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

پودمان:Transclusion count

.□□□□ □□□□ [□□□□□□□□/Transclusion count:□□□□□□](#) □□ □□□□□□□□ □□□□□□□ □□□□ □□□□□□□□

```
local p = {}

function p._fetch(args)
        local template = nil
        local return_value = nil

        -- Use demo parameter if it exists, otherwise use current template
name
        local namespace = mw.title.getCurrentTitle().namespace
        if args["demo"] and args["demo"] ~= "" then
                template = mw.ustring.gsub(args["demo"],"^[Tt]emplate:","")
        elseif namespace == 10 then -- Template namespace
                template = mw.title.getCurrentTitle().text
        elseif namespace == 828 then -- Module namespace
                template = (mw.site.namespaces[828].name .. ":" ..
mw.title.getCurrentTitle().text)
        end

        -- If in template or module namespace, look up count in /data
        if template ~= nil then
                namespace = mw.title.new(template, "Template").namespace
                if namespace == 10 or namespace == 828 then
                        template =  mw.ustring.gsub(template, "/doc$", "") --
strip /doc from end
                        template =  mw.ustring.gsub(template, "/sandbox$",
"") -- strip /sandbox from end
                        local index =
mw.ustring.sub(mw.title.new(template).text,1,1)
                        local status, data = pcall(function ()
return(mw.loadData('Module:Transclusion_count/data/' ..
(mw.ustring.find(index, "%a") and index or "other")))
                        end)
                        if status then
                                return_value =
tonumber(data[mw.ustring.gsub(template, " ", "_")])
                        end
                end
        end
        -- If database value doesn't exist, use value passed to template
        if return_value == nil and args[1] ~= nil then
                local arg1=mw.ustring.match(args[1], '[%d,]+')
                if arg1 and arg1 ~= '' then
```

```
                        return_value =
tonumber(mw.getCurrentFrame():callParserFunction('formatnum', arg1, 'R'))
                end
        end
        return return_value
end

function p.fetch(frame)
        return p._fetch(frame.args)
end

-- Tabulate this data for [[Wikipedia:Database reports/Templates transcluded
on the most pages]]
function p.tabulate()
        local list = {}
        for i = 65, 91 do
                local data = mw.loadData('Module:Transclusion count/data/' ..
((i == 91) and 'other' or string.char(i)))
                for name, count in pairs(data) do
                        local title = mw.title.new(name, "Template")
                        table.insert(list, {title, count})
                end
        end
        table.sort(list, function(a, b)
                return (a[2] == b[2]) and (a[1] < b[1]) or (a[2] > b[2])
        end)
        local lang = mw.getContentLanguage();
        for i = 1, #list do
                local protLevel = ""
                local title = list[i][1]
                if i < 400 then
                        -- Stop at 400 to avoid breaching the expensive
parser function call limit
                        protLevel = title.protectionLevels.edit[1] or
"(unprotected)"
                end
                list[i] = ('|-\n| %d || [[%s]] || %s || %s \n'):format(i,
title.fullText:gsub('_', ' '), lang:formatNum(list[i][2]), protLevel)
        end
        return table.concat(list)
end

return p
```