

Plugin Fancy Table

Use this [wiki plugin](#) to display data using the odd/even table style. Settings allow the table to be sortable by multiple rows, cause rows to span multiple columns and vice versa, set individual column widths and align text horizontally and vertically. Beginning with [Tiki12](#) add sorting, filtering and pagination using the [tableserter](#) parameters (separate tab below).

Parameters

Create a formatted table that can be filtered and sorted

Introduced in Tiki 1.

[Go to the source code](#)

Preferences required: `wikiplugin_fancytable`

Parameters	Accepted Values	Description	Default	Since
(body of plugin)		Rows separated by <code>>></code> in the header; for the table body, one row per line. Cells separated by <code> </code> (since Tiki4) or <code>~ ~</code> in both cases.		
<code>colwidths</code>		Column widths followed by "px" for pixels or "%" for percentages. Each column separated by <code> </code> .		4.1
<code>headclass</code>		CSS class to apply to the heading row.		1
<code>head</code>		Header rows of the table. Use <code>>></code> to separate multiple rows.		1
<code>colaligns</code>	text	Table body column horizontal alignments, separated by <code> </code> . Choices: <code>left</code> , <code>right</code> , <code>center</code> , <code>justify</code>		4.1
<code>colvaligns</code>	text	Table body column vertical alignments, separated by <code> </code> . Choices: <code>top</code> , <code>middle</code> , <code>bottom</code> , <code>baseline</code>		4.1
<code>headaligns</code>	text	Horizontal alignments for header cells, separated by <code> </code> . Choices: <code>left</code> , <code>right</code> , <code>center</code> , <code>justify</code>		4.1
<code>headvaligns</code>	text	Vertical alignments for header cells, separated by <code> </code> ". Choices: <code>top</code> , <code>middle</code> , <code>bottom</code> , <code>baseline</code>		4.1
<code>allowStickyHeaders</code>	(blank) n y	Sticky Headers for the table when scrolling top Default value: No	n	26

Create a formatted table that can be filtered and sorted

[Go to the source code](#)

Parameters	Accepted Values	Description
------------	-----------------	-------------

Prerequisites and Tips

- In order to use `tablesorter`,
 - Javascript must be enabled (enabled by default at **Control Panels > Features > Programmer (tab)**)
 - jQuery Sortable Tables must be activated (at **Control Panels > Features > Interface (tab)**)
- Sorting a table with cells that span multiple columns or rows can give unexpected results
- Any ` ` that you don't want interpreted as cell dividers may need to be placed inside of the ` ` and ` ` tags.

Examples

Sorting, Filtering and Paginating

There are a number of options that allow the user to filter, sort and paginate a table. Below is an example that demonstrates many of these options.

This code:

```
{FANCYTABLE(head=" Fruit | Number | Vegetables | Date | Amount" sortable="type:reset"
sortList="[0,0],[1,0]"
tsortcolumns="type:text;group:letter|type:digit;group:number|type:word;group:word|type:shortDate;gr
oup:date-year|type:digit;group:number-10" tsfilters="type:text;placeholder:Type to
filter...|type:range;from:0;to:200|type:dropdown|type:date|type:range;from:5;to:60;style:popup"
tsfilteroptions="type:reset" tspaginate="max:5")} apples|10 | onions | 2/1/2010|40 lemons|200 |
cucumbers | 3/3/2011|50 oranges|100 | carrots | 4/3/2012|60 lemons |10| tomatoes|1/2/2011|30 berries
|50 | peas|6/6/2011|55 apples|10 | onions | 2/1/2010|66 lemons|200 | cucumbers | 3/3/2011|66
oranges|100 | carrots | 4/3/2012|30 lemons |10| tomatoes|1/2/2011|22 berries |50 | peas|6/6/2011|77
apples|10 | onions | 2/1/2010|77 lemons|200 | cucumbers | 3/3/2011|34 oranges|100 | carrots |
4/3/2012|56 lemons |10| tomatoes|1/2/2011|67 berries |50 | peas|6/6/2011|78 apples|10 | onions |
2/1/2010|89 lemons|200 | cucumbers | 3/3/2011|12 oranges|100 | carrots | 4/3/2012|43 lemons |10|
tomatoes|1/2/2011|32 berries |50 | peas|6/6/2011|76 apples|10 | onions | 2/1/2010|76 lemons|200 |
cucumbers | 3/3/2011|87 oranges|100 | carrots | 4/3/2012|11 lemons |10| tomatoes|1/2/2011|22 berries
|50 | peas|6/6/2011|6 {FANCYTABLE}
```

Would produce:

Fruit	Number	Vegetables	Date	Amount
apples	10	onions	2/1/2010	40
lemons	200	cucumbers	3/3/2011	50
oranges	100	carrots	4/3/2012	60

Fruit	Number	Vegetables	Date	Amount
lemons	10	tomatoes	1/2/2011	30
berries	50	peas	6/6/2011	55
apples	10	onions	2/1/2010	66
lemons	200	cucumbers	3/3/2011	66
oranges	100	carrots	4/3/2012	30
lemons	10	tomatoes	1/2/2011	22
berries	50	peas	6/6/2011	77
apples	10	onions	2/1/2010	77
lemons	200	cucumbers	3/3/2011	34
oranges	100	carrots	4/3/2012	56
lemons	10	tomatoes	1/2/2011	67
berries	50	peas	6/6/2011	78
apples	10	onions	2/1/2010	89
lemons	200	cucumbers	3/3/2011	12
oranges	100	carrots	4/3/2012	43
lemons	10	tomatoes	1/2/2011	32
berries	50	peas	6/6/2011	76
apples	10	onions	2/1/2010	76
lemons	200	cucumbers	3/3/2011	87
oranges	100	carrots	4/3/2012	11
lemons	10	tomatoes	1/2/2011	22
berries	50	peas	6/6/2011	6

Automatic Totals

Automatic columns, row and table totals can be added. In addition to sums, the totals can be the count, max, min, mean, median, mode, range, varp, vars, stdevp, and stdevs (click [here](#) for a description of these options). The number format can also be set to apply to all totals or specific formats for each. The example below has all three types of totals (column, row and table) as well as variations showing totals based on whether values are hidden or not, including due to filtering.

This code:

```
{FANCYTABLE(head=" Fruit | Number | Vegetables | Amount" sortable="type:reset"
sortList="[0,0],[1,0],[2,y],[3,y],[4,n]" colaligns="left|right|left|right"
tsortcolumns="type:text|type:digit|type:word|type:digit" tsfilters="type:text;placeholder:Type to
filter...|type:range;from:0;to:200|type:dropdown|type:range;from:5;to:60;style:popup"
tsfilteroptions="type:reset" tspaginate="max:5"
tstotals="type:col|type:col;filter:hidden|type:col;filter:all|type:all|type:all;filter:hidden|type:all;filter:all|t
ype:row" tstotaloptions="ignore||ignore|" tstotalformat="#.###.")} apples|10 | onions | 40 lemons|200
| cucumbers | 50 ... (leaving out some rows for illustration) lemons |10| tomatoes|22 berries |50 | peas|6
{FANCYTABLE}
```

Would produce:

Fruit	Number	Vegetables	Amount
apples	10	onions	40
lemons	200	cucumbers	50
oranges	100	carrots	60
lemons	10	tomatoes	30
berries	50	peas	55
apples	10	onions	66
lemons	200	cucumbers	66
oranges	100	carrots	30
lemons	10	tomatoes	22
berries	50	peas	77
apples	10	onions	77
lemons	200	cucumbers	34
oranges	100	carrots	56
lemons	10	tomatoes	67
berries	50	peas	78
apples	10	onions	89
lemons	200	cucumbers	12
oranges	100	carrots	43
lemons	10	tomatoes	32
berries	50	peas	76

Fruit	Number	Vegetables	Amount
apples	10	onions	76
lemons	200	cucumbers	87
oranges	100	carrots	11
lemons	10	tomatoes	22
berries	50	peas	6

Multiple live filtering

In a simpler example, all you need to do is set `sortable` to `y`, in addition to being able to sort each column, you also get a line with fields to filter your data by matching a search string in one or more columns.

Example from <http://i18n.tiki.org/Status> :

This code:

```
{FANCYTABLE(head="Language code (ISO)|English name|Native
Name|Completion|Percentage|Number of strings", sortable="y")} ar | Arabic | العربية | {gauge
value="2,29" size="100" showvalue="false"} | 2,29% | Total: 14923 %%% Translated: 341 %%%
Untranslated: 14582 bg | Bulgarian | български език | {gauge value="0,01" size="100"
showvalue="false"} | 0,01% | Total: 14922 %%% Translated: 2 %%% Untranslated: 14920 ca | Catalan |
Català | {gauge value="39,22" size="100" showvalue="false"} | 39,22% | Total: 14996 %%%
Translated: 5882 %%% Untranslated: 9114 ... {FANCYTABLE}
```

Would produce (upper part):

Language code (ISO)	English name	Native Name	Completion	Percentage	Number of strings
ar	Arabic	العربية		2,29%	Total: 14923 Translated: 341 Untranslated: 14582
bg	Bulgarian	български език		0,01%	Total: 14922 Translated: 2 Untranslated: 14920
ca	Catalan	Català		39,22%	Total: 14996 Translated: 5882 Untranslated: 9114

Click to expand

Then you can sort by one or more columns, and filter your results by searching for some string in one or more columns. In the example below, sorted by one column ("Percentage"), and filtered by content in another column ("Native name" containing "de"):

Language code (ISO)	English name	Native Name	Completion	Percentage	Number of strings
de	German	Deutsch		93,59%	Total: 14983 Translated: 14023 Untranslated: 960
nl	Dutch	Nederlands		21,06%	Total: 14964 Translated: 3152 Untranslated: 11812
fy-NL	Frisian Netherlands	Frysk Nederlâns		0,66%	Total: 14922 Translated: 98 Untranslated: 14824

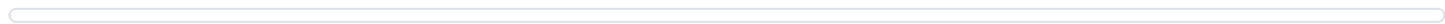
Click to expand

Cells Spanning More Than One Row Or More Than One Col

If the cell begins with (multiple) \, then the cell spans (multiple) rows. Define lesser cols in the following rows.

If the cell begins with (multiple) /, then the cell spans (multiple) cols. Define lesser cols in this row.

This code:



```
{FANCYTABLE( head=" Col 1 | Col 2 ")} \\1,1 (spans 2 rows)|1,2 2,2 //3,1 (spans 2 cols)
{FANCYTABLE}
```

Would produce:

Col 1	Col 2
1,1 (spans 2 rows)	1,2
	2,2
3,1 (spans 2 cols)	

Related pages

- [PluginSplit](#) - arranges elements into an unformatted table
- [jQuery](#)
- [Tablesorter](#) - jQuery plugin used for sorting, filtering and pagination features
- <http://mottie.github.io/tablesorter/docs/> - documentation for the jQuery plugin used for the sorting, filtering and pagination features