

Toggle menu
Blue Gold Program Wiki

Navigation

- [Main page](#)
- [Recent changes](#)
- [Random page](#)
- [Help about MediaWiki](#)

Tools

- [What links here](#)
- [Related changes](#)
- [Special pages](#)
- [Permanent link](#)
- [Page information](#)

Personal tools

- [Log in](#)

personal-extra

Toggle search

Search

Random page

Views

- [View](#)
- [View source](#)
- [History](#)
- [PDF Export](#)

Actions

Module:Tlg

From Blue Gold Program Wiki

The printable version is no longer supported and may have rendering errors. Please update your browser bookmarks and please use the default browser print function instead.

Documentation for this module may be created at [Module:Tlg/doc](#)

```

-- This implements Template:Tlg
local getArgs = require('Module:Arguments').getArgs
local p = {}

-- Is a string non-empty?
local function _ne(s)
    return s ~= nil and s ~= ""
end

local nw = mw.text.nowiki

local function addTemplate(s)
    local i, _ = s:find(':', 1, true)
    if i == nil then
        return 'Template:' .. s
    end
    local ns = s:sub(1, i - 1)
    if ns == '' or mw.site.namespaces[ns] then
        return s
    else
        return 'Template:' .. s
    end
end

local function trimTemplate(s)
    local needle = 'template:'
    if s:sub(1, needle:len()):lower() == needle then
        return s:sub(needle:len() + 1)
    else
        return s
    end
end

local function linkTitle(args)
    if _ne(args.nolink) then
        return args['1']
    end
    local titleObj
    local titlePart = '['[
    if args['1'] then
        -- This handles :Page and other NS
        titleObj = mw.title.new(args['1'], 'Template')
    else
        titleObj = mw.title.getCurrentTitle()
    end
    titlePart = titlePart .. (titleObj ~= nil and titleObj.fullText or
        addTemplate(args['1']))
    local textPart = args.alttext
    if not _ne(textPart) then
        if titleObj ~= nil then
            textPart = titleObj.inNamespace("Template") and

```

```

args['1'] or titleObj.fullText
    else
        -- redlink
        textPart = args['1']
    end
end
if _ne(args.subst) then
    -- HACK: the ns thing above is probably broken
    textPart = 'subst:' .. textPart
end
if _ne(args.brace) then
    textPart = nw('{{') .. textPart .. nw('}}')
elseif _ne(args.braceinside) then
    textPart = nw('{') .. textPart .. nw('}')
end
titlePart = titlePart .. '|' .. textPart .. ']]'
if _ne(args.braceinside) then
    titlePart = nw('{') .. titlePart .. nw('}')
end
return titlePart
end

end

function p.main(frame)
    local args = getArgs(frame, {
        trim = true,
        removeBlanks = false
    })
    return p._main(args)
end

function p._main(args)
    local bold = _ne(args.bold) or _ne(args.boldlink) or
    _ne(args.boldname)
    local italic = _ne(args.italic) or _ne(args.italics)
    local dontBrace = _ne(args.brace) or _ne(args.braceinside)
    local code = _ne(args.code) or _ne(args.tt)
    -- Build the link part
    local titlePart = linkTitle(args)
    if bold then titlePart = "'''" .. titlePart .. "'''" end
    if _ne(args.nowrapname) then titlePart = '<span class="nowrap">' ..
titlePart .. '</span>' end
    -- Build the arguments
    local textPart = ""
    for i = 2, #args do
        local ii = tostring(i)
        local val = args[ii]
        if val ~= nil then val = mw.text.unstripNoWiki(val) else
break end
        if _ne(args.nowiki) then val = nw(val) end
        if italic then val = '<span style="font-style:italic;">' ..
val .. '</span>' end

```

```

        textPart = textPart .. '&#124;' .. val
    end
    if _ne(args['12']) then textPart = textPart .. '&#124;...' end
    -- final wrap
    local ret = titlePart .. textPart
    if not dontBrace then ret = nw('{{' .. ret .. '}}') end
    if _ne(args.a) then ret = nw('*') .. '&nbsp;' .. ret end
    if _ne(args.kbd) then ret = '<kbd>' .. ret .. '</kbd>' end
    if code then
        ret = '<code>' .. ret .. '</code>'
    elseif _ne(args.plaincode) then
        ret = '<code style="border:none;background:transparent;">' ..
ret .. '</code>'
    end
    if _ne(args.nowrap) then ret = '<span class="nowrap">' .. ret ..
'</span>' end
    --[[ Wrap as html??
    local span = mw.html.create('span')
    span:wikitext(ret)
    --]]
    if _ne(args.debug) then ret = ret .. '\n<pre>' ..
mw.text.encode(mw.dumpObject(args)) .. '</pre>' end
    return ret
end

```

return p

Retrieved from "<https://www.bluegoldwiki.com/index.php?title=Module:Tlg&oldid=2421>"

Namespaces

- [Module](#)
- [Discussion](#)

Variants

This page was last edited on 26 April 2020, at 05:11.

Blue Gold Program Wiki

The wiki version of the Lessons Learnt Report of the Blue Gold program, documents the experiences of a technical assistance (TA) team working in a development project implemented by the Bangladesh Water Development Board (BWDB) and the Department of Agricultural Extension (DAE) over an eight+ year period from March 2013 to December 2021. The wiki lessons learnt report (LLR) is intended to complement the BWDB and DAE project completion reports (PCRs), with the aim of recording lessons learnt for use in the design and implementation of future interventions in the coastal zone.

- [Privacy policy](#)
- [About Blue Gold Program Wiki](#)

- [Disclaimers](#)

Developed and maintained by Big Blue Communications for Blue Gold Program



[Blue Gold Program Wiki](#)