Angular Js v1.3.0 Filters Cheat Sheet

by Roman K. (Roman) via cheatography.com/19465/cs/2483/

currency11

HTML:

```
{{ curren cy_ exp ression | currency : symbo
l}}
```

JS:

```
$filte r(' cur ren cy' )(a mount, symbol)
```

Params:

amount, *number*: Input to filter.

symbol (optional), *string*: Currency symbol or identifier to be displayed.

number11

Formats a number as text.

In HTML Template Binding:

```
{{ number _ex pre ssion | number : fracti onS -
ize}}
```

In JavaScript:

```
$filte r(' num ber ')( number, fracti onSize)
```

Arguments:

number, *number* / *string*: Number to format.

fractionSize (optional), *number* / *string*: Number of decimal places to round the number to. If this is not provided then the fraction size is computed from the current locale's number formatting pattern. In the case of the default locale, it will be 3.

Returns:

string, Number rounded to decimalPlaces and places a "," after each third digit.

date11

Formats date to a string based on the requested format.

HTML:

```
{{ date_e xpr ession | date : format
}}
```

JS:

```
$filte r(' dat e') (date, format)
```

Arguments:

date11 (cont)

date, *Date* / *number* /*string*: Date to format either as Date object, milliseconds (string or number) or various ISO 8601 datetime string formats (e.g. yyyy-MM-ddTHH:mm:ss.sssZ and its shorter versions like yyyy-MM-ddTHH:mmZ, yyyy-MM-dd or yyyyMMddTHHmmssZ). If no timezone is specified in the string input, the time is considered to be in the local timezone.

format (optional), *string*: Formatting rules (see Description). If not specified, mediumDate is used.

Return:

string, Formatted string or the input if input is not recognized as date/millis.

json11

Allows you to convert a JavaScript object into JSON string.

In HTML Template Binding:

```
{{ json_e xpr ession | json}}
```

In JavaScript:

```
$filte r(' jso n') (ob ject)
```

Arguments:

object, * Any JavaScript object (including arrays and primitive types) to filter.

lowercase11

Converts string to lowercase.

In HTML Template Binding

```
{{ lowerc ase _ex pre ssion | lowerc -
ase}}
```

In JavaScript

```
$filte r(' low erc ase')()
```

uppercase11

Converts string to uppercase.

In HTML Template Binding

```
{{ upperc ase _ex pre ssion | upperc -
ase}}
```

In JavaScript

```
$filte r(' upp erc ase')()
```

filter11

Selects a subset of items from array and retur

HTML:

```
{{ filter _ex pre ssion | filter :
r}}
```

JS:

```
$filte r(' fil ter ')( array, expr
```

Arguments:

array, *Array*: The source array.

expression, *string* / *Object* / *function()*: The pre items from array.

Can be one of:

string: The string is evaluated as an expressio for substring match against the contents of the string properties in array that contain this strin can be negated by prefixing the string with !.

Object: A pattern object can be used to filter s contained by array. For example {name:"M", p an array of items which have property name c phone containing "1". A special property name to accept a match against any property of the simple substring match with a string as describ

function(value): A predicate function can be us function is called for each element of array. Th elements that the predicate returned true for.

comparator, *function(actual, expected)* / *true* / used in determining if the expected value (fron actual value (from the object in the array) shou

Can be one of:

filter11 (cont)

function(actual, expected): The function will be given the object value and the predicate value to compare and should return true if the item should be included in filtered result.
true: A shorthand for function(actual, expected) { return angular.equals(expected, actual)}. this is essentially strict comparison of expected and actual.
false|undefined: A short hand for a function which will look for a substring match in case insensitive way.

orderBy11

Orders a specified array by the expression predicate. It is ordered alphabetically for strings and numerically for numbers. Note: if you notice numbers are not being sorted correctly, make sure they are actually being saved as numbers and not strings.
In HTML Template Binding:

```
{{ orderB y_e xpr ession | orderBy : e}}
```

In JavaScript:

```
$filte r(' ord erB y') (array, expres sion, reverse)
```

Arguments:
array, *Array*: The array to sort.
expression *function(*) | string | Array.(function(*)|string)>*: A predicate to be used by the comparator to determine the order of elements.
Can be one of:
function: Getter function. The result of this function will be sorted using the <, =, > operator.
string: An Angular expression. The result of this expression is used to compare elements (for example name to sort by a property called name or name.substr(0, 3) to sort by 3 first characters of a property called name). The result of a constant expression is interpreted as a property name to be used in comparisons (for example "special name" to sort object by the value of their special name property).

orderBy11 (cont)

An expression can be optionally prefixed with + or - to control ascending or descending sort order (for example, +name or -name).. *Array*: An array of function or string predicates. The first predicate in the array is used for sorting, but when two items are equivalent, the next predicate is used.
reverse(optional), *boolean*: Reverse the order of the array.
Returns
Array: Sorted copy of the source array.

limitTo32

Creates a new array or string containing only a specified number of elements. The elements are taken from either the beginning or the end of the source array or string, as specified by the value and sign (positive or negative) of limit.
In HTML Template Binding:
{{ limitTo_expression | limitTo : limit}}
In JavaScript:
$filter('limitTo')(input, limit)
Arguments:
input *Array | string*: Source array or string to be limited.
limit *string | number*: The length of the returned array or string. If the limit number is positive, limit number of items from the beginning of the source array/string are copied. If the number is negative, limit number of items from the end of the source array/string are copied. The limit will be trimmed if it exceeds array.length
Returns
Array | string: A new sub-array or substring of length limit or less if input array had less than limit elements.