Core - Feature # 48221

Status:	New	**Priority:**	Must have
Author:	Dave no-lastname-given	**Category:**	Fluid
Created:	2013-05-15	**Assigned To:**	
Updated:	2015-02-15	**Due date:**	
PHP Version:			
Complexity:			
Sprint Focus:			
Subject:	BooleanNode must support logical OR/AND		

Description

Fluid\Core\Parser\SyntaxTree\BooleanNode must support logical OR/AND

It feels very incomplete not being able to do simple OR/AND in a if condition.
I think we need || and && comparators.

<f:if condition="{foo} || {bar}">foobar</f:if>

Even though logic should be kept to a minimal in templates, when you are trying to create custom fluid templates on a pre-existing extension you dont want to be adjusting the extensions original controllers or extend a whole new extension just because you need a simple logical comparison in a template. That just seems insane and complete overkill. These comparators IMHO are a must.

Also the alternative by using nesting to evaluate OR/AND logic via THEN and ELSE is surely more work for the fluid parser than is necessary.

Related issues:

blocked by TYPO3.Fluid - Feature # 33394: Logical expression parser for Boole...	**Needs Feedback**	**2012-01-23**

History

#1 - 2013-09-06 14:45 - Tymoteusz Motylewski

See sample implementation of combining conditions from Claus Due here
https://github.com/NamelessCoder/vhs/blob/master/Classes/ViewHelpers/ConditionViewHelper.php

#2 - 2013-09-06 14:47 - Philipp Gampe

We need to wait for #33394.

#3 - 2014-09-16 11:45 - Alexander Opitz

- Project changed from Fluid to Core

- Category changed from Fluid: Core to Fluid

#4 - 2015-02-15 09:41 - Alexander Schnitzler

@Philipp: After 3 years of waiting the only solution Flow Fluid comes up with is EEL... We should really consider implementing this stuff ourselves. Just my 2 cents. TYPO3 CMS has the much bigger user base and therefore really needs these issues fixed more than flow which can wait for a sophisticated solution somewhen in the future.

#5 - 2015-02-15 11:53 - Dave no-lastname-given

@Alexander: I agree, however there is IMHO a simpler solution to solve up to 99% of all these missing features in fluid and that is to open fluid up to the entire PHP API. PHP **is** a templating language and it is crazy not to be able to access it from fluid. Someone told me that fluid has a design philosophy, up until this point I have never seen or found the forementioned fluid philosophy document. Porting of features from NEOS to TYPO3 CMS also appears to be a one-way street, I also don't understand why that should be, so I see no harm in developing new features for fluid for the CMS and porting them backwards for NEOS because NEOS will suffer from the same problems.