

# NanoUI Template Reference

Zaers

June 24, 2015

## Contents

<b>1</b>	<b>Syntax</b>	<b>1</b>
1.1	Displaying a variable . . . . .	1
1.2	Calling a function . . . . .	1
1.3	Conditionals . . . . .	1
1.4	Progress bars . . . . .	1
1.5	Links . . . . .	1
1.6	For loops . . . . .	1
1.7	Data linking . . . . .	1
<b>2</b>	<b>Other resources</b>	<b>2</b>

# 1 Syntax

The basic syntax of the templates could be described as HTML with JSrender extensions for displaying dynamic data.

The CSS classes and icons used in NanoUI are defined in `/nano/css/*.css`  
JSrender functionality is called between `{{ }}`.

The variables used are defined in the game code for each particular object that use nanoUI

## 1.1 Displaying a variable

Variables are displayed via `{{:variable}}`

## 1.2 Calling a function

Functions are called using `{{~function(ARGS)}}`

## 1.3 Conditionals

JSrender conditionals are used via

```
{{if condition}}  
and  
{{else optional contidition}}  
terminated with  
{{/if}}
```

## 1.4 Progress bars

Progress bars are used as a function via

```
displayBar(variable, minimum, maximum, styleClass, text)
```

## 1.5 Links

BYOND links are generated as a function via

```
link(text, icon, parameters, status, elementClass, elementId)
```

## 1.6 For loops

JSrender supports for loops, used as

```
{{for object or array}}  
...  
{{/for}}
```

## 1.7 Data linking

JSViews data linking is done via `{{^}}` and automatically refreshes the data displayed if it changes

## 2 Other resources

If the above reference was insufficient, JSrender API documentation is available at <http://www.jsviews.com> and complete templates are viewable in `/nano/templates/*.tpl`