```
/***
   USAGE:

   ListPages(pages, sort, reverse, limit, style, stripTitlePrefix)
      create a bulleted list of sorted pages with an optional limit

   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) sort : str
      sort order for pages; one of 'custom', 'updated', 'created', 'viewed', 'rated', or 'title'; defaults to 'title'

   (optional) reverse : bool
      reverse sort order; defaults to false

   (optional) limit : num
      maximum number of pages to show; defaults to no limit

   (optional) style : str
      list style to use; one of 'numbers' or 'bullets'; defaults to 'numbers'

   (optional) stripTitlePrefix : str
      strip title prefix when present; defaults to none

***/

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 sort = $1 ?? $sort ?? 'title';
var reverse = $2 ?? $reverse ?? false;
var limit = $3 ?? $limit;
var liststyle = $4 ?? $style ?? 'numbers';
var striptitleprefix = $5 ?? $striptitleprefix;
var dateformat = $dateformat ??
xml.text(wiki.localize('MindTouch.Templates.Controls.ListPages.dateformat'));
var numformat = $numformat ??
xml.text(wiki.localize('MindTouch.Templates.Controls.ListPages.numformat'));

// sort pages list depending on sort order
switch(sort) {
case 'updated':
   let pages = [ p .. { sortkey: date.format(d, 'yyyyMMddHHmmss'), sortlabel: date.format(d, dateformat) }
foreach var p in pages, var d = p.date ];
case 'created':
   let pages = [ p .. { sortkey: date.format(d, 'yyyyMMddHHmmss'), sortlabel: date.format(d, dateformat) }
```

```
foreach var p in pages, var d = p.revisions[0].date ];
case 'viewed':
case 'viewcount':
   let pages = [ p .. { sortkey: num.format(p.viewcount, '0000000000'), sortlabel: num.format(p.viewcount,
numformat) } foreach var p in pages ];
case 'rated':
   let pages = [ p .. { sortkey: num.format(r.score ?? 0, '0.000000000'), sortlabel: r.score ?
num.format(r.score, '###%') : nil } foreach var p in pages, var r = p.rating ];
   break;
case 'custom':
   break;
case 'title':
default:
   let pages = [ p .. { sortkey: p.title, sortlabel: nil } foreach var p in pages ];
}

// determine list style
var listelem;
switch(string.tolower(liststyle)) {
case 'bullets':
   let listelem = 'ul';
case 'numbers':
default:
   let listelem = 'ol';
}

// check if there is at least one page to show
if(#pages) {
   let pages = list.sort(pages, 'sortkey', reverse);
   <font size="-1">
      <(listelem)>
         foreach(var p in pages where limit ? (__count < limit) : true) {
            <li>
               var title = p.title;
               if(striptitleprefix && string.startswith(p.title, striptitleprefix, true)) {
                  let title = string.trim(string.substr(p.title, #striptitleprefix));
               }
               web.link(p.uri, title);

               // check if there is a label to show
               if(p.sortlabel) {
                  <span style="color: rgb(128, 128, 128); font-size: smaller;">
                     ' ('; p.sortlabel; ')';
                  </span>
               }
            </li>
         }
      </>
   </font>
}
```