

<b>Status:</b>	Resolved	<b>Priority:</b>	Must have
<b>Author:</b>	Sebastian Kurfuerst	<b>Category:</b>	
<b>Created:</b>	2009-10-19	<b>Assigned To:</b>	Sebastian Kurfuerst
<b>Updated:</b>	2010-10-20	<b>Due date:</b>	

**Has patch:**

**Subject:** Improve Shorthand Syntax of ViewHelpers

**Description**

## Proposal for shorthand syntax

The goal of this RFC is to propose a re-done shorthand syntax for fluid after some real-world usage.

### Current situation

Currently, we refer to "shorthand syntax" as another way to call a ViewHelper.

Here, there is a basic example for this:

```
{f:bla.blubb("unnamed argument" a="b" c="42" d="{object}")}
```

This is equivalent to the following:

```
<f:bla.blubb a="b" c="42" d="{object}">unnamed argument</f:bla.blubb>
```

- Every argument needs to be quoted
- The first argument is the data inside the tag body, in case it is unnamed.
- Nesting of ViewHelpers is arbitrarily possible:  

```
{f:outer(" first VH: {f:bla()} second VH: {f:blubb()}")}
```

(also inside arguments of ViewHelpers, other VHs can be called)

- ViewHelpers can **not** be called inside arrays

Note: Right now, the shorthand syntax is functionally equivalent like the Tag-based ViewHelper syntax.

### Problems with current syntax

In the most common use-case, we write things like

```
{f:bla(argument="{blubb}" argument2="{bla}")}
```

There, two things feel strange:

- The "{...}" syntax feels weird
- It feels strange that the argument needs to be quoted
- There is no comma between the arguments, which feels strange to programmers
- Named arguments in many languages are delimited by :, and not by =. The a="b" syntax feels strange.

### Draft for new syntax

We remove the "Shorthand Syntax" as it has been described above, and instead extend the "Object Access" syntax.

The basic object access syntax looks like this:

```
{object} {object.bla.blubb}
```

Now, we start to **post-process** this value with a ViewHelper:

```
{object.bla->f:postProcess()}
```

This could also be chained

```
{object.bla->f:postProcess()->f:anotherPostProcess()}
```

Additionally, if the ViewHelper does NOT work on any object, the first part can be omitted:

```
{f:process()->f:postProcess()}
```

## Arguments

Arguments look like this:

```
{f:bla.blubb(arg1: 'literal string', arg2: 42, arg3: object1.bla, arg4:{a1: "string"})}
```

- All arguments have a name
- The syntax is the same as in JSON arrays, just without the outer brackets
- All string arguments are NOT evaluated, but taken as **literal** strings.
- arg1 is a string (which will not be post-processed)
- arg2 is a number
- arg3 is an object accessor
- arg4 is an array

## Comparison

The new syntax has the following limitations:

- There is no equivalent of the following: {f:process("bla")}, i.e. there is no {"bla"->f:process()} and I would not introduce it.
- Is is NOT possible anymore to use ViewHelpers as arguments in other shorthand ViewHelpers (but I think this is not a problem, as you could use the XML-based syntax where this works)
- It is currently NOT possible to call ViewHelpers in arrays (but this has not been possible before either)

**Please discuss this with us in the TYPO3 5.0 General Mailing list on [lists.typo3.org](http://lists.typo3.org)!**

### Related issues:

related to TYPO3.Fluid - Bug # 5150: Rewrite ViewHelpers to support new inlin...

Resolved

2009-10-27

## Associated revisions

### Revision 9829b9ef - 2009-10-21 11:52 - Sebastian Kurfuerst

[!!!][+FEATURE] Fluid (Core): Removed the old Shorthand Syntax, and added a new chainable syntax for ViewHelpers. ADJUST YOUR TEMPLATES in case you use the {f:...()} syntax! Resolves #5033.

[+TASK] Fluid (Core): The object accessor node now uses FLOW3 reflection service instead of custom implementation. This should be functionally equivalent

[+TASK] Fluid: Improved doc comments

[+TASK] Fluid: Improved test cases by removing some not needed stub classes

## History

### #1 - 2009-10-19 11:39 - Bastian Waidelich

Hi,

obviously I strongly agree to this change ;)

Just a few additional notes to clarify the intention of this RFC even more:

Currently tag- and "shorthand"-syntax are interchangeable - and technically they'll still be. But with the new syntax it gets a lot clearer when to use one or the other:

For tag-based and widget like view helpers, you'd want to use the tag syntax:

```
<f:image src="EXT:myext/images/foo.jpg" width="600m" />
// ...
<my:googleMap controls="{0: 'zoom'}" />
```

If you only want to *modify* a string, the shorthand syntax is more readable:

```
{post.title}
```

->

```
{post.title->f:format.crop(maxChars: 30)}
```

and with the new syntax you'd be able to nest them too:

```
{post.title->f:format.crop(maxChars: 30)->f:format.nl2br()}
```

Obviously, this can get messy too - And it's still your choice to create your custom view helper in this case.

The new syntax has the following limitations: - There is no equivalent of the following: `{f:process("bla")}`, i.e. there is no `{"bla"->f:process()}` and I would not introduce it.

You could still use the alias view helper to get around this:

```
<f:alias map="{foo: 'foo'}">
  {foo->f:process()}
</f:alias>
```

or, even more flexible:

```
{settings.foo->f:process()}
```

Bastian

## #2 - 2009-10-20 10:39 - Sebastian Kurfuerst

- File `Fluid_v5.patch` added

attached is a first patch which implements this.

## #3 - 2009-10-21 12:02 - Sebastian Kurfuerst

- Status changed from *Accepted* to *Resolved*

- % Done changed from 0 to 100

Applied in changeset r3333.

#### #4 - 2009-10-21 14:40 - Bastian Waidelich

Hi @all,

as this is a breaking change (if you have made use of the shorthand syntax in your templates) here some "upgrade" hints:

```
{f:uri.action(action='delete' arguments='{blog: blog}')}
```

now becomes

```
{f:uri.action(action:'delete', arguments:{blog: blog})}
```

--

```
{f:format.crop(blog.title maxChars=30)}
```

gets

```
{blog.title -> f:format.crop(maxChars: 30)}
```

--

There are a few cases, where the new syntax limits the flexibility.

E.g. it was possible to write

```
{f:uri.resource('styles.css' absolute='true')}
```

and

```
{'styles.css' -> f:uri.resource(absolute:'true')}
```

is not possible.

But you can always work around this by using the alias view helper, storing values in the plugin settings or using the tag syntax:

```
<f:uri.resource absolute="true">styles.css</f:uri.resource>
```

We might also rework some of the view helper to use an argument instead of the child nodes.

#### Files

---

Fluid\_v5.patch

45.7 kB

2009-10-20

Sebastian Kurfuerst