

Modul:If preview

Vorlage:Lua Vorlage:Uses TemplateStyles

This module implements [Vorlage:TI](#) and [Vorlage:TL](#). It helps templates/modules determine if they are being previewed.

Prefer implementing the template versions in other templates.

In a module to use the `main()`, you need to pass a frame table with an args table.

For the preview warning, use `_warning()`.

```
local p = {}

local cfg = mw.loadData('Module:If preview/configuration')

--[[

main

This function returns either the first argument or second argument passed to
this module, depending on whether the page is being previewed.

]]
function p.main(frame)
    if cfg.preview then
        return frame.args[1] or ''
    else
        return frame.args[2] or ''
    end
end

--[[

pmain

This function returns either the first argument or second argument passed to
this module's parent (i.e. template using this module), depending on whether it
is being previewed.

]]
function p.pmain(frame)
    return p.main(frame:getParent())
end

local function warning_text(warning)
    return mw.ustring.format(
        cfg.warning_infrastructure,
        cfg.templatestyles,
        warning
    )
end

function p._warning(args)

    local warning = args[1] and args[1]:match('^%s*(.-)%s*$') or ''
```

Modul:If preview

```
    if warning == '' then
        return warning_text(cfg.missing_warning)
    end

    if not cfg.preview then return '' end

    return warning_text(warning)
end

--[[  
warning

This function returns a "preview warning", which is the first argument marked
up with HTML and some supporting text, depending on whether the page is being pre-
disabled since we'll implement the template version in general

]]
--function p.warning(frame)
--    return p._warning(frame.args)
--end

--[[  
warning, but for pass-through templates like {{preview warning}}
]]
function p.pwarning(frame)
    return p._warning(frame.getParent().args)
end

return p
```