https://fa.wikimarefat.ir/view/%D9%BE%D9%88%D8%AF%D9%85%D8%A7%D9%86:Navbox_with_collapsible_groups

پودمان:Navbox with collapsible groups

□□□□ □□□□□□□□/Navbox with collapsible groups:□□□□□□ □□ □□□□□□□□ □□□□□□□ □□□□ □□□□□□□□ .□□□□

```
-- This module implements {{Navbox with collapsible groups}}
local q = {}
local Navbox = require('پودمان:Navbox')

-- helper functions
local function concatstrings(s)
        local r = table.concat(s, '')
        if r:match('^%s*$') then r = nil end
        return r
end

local function concatstyles(s)
        local r = table.concat(s, ';')
        while r:match(';%s*;') do
                r = mw.ustring.gsub(r, ';%s*;', ';')
        end
        if r:match('^%s*;%s*$') then r = nil end
        return r
end

function q._navbox(pargs)
        -- table for args passed to navbox
        local targs = {}

        -- process args
        local passthrough = {
                ['name']=true,['navbar']=true,['state']=true,['border']=true,
                ['bodyclass']=true,['groupclass']=true,['listclass']=true,
                ['style']=true,['bodystyle']=true,['basestyle']=true,
                ['title']=true,['titleclass']=true,['titlestyle']=true,
                ['above']=true,['aboveclass']=true,['abovestyle']=true,
                ['below']=true,['belowclass']=true,['belowstyle']=true,
                ['image']=true,['imageclass']=true,['imagestyle']=true,
                ['imageleft']=true,['imageleftstyle']=true,
                ['imageright']=true,['imagerightstyle']=true
        }
        for k,v in pairs(pargs) do
                if k and type(k) == 'string' then
                        if passthrough[k] then
                                targs[k] = v
                        elseif (k:match('^list[0-9][0-9]*$')
```

```
                                        or k:match('^content[0-9][0-9]*$') )
then
                                local n = mw.ustring.gsub(k, '^[a-
z]*([0-9]*)$', '%1')
                                if (targs['list' .. n] == nil and
pargs['group' .. n] == nil
                                        and pargs['sect' .. n] == nil and
pargs['section' .. n] == nil) then
                                        targs['list' .. n] = concatstrings(
                                                {pargs['list' .. n] or '',
pargs['content' .. n] or ''})
                                end
                        elseif (k:match('^group[0-9][0-9]*$')
                                        or k:match('^sect[0-9][0-9]*$')
                                        or k:match('^section[0-9][0-9]*$') )
then
                                local n = mw.ustring.gsub(k, '^[a-
z]*([0-9]*)$', '%1')
                                if targs['list' .. n] == nil then
                                        local titlestyle = concatstyles(
                                                {pargs['groupstyle'] or
'',pargs['secttitlestyle'] or '',
                                                        pargs['group' .. n ..
'style'] or '',
                                                        pargs['section' .. n
..'titlestyle'] or ''})
                                        local liststyle = concatstyles(
                                                {pargs['liststyle'] or '',
pargs['contentstyle'] or '',
                                                        pargs['list' .. n ..
'style'] or '',
                                                        pargs['content' .. n
.. 'style'] or ''})
                                        local title = concatstrings(
                                                {pargs['group' .. n] or '',
                                                        pargs['sect' .. n] or
'',
                                                        pargs['section' .. n]
or ''})
                                        local list = concatstrings(
                                                {pargs['list' .. n] or '',
                                                        pargs['content' .. n]
or ''})
                                        local state = (pargs['abbr' .. n] and
pargs['abbr' .. n] == pargs['selected'])
                                                and 'uncollapsed' or
pargs['state' .. n] or 'collapsed'
                                        targs['list' .. n] = Navbox._navbox(
                                                {'child', navbar = 'plain',
state = state,
                                                basestyle =
```

```
pargs['basestyle'],
                                                        title = title, titlestyle =
titlestyle,
                                                        list1 = list, liststyle =
liststyle,
                                                        listclass = pargs['list' .. n
.. 'class'],
                                                        image = pargs['image' .. n],
                                                        imageleft = pargs['imageleft'
.. n],
                                                        imageright =
pargs['imageright' .. n],
                                                        listpadding =
pargs['listpadding']})
                                    end
                            end
                    end
            end
        -- ordering of style and bodystyle
        targs['style'] = concatstyles({targs['style'] or '',
targs['bodystyle'] or ''})
        targs['bodystyle'] = nil
        -- child or subgroup
        if targs['border'] == nil then targs['border'] = pargs[1] end

        return Navbox._navbox(targs)
end

function q.navbox(frame)
        local pargs = require('پودمان:Arguments').getArgs(frame, {wrappers =
{'الگو:Navbox with collapsible groups'}})

        -- Read the arguments in the order they'll be output in, to make
references number in the right order.
        local _
        _ = pargs.title
        _ = pargs.above
        for i = 1, 20 do
                _ = pargs["group" .. tostring(i)]
                _ = pargs["list" .. tostring(i)]
        end
        _ = pargs.below

        return q._navbox(pargs)
end

return q
```