TYPO3.Fluid - Feature # 6149

Status:	Resolved	**Priority:**	Should have
Author:	Stefan Froemken	**Category:**	ViewHelpers
Created:	2010-01-21	**Assigned To:**	Bastian Waidelich
Updated:	2010-10-20	**Due date:**	
Has patch:			
Subject:	for: add iteration informations		

Description

This is a feature request. I first saw this functionality in smarty and I think you should insert this in fluid, too. You're right, when you told me to program this viewHelper on my own. But I think this should be in Core.

Here is my first try changing forViewHelper.php:

```
   /**
    * Iterates through elements of $each and renders child nodes
    *
    * @param array $each The array or SplObjectStorage to iterated over
    * @param string $as The name of the iteration variable
    * @param string $key The name of the variable to store the current array key
    * @param boolean $reverse If enabled, the iterator will start with the last element and proceed reversely
    * @param integer $iterate If >= 0, you got a new element {iterate} which counts through the Arrays
    * @param integer $step If iterate is set, then you can set your prefered stepping 2 = 1,3,5,7,9
    * @return string Rendered string
    * @author Sebastian Kurfürst &lt;sebastian@typo3.org&gt;
    * @author Bastian Waidelich &lt;bastian@typo3.org&gt;
    * @author Robert Lemke &lt;robert@typo3.org&gt;
    * @api
      */
     public function render($each, $as, $key = '', $reverse = FALSE, $iterate = FALSE, $step = 1) {
       $output = '';
       if ($each === NULL) {
         return '';
       }
       if (is_object($each)) {
         if (!$each instanceof Traversable) {
           throw new Tx_Fluid_Core_ViewHelper_Exception('ForViewHelper only supports arrays and objects implementing
   Traversable interface' , 1248728393);
         }
         $each = $this->convertToArray($each);
       }
   if ($reverse === TRUE) {
         $each = array_reverse($each);
       }
   $output = '';
       foreach ($each as $keyValue => $singleElement) {
         $this->templateVariableContainer->add($as, $singleElement);
         if ($key !== '') {
           $this->templateVariableContainer->add($key, $keyValue);
         }
         if ($iterate !== FALSE) {
           $this->templateVariableContainer->add('iterate', $iterate);
```

```
                ($step != 1)? $iterate = $iterate + $step: $iterate++;
            }
            $output .= $this->renderChildren();
            $this->templateVariableContainer->remove($as);
            if ($key !== '') {
                $this->templateVariableContainer->remove($key);
            }
            if ($iterate !== FALSE) {
                $this->templateVariableContainer->remove('iterate');
            }
        }
        return $output;
    }
```

Now you can do something like this:

```
<f:for each="{customer.car}" as="car" iterate="2" step="2">
<tr>
<td>Auto {iterate}</td>
<td>{car.type}</td>
</tr>
</f:for>
```

and get:
Auto 1: BMW
Auto 3: Audi
Auto 5: VW

or remove step to get:
Auto 1: BMW
Auto 2: Audi
Auto 3: VW

Related issues:

duplicated by Fluid - ViewHelper - Incubator - Feature # 6459: Loop informati...	**Closed**	**2010-02-13**
duplicated by Fluid - ViewHelper - Incubator - Suggestion # 7654: loop	**Rejected**	**2010-05-05**

Associated revisions

Revision 1c1ca47b - 2010-07-26 13:04 - Bastian Waidelich

[+FEATURE] Fluid (ViewHelpers): Added iteration information to for ViewHelper. Thanks to all the contributors for your input and patches! This resolves #6149

History

#1 - 2010-01-30 21:21 - Franz Koch

I did something simmilar in my custom for viewHelper, only that I provide a itterator array (with customizable name) that is containing some additional information:
- count/total (integer, the total count of the array items)
- isFirst (boolean, is first item in cycle)
- isLast (boolean, is last item in cycle)
- cycle (integer, the current cycle count, 1 based)
- index (integer, the current cycle index/count, 0 based)

- isOdd
- isEven

So in your templates you can use:
```
<f:for each="{objects}" as="{object}">
 <f:if condition="{itterator.isFirst}">do something</f:if>
 <f:if condition="{itterator.isLast}">do something different</f:if> {object.whatever}
</f:for>
```

#2 - 2010-03-11 10:13 - Robert Lemke

- Target version changed from 1.0.0 alpha 7 to 1.0.0 alpha 8

#3 - 2010-07-04 00:09 - Christian Müller

In future that could be done with the calculation view helper (just adding two to some template variable for every for cycle).

#4 - 2010-07-07 10:53 - Alexander Stehlik

I think the approach by Franz is very handy and useful and would make many things a lot easier.

But the variable name "iterator" should not be static because this could cause problems if you want to use nested loops.

So maybe the best way to do this, is do define a new property (e.g. "iteration") as proposed in the mvc mailing list http://lists.typo3.org/pipermail/typo3-project-typo3v4mvc/2010-January/002404.html by Martin Kutschker:

```
<f:for each="{foobar}" as="item" iteration="row">
  <f:if condition="{row.isFirst}"> ... </f:if>
</f:for>
```

#5 - 2010-07-07 11:22 - Alexander Stehlik

- File patch_6149.diff added

I attached a patch for this feature. Please let me know if there are any problems or if you have any improvements.

#6 - 2010-07-08 08:51 - Michael Feinbier

I really like to have that feature. It is missing currently and heavily used by Frontend Designer / CSS-Gurus which often need a "last" class in for example the last <li> of an unordered list

#7 - 2010-07-12 12:44 - Bastian Waidelich

- Project changed from TYPO3.Fluid to Fluid - ViewHelper - Incubator

- Category deleted (ViewHelpers)

- Target version deleted (1.0.0 alpha 8)

#8 - 2010-07-12 13:38 - Bastian Waidelich

- File ForViewHelper.diff added

Merging files and comments from #6459 to have everything in one ticket:

> In the past i used the template engine h2o. One thing about the For loop i liked was the special var {loop} which was available in For loop which contains some information about the loop like:
> *first
> *last
> *odd
> *even
> *current
> *length
> http://wiki.github.com/speedmax/h2o-php/built-in-tags
>
> i made some changes to the ForViewHelper which adds this behavior. Would be cool if you could integrate it.

(Marc Neuhaus, 2010-02-13 20:45)

#9 - 2010-07-12 13:43 - Bastian Waidelich

- Subject changed from Iteration in f:for like 1,2,3,4,5 to for: add iteration informations

#10 - 2010-07-12 13:59 - Georg Leitner

Alexander Stehlik wrote:

> *I attached a patch for this feature. Please let me know if there are any problems or if you have any improvements.*

Tested the patch with TYPO3 4.4 (extbase/fluid 1.2.0): works like a charm. This should definitely find it's way into the core. Thanks a lot, Alexander!

#11 - 2010-07-19 13:20 - Franz Koch

I did it just like in Alexander Stehliks patch with one exception: I always add the itterator to the view. If no specific name is given in the viewHelper I build one based on $as:

```
if (!$itterator) {
    $itterator = $as.'Itterator';
}
```

#12 - 2010-07-26 12:34 - Bastian Waidelich

- Project changed from Fluid - ViewHelper - Incubator to TYPO3.Fluid
- Status changed from New to Accepted
- Assigned To set to Bastian Waidelich

#13 - 2010-07-26 12:35 - Bastian Waidelich

- Category set to ViewHelpers
- Branch changed from v4 to v4 + v5

Sorry for moving this around and around. But, good news: We decided that we don't need to go the "incubator way" for this feature. Instead I'll add it to the v4 & v5 core directly. Stay tuned.

#14 - 2010-07-26 13:05 - Bastian Waidelich

- Status changed from Accepted to Resolved

- % Done changed from 0 to 100

Applied in changeset r4904.

#15 - 2010-07-27 12:02 - Bastian Waidelich

FYI: I got some really positive feedback. Apparently this was a heavily missing feature - I promised to forward the credits and here you go: Thanks for your input and patches ;)

Files

patch_6149.diff	2.1 kB	2010-07-07	Alexander Stehlik
ForViewHelper.diff	1.8 kB	2010-07-12	Bastian Waidelich