use a **return** statement for the value you want to return

> ⓘ Please note that nothing will prevent you from returning values without having the return parameter defined on the operation.
>
> - If you defined a **return** parameter but didn't use a `return` statement in your script, the return value is NULL.
> - If you did not define a **return** parameter but exited the JavaScript with `return <something>`, the value of the **return** parameter can not be accessed by the operation caller.

- Every unassigned xUML object property is set to **`NULL`**.
- The xUML object properties defined in the model cannot be deleted.
- The JavaScript code will be executed in strict mode ("use strict"), using the **V8** engine.

Limitations

- There are some limitations to array handling due to the ECMA standard. This is discussed on JavaScript Type Mapping > Arrays.
- You cannot call operations that are defined on Designer classes from a JavaScript. This applies to both static and non-static operations.

Exception Handling

Thrown exceptions will be propagated outside the operation as standard Runtime exceptions. All JavaScript exceptions will be reported to the Designer as `SCRIPTSM/1`. They can be caught using the JavaScript statement `try{} catch(){}`. You can throw a user-defined exception using the `throw` statement.

Related Pages:

- Working with the JavaScript Editor
- JavaScript Specifics
- this Context