# Creating a template

You can find several template examples in the exampleTemplates folder on generating different types of placeholders.

For simple placeholders:

### **INS (=) (Insert)**

Inserts the result of a given JavaScript snippet:

+++INS project.name+++ (+++INS project.details.year+++)

or...

+++INS `${project.name} (${$details.year})`+++

Note that the last evaluated expression is inserted into the document, so you can include more complex code if you wish:

+++INS

const a = Math.random();

const b = Math.round((a - 0.5) \* 20);

`A number between -10 and 10: ${b}.`

+++

You can also use this shorthand notation:

+++= project.name+++ (+++= project.details.year+++)

+++= `${project.name} (${$details.year})`+++

You can also access functions in the additionalJsContext parameter to createReport(), which may even return a Promise. The resolved value of the Promise will be inserted in the document.

Use JavaScript's ternary operator to implement an if-else structure:

+++= $details.year != null ? `(${$details.year})` : ''+++

### **EXEC (!)**

Executes a given JavaScript snippet, just like INS or =, but doesn't insert anything in the document. You can use EXEC, for example, to define functions or constants before using them elsewhere in your template.

+++EXEC

const myFun = () => Math.random();

const MY\_CONSTANT = 3;

+++

+++! const ANOTHER\_CONSTANT = 5; +++

### **IMAGE**

Includes an image with the data resulting from evaluating a JavaScript snippet:

+++IMAGE ({ width: 3, height: 3, path: './sample.png' })+++

+++IMAGE ({ width: 3, height: 3, path: './sample.jpg' })+++

+++IMAGE ({ width: 3, height: 3, path: './sample.jpeg' })+++

+++IMAGE ({ width: 3, height: 3, path: './sample.gif' })+++

QR codes can be generated from links provided

+++IMAGE qrCode(project.url)+++

Image attributes:

* width in cm
* height in cm
* path *[optional]* (in Node only): path to the image to be embedded (absolute or relative to the current working directory)
* data *[optional]*: either an ArrayBuffer or a base64 string with the image data
* extension *[optional]*: e.g. .png

Either specify the path or data + extension.

### **LINK**

Includes a hyperlink with the data resulting from evaluating a JavaScript snippet:

+++LINK ({ url: project.url, label: project.name })+++

If the label is not specified, the URL is used as a label.

### **FOR and END-FOR**

Loop over a group of elements (resulting from the evaluation of a JavaScript expression):

+++FOR person IN project.people+++

+++INS $person.name+++ (since +++INS $person.since+++)

+++END-FOR person+++

Since JavaScript expressions are supported, you can for example filter the loop domain:

+++FOR person IN project.people.filter(person => person.since > 2013)+++

...

FOR loops also work over table rows:

----------------------------------------------------------

| Name | Since |

----------------------------------------------------------

| +++FOR person IN | |

| project.people+++ | |

----------------------------------------------------------

| +++INS $person.name+++ | +++INS $person.since+++ |

----------------------------------------------------------

| +++END-FOR person+++ | |

----------------------------------------------------------

Finally, you can nest loops (this example assumes a different data set):

+++FOR company IN companies+++

+++INS $company.name+++

+++FOR person IN $company.people+++

\* +++INS $person.firstName+++

+++FOR project IN $person.projects+++

- +++INS $project.name+++

+++END-FOR project+++

+++END-FOR person+++

+++END-FOR company+++

### **IF and END-IF**

Include contents conditionally (depending on the evaluation of a JavaScript expression):

+++IF person.name === 'Guillermo'+++

+++= person.fullName +++

+++END-IF+++

Similarly to the FOR command, it also works over table rows. You can also nest IF commands and mix & match IF and FOR commands. In fact, for the technically inclined: the IF command is implemented as a FOR command with 1 or 0 iterations, depending on the expression value.