```
/***
    USAGE:

    PageDirectory(pages, mapping, columnCount, listPagesOptions, id,
description, missing, unclassified)
        build a multi-column page directory

    PARAMETERS:

    (optional) pages : list/map/str
        list/map of pages to list; if pages is a str, then it is used as
a path to a parent page to list all subpages;
        defaults to list of subpages of current page

    (optional) mapping : str
        if nil, map pages by first letter in title
        if empty, map pages by tags without a prefix
        if non-empty, map pages by tags with given tag prefix

    (optional) columns : num
        number of columns to display; defaults to 2

    (optional) listPagesOptions : map
        options passed into the ListPages template for listing each
column entry;
        defaults to { sort: 'viewcount', reverse: true, style: 'bullets' }

    (optional) id : str
        dom id used for table; default: none

    (optiona) description : str
        description of featured pages; default: nothing

    (optional) missing : xml
        placeholder text when no pages were found

    (optional) unclassified : str
        text to use for unclassified pages; default: '(Unclassified)'

***/

var pages = $0 ?? $pages ?? page.subpages;
if(pages is str) let pages = wiki.getpage(pages).subpages;
if(pages is map) let pages = map.values(pages);
var mapping = $1 ?? $mapping;
var columns = num.max($2 ?? $columns ?? 2, 1);
var listPagesOptions = { style: 'bullets' } .. ($3 ?? $listPagesOptions
?? { });
var description = $4 ?? $description;
```

```
var id = $5 ?? $id;
var missing = $6 ?? $missing;
var unclassified = $7 ?? $unclassified ?? '(Unclassified)';

// build map of all tags in pages
var pagemap = { };
foreach(var p in pages) {
    if(mapping is nil) {
        var c = p.title[0];
        var tags = [ { name: string.isletter(c) ? c : '#', type: 'text' }
];
    } else if(#mapping) {
        var prefix = mapping .. ':';

        // gather all tags that match the prefix
        var tags = [ { name: string.replace(string.substr(t.name,
#prefix), '_', ' '), type: 'text' }
            foreach var t in p.tags
            where t.type === 'text' && string.startswith(t.name, prefix,
true)
        ];
    } else {

        // gather all tags without prefix
        var tags = [ { name: string.replace(t.name, '_', ' '), type:
'text' }
            foreach var t in p.tags
            where t.type === 'text' && !string.contains(t.name, ':')
        ];
    }

    // check if page has no matching tags; if so make up a default list
    if(!#tags) {
        let tags = [ { name: unclassified, type: 'text' } ];
    }

    // for each tag on the page, append the page to that tag's list
    foreach(var t in tags, var name = t.name) {
        let pagemap ..= { (name) : pagemap[name] .. [ p ] };
    }
}

if(#pagemap) {

    // count how many pages each tag has
    var tag_count = [ { tag: tag, count: #pagemap[tag] } foreach var tag
in map.keys(pagemap) ];

    // balance columns so that their heights are as equal as possible
    var column_tags = list.new(columns, [ ]);
```

```
    var column_sums = list.new(columns, 0);

    foreach(var t in list.sort(tag_count, 'count', true)) {

        // find shortest column
        var column = list.indexof(column_sums, list.min(column_sums));

        // update column
        let column_tags = list.splice(column_tags, column) .. [
column_tags[column] .. [ t.tag ] ] .. list.splice(column_tags, 0, column
+ 1);
        let column_sums = list.splice(column_sums, column) .. [
column_sums[column] + (listPagesOptions.limit ?? t.count) + 2 ] ..
list.splice(column_sums, 0, column + 1);
    }

    // emit the table with N columns
    <table border="0" id=(id) width="100%">
        if(#string.trim(description ?? '')) {
            <tr>
                <td align="left" colspan="2" valign="top">
                    <p> description </p>
                </td>
            </tr>
        }
        <tr>

            // loop over each column
            foreach(var column in column_tags) {
                <td align="left" valign="top" width=(num.format(1 /
columns, '##%'))>
                    <div class="left">

                        // loop over all tags in column, sorted
alphabetically
                        foreach(var tag in list.sort(column)) {
                            <h3> string.tocamelcase(tag) </h3>
                            template("MindTouch/Controls/ListPages",
listPagesOptions .. { pages: pagemap[tag] });
                        }
                    </div>
                </td>
            }
        </tr>
    </table>
} else {

    // emit the table with N columns
    <table border="0" id=(id) width="100%">
        if(#string.trim(description ?? '')) {
```

```
            <tr>
                <td align="left" colspan="2" valign="top">
                    <p> description </p>
                </td>
            </tr>
        }
        <tr>
            <td>
                missing;
            </td>
        </tr>
    </table>
}
```