Modul:Yesno

This module provides a consistent interface for processing boolean or boolean-style string input. While Lua allows the `true` and `false` boolean values, wikicode templates can only express boolean values through strings such as "yes", "no", etc. This module processes these kinds of strings and turns them into boolean input for Lua to process. It also returns `nil` values as `nil`, to allow for distinctions between `nil` and `false`. The module also accepts other Lua structures as input, i.e. booleans, numbers, tables, and functions. If it is passed input that it does not recognise as boolean or `nil`, it is possible to specify a default value to return.

Inhaltsverzeichnis

1 Syntax 1
2 Usage 1
2.1 Undefined input ('foo') 2
2.2 Handling nil results 3

Syntax

```
yesno(value, default)
```

`value` is the value to be tested. Boolean input or boolean-style input (see below) always evaluates to either `true` or `false`, and `nil` always evaluates to `nil`. Other values evaluate to `default`.

Usage

First, load the module. Note that it can only be loaded from other Lua modules, not from normal wiki pages. For normal wiki pages you can use Vorlage:Tl instead.

```
local yesno = require('Module:Yesno')
```

Some input values always return `true`, and some always return `false`. `nil` values always return `nil`.

```
-- These always return true:
yesno('yes')
yesno('y')
yesno('true')
yesno('t')
yesno('1')
yesno(1)
yesno(true)

-- These always return false:
yesno('no')
```

```
yesno('n')
yesno('false')
yesno('f')
yesno('0')
yesno(0)
yesno(false)

-- A nil value always returns nil:
yesno(nil)
```

String values are converted to lower case before they are matched:

```
-- These always return true:
yesno('Yes')
yesno('YES')
yesno('yEs')
yesno('Y')
yesno('tRuE')

-- These always return false:
yesno('No')
yesno('NO')
yesno('nO')
yesno('N')
yesno('fALsE')
```

Undefined input ('foo')

You can specify a default value if yesno receives input other than that listed above. If you don't supply a default, the module will return `nil` for these inputs.

```
-- These return nil:
yesno('foo')
yesno({})
yesno(5)
yesno(function() return 'This is a function.' end)
yesno(nil, true)
yesno(nil, 'bar')

-- These return true:
yesno('foo', true)
yesno({}, true)
yesno(5, true)
yesno(function() return 'This is a function.' end, true)

-- These return "bar":
yesno('foo', 'bar')
yesno({}, 'bar')
yesno(5, 'bar')
yesno(function() return 'This is a function.' end, 'bar')
```

Note that the empty string also functions this way:

```
yesno('')        -- Returns nil.
yesno('', true)  -- Returns true.
yesno('', 'bar') -- Returns "bar".
```

Although the empty string usually evaluates to false in wikitext, it evaluates to true in Lua. This module prefers the Lua behaviour over the wikitext behaviour. If treating the empty string as false is important for your module, you will need to convert empty strings to a value that evaluates to false before passing them to this module. In the case of arguments received from wikitext, this can be done by using Module:Arguments.

Handling nil results

By definition

```
yesno(nil)         -- Returns nil.
yesno('foo')       -- Returns nil.
yesno(nil, true)   -- Returns nil.
yesno(nil, false)  -- Returns nil.
yesno('foo', true) -- Returns true.
```

To get the binary `true/false`-only values, use code like:

```
myvariable = yesno(value) or false -- When value is nil, result is false.
myvariable = yesno(value) or true  -- When value is nil, result is true.
myvariable = yesno('foo') or false  -- Unknown string returns nil, result is fals
myvariable = yesno('foo', true) or false  -- Default value (here: true) applies,
```

```
-- Function allowing for consistent treatment of boolean-like wikitext input.
-- It works similarly to the template {{yesno}}.

return function (val, default)
        -- If your wiki uses non-ascii characters for any of "yes", "no", etc., y
        -- should replace "val:lower()" with "mw.ustring.lower(val)" in the
        -- following line.
        val = type(val) == 'string' and val:lower() or val
        if val == nil then
                return nil
        elseif val == true
                or val == 'yes'
                or val == 'y'
                or val == 'true'
                or val == 't'
                or val == 'on'
                or tonumber(val) == 1
        then
                return true
        elseif val == false
                or val == 'no'
                or val == 'n'
                or val == 'false'
                or val == 'f'
                or val == 'off'
                or tonumber(val) == 0
```

```
        then
                return false
        else
                return default
        end
end
```