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

پودمان:Check for unknown parameters

□□□□ [□□□□□□□□/Check for unknown parameters:□□□□□□](#) □□ □□□□□□□□ □□□□□□□ □□□□ □□□□□□□□ .□□□□

```
-- This module may be used to compare the arguments passed to the parent
-- with a list of arguments, returning a specified result if an argument is
-- not on the list

require ('strict');

local p = {}

local mapframe_params = {
        'coord',
        'coordinates',
        'id',
        'qid',
        'mapframe',
        'mapframe-area_km2',
        'mapframe-area_mi2',
        'mapframe-caption',
        'mapframe-coord',
        'mapframe-coordinates',
        'mapframe-custom',
        'mapframe-frame-coord',
        'mapframe-frame-coordinates',
        'mapframe-frame-height',
        'mapframe-frame-width',
        'mapframe-geomask',
        'mapframe-geomask-fill',
        'mapframe-geomask-fill-opacity',
        'mapframe-geomask-stroke-color',
        'mapframe-geomask-stroke-colour',
        'mapframe-geomask-stroke-width',
        'mapframe-height',
        'mapframe-id',
        'mapframe-length_km',
        'mapframe-length_mi',
        'mapframe-line',
        'mapframe-line-stroke-color',
        'mapframe-line-stroke-colour',
        'mapframe-marker',
        'mapframe-marker-color',
        'mapframe-marker-colour',
        'mapframe-point',
```

```
        'mapframe-population',
        'mapframe-shape',
        'mapframe-shape-fill',
        'mapframe-shape-fill-opacity',
        'mapframe-shape-stroke-color',
        'mapframe-shape-stroke-colour',
        'mapframe-stroke-color',
        'mapframe-stroke-colour',
        'mapframe-stroke-width',
        'mapframe-switcher',
        'mapframe-type',
        'mapframe-width',
        'mapframe-wikidata',
        'mapframe-zoom'
}

local function trim(s)
        return s:match('^%s*(.-)%s*$')
end

local function isnotempty(s)
        return s and s:match('%S')
end

local function clean(text)
        -- Return text cleaned for display and truncated if too long.
        -- Strip markers are replaced with dummy text representing the
original wikitext.
        local pos, truncated
        local function truncate(text)
                if truncated then
                        return ''
                end
                if mw.ustring.len(text) > 25 then
                        truncated = true
                        text = mw.ustring.sub(text, 1, 25) .. '...'
                end
                return mw.text.nowiki(text)
        end
        local parts = {}
        for before, tag, remainder in text:gmatch('([^\127]*)\127[^\127]*%-
(%l+)%-[^\127]*\127()') do
                pos = remainder
                table.insert(parts, truncate(before) .. '&lt;' .. tag ..
'&gt;...&lt;/' .. tag .. '&gt;')
        end
        table.insert(parts, truncate(text:sub(pos or 1)))
        return table.concat(parts)
end

function p._check(args, pargs)
```

```
        if type(args) ~= "table" or type(pargs) ~= "table" then
                -- TODO: error handling
                return
        end

        -- create the list of known args, regular expressions, and the return
string
        local knownargs = {}
        if isnotempty(args['mapframe_args']) then
                for _, v in ipairs(mapframe_params) do
                knownargs[v] = 1
                end
        end
        local regexps = {}
        for k, v in pairs(args) do
                if type(k) == 'number' then
                        v = trim(v)
                        knownargs[v] = 1
                elseif k:find('^regexp[1-9][0-9]*$') then
                        table.insert(regexps, '^' .. v .. '$')
                end
        end
        -- loop over the parent args, and make sure they are on the list
        local ignoreblank = isnotempty(args['ignoreblank'])
        local showblankpos = isnotempty(args['showblankpositional'])
        local values = {}
        for k, v in pairs(pargs) do
                if type(k) == 'string' and knownargs[k] == nil then
                        local knownflag = false
                        for _, regexp in ipairs(regexps) do
                                if mw.ustring.match(k, regexp) then
                                        knownflag = true
                                        break
                                end
                        end
                        if not knownflag and ( not ignoreblank or
isnotempty(v) )  then
                                table.insert(values, clean(k))
                        end
                elseif type(k) == 'number' and knownargs[tostring(k)] == nil
then
                        local knownflag = false
                        for _, regexp in ipairs(regexps) do
                                if mw.ustring.match(tostring(k), regexp) then
                                        knownflag = true
                                        break
                                end
                        end
                        if not knownflag and ( showblankpos or isnotempty(v)
) then
                                table.insert(values, k .. ' = ' .. clean(v))
```

```
                        end
                end
        end

        -- add results to the output tables
        local res = {}
        if #values > 0 then
                local unknown_text = args['unknown'] or 'Found _VALUE_, '

                if mw.getCurrentFrame():preprocess( "{{REVISIONID}}" ) == ""
then
                        local preview_text = args['preview']
                        if isnotempty(preview_text) then
                                preview_text = require('Module:If
preview')._warning({preview_text})
                        elseif preview_text == nil then
                                preview_text = unknown_text
                        end
                        unknown_text = preview_text
                end
                for _, v in pairs(values) do
                        -- Fix odd bug for | = which gets stripped to the
empty string and
                        -- breaks category links
                        if v == '' then v = ' ' end

                        -- avoid error with v = 'example%2' ("invalid capture
index")
                        local r = unknown_text:gsub('_VALUE_', {_VALUE_ = v})
                        table.insert(res, r)
                end
        end

        return table.concat(res)
end

function p.check(frame)
        local args = frame.args
        local pargs = frame:getParent().args
        return p._check(args, pargs)
end

return p
```