

Lua [placeholders] Example*

Erik Nijenhuis <erik@xerdi.com>

January 12, 2024

Basics

Wrong parameter:

```
\param{non existing}  $\implies$  [ $\langle$ unknown $\rangle$  non existing]
```

Conditional Parameter:

```
\hasparam{list example}{is set}{is not set}  $\implies$  is not set
```

Before values loaded

Boolean example:

```
\param{bool example}  $\implies$  false
```

```
\ifparam{bool example}{TRUE}{FALSE}  $\implies$  FALSE
```

String example:

```
\param{string example}  $\implies$  “[A string]”
```

Number example:

```
\param{number example}  $\implies$  -1
```

List example:

```
\param{list example}  $\implies$  [A string, A second string]
```

*This example corresponds to lua-placeholders version 0.1.0 written on 2024-01-12.

```

\begin{enumerate}
  \newcommand\formatitem[1]{\item #1}
  \forlistitem{list example}{\formatitem}
\end{enumerate}

```

⇒

1. [A string]
2. [A second string]

Object example:

```

\paramfield{object example}{name}
\paramfield{object example}{email}
⇒ [Your name] [Your email]

```

```

\newcommand\name{...}
\begin{paramobject}{object example}
  \name \email
\end{paramobject}
% And here it works again
\name

```

⇒ [Your name] [Your email] ...

Table example:

```

\newcommand\formatrow{\description & \price \\}%
\begin{tabular}{l | l}
  \textbf{Description} & \textbf{Price} \\ \hline
  \fortablerow{table example}{\formatrow}
\end{tabular}

```

⇒

Description	Price
[The description]	[The price]

After values loaded

Boolean example:

```
\param{bool example} ⇒ true
```

```
\ifparam{bool example}{TRUE}{FALSE} ⇒ TRUE
```

String example:

`\param{string example}` \Rightarrow “PeelInc.”

Number example:

`\param{number example}` \Rightarrow 1

List example:

`\param{list example}` \Rightarrow Tomatoes, Potatoes

```
\begin{enumerate}
  \newcommand\formatitem[1]{\item #1}
  \forlistitem{list example}{\formatitem}
\end{enumerate}
```

\Rightarrow

1. Tomatoes
2. Potatoes

Object example:

```
\paramfield{object example}{name}
\paramfield{object example}{email}
 $\Rightarrow$  John Doe j.doe@example.com
```

```
\newcommand\name{...}
\begin{paramobject}{object example}
  \name \email
\end{paramobject}
% And here it works again
\name
```

\Rightarrow John Doe j.doe@example.com ...

Table example:

```
\newcommand\formatrow{\description & \price \\\%
\begin{tabular}{l | l}
  \textbf{Description} & \textbf{Price} \\\hline
  \fortablerow{table example}{\formatrow}
\end{tabular}
```

\Rightarrow

Description	Price
Peeling tomatoes	50
Peeling potatoes	25

Specification File

```
1 bool example:
2   type: bool
3   default: false
4 string example:
5   type: string
6   placeholder: A string
7 number example:
8   type: number
9   default: -1
10 list example:
11   type: list
12   item type: string
13   default:
14     - A string
15     - A second string
16 object example:
17   type: object
18   fields:
19     name:
20       type: string
21       placeholder: Your name
22     email:
23       type: string
24       placeholder: Your email
25 table example:
26   type: table
27   columns:
28     description:
29       type: string
30       placeholder: The description
31     price:
32       type: number
33       placeholder: The price
```

Listing 1: example-specification.yaml

Payload File

```
1 bool example: true
2 string example: PeelInc.
3 number example: 1
4 list example:
5   - Tomatoes
6   - Potatoes
7 object example:
8   name: John Doe
9   email: j.doe@example.com
10 table example:
11   - description: Peeling tomatoes
12     price: 50.00
13   - description: Peeling potatoes
14     price: 25.00
```

Listing 2: example.yaml

Fallback Payload

```
1 {
2   "bool example": true,
3   "string example": "PeelInc.",
4   "number example": 1,
5   "list example": ["Tomatoes", "Potatoes"],
6   "object example": {
7     "name": "John Doe",
8     "email": "j.doe@example.com"
9   },
10  "table example": [
11    {
12      "description": "Peeling tomatoes",
13      "price": 50.00
14    }, {
15      "description": "Peeling potatoes",
16      "price": 25.00
17    }
18  ]
19 }
```

Listing 3: example.json