

Modul:For

Implements [Vorlage:TI](#).

```
local mArguments --initialize lazily
local mHatlist = require('Module:Hatnote list')
local mHatnote = require('Module:Hatnote')
local yesNo = require('Module:Yesno')
local p = {}

--Implements {{For}} from the frame
--uses capitalized "For" to avoid collision with Lua reserved word "for"
function p.For (frame)
    mArguments = require('Module:Arguments')
    return p._For(mArguments.getArgs(frame))
end

--Implements {{For}} but takes a manual arguments table
function p._For (args)
    local use = args[1]
    if (not use) then
        return mHatnote.makeWikitextError(
            'no context parameter provided. Use {{other uses}} for "c
            'Template:For#Errors',
            args.category
        )
    end
    local pages = {}
    function two (a, b) return a, b, 1 end --lets us run ipairs from 2
    for k, v in two(ipairs(args)) do table.insert(pages, v) end
    local title = mw.title.getCurrentTitle()
    local skipCat = title.isTalkPage or title.namespace == 2 --don't categori
    local oddCat = skipCat and '' or '[[Category:Hatnote templates using unus
    local category = yesNo(args.category)
    return mHatnote._hatnote(
        mHatlist.forSeeTableToString({{use = use, pages = pages}}),
        {selfref = args.selfref}
    ) .. (
        (use == 'other uses') and ((category == true) or (category == oddCat or ''
    )
end

return p
```