Inhaltsverzeichnis

1. Modul:OutputBuffer/Doku 2
2. Modul:OutputBuffer 4

Modul:OutputBuffer/Doku

Dies ist die Dokumentationsseite für Modul:OutputBuffer

This module provides an easy and efficient way to generate lengthy strings.

Inhaltsverzeichnis

1 Usage 2
1.1 getBuffer 2
1.2 print 2
1.3 printf 2
2 Example 3

Usage

First, load the module.

```
local newBuffer = require('Module:OutputBuffer')
```

Then, create the buffer and the functions that act on it.

```
local getBuffer, print, printf = newBuffer()
```

getBuffer

```
local text = getBuffer(sep)
```

Returns the contents of the buffer, with an optional separator string *sep*.

print

```
print(s)
```

Adds the string *s* to the buffer.

printf

```
printf(s, ...)
```

Adds the string *s* to the buffer. The string is formatted with any subsequent arguments, following the rules used for string.format.

Example

```
local newBuffer = require('Module:OutputBuffer')
local p = {}

function p.main()
local getBuffer, print, printf = newBuffer()
print('Welcome to the English Wikipedia.')
printf('There are currently %d articles on this site.', mw.site.stats.articles)
return getBuffer(' ')
end

return p

-- Assuming this module's name is "TestModule", and that mw.site.stats.articles r
-- {{#invoke:TestModule|main}} would output:
-- "Welcome to the English Wikipedia. There are currently 4500000 articles on thi
```

Modul:OutputBuffer

This module provides an easy and efficient way to generate lengthy strings.

Inhaltsverzeichnis

1 Usage 4
1.1 getBuffer 4
1.2 print 4
1.3 printf 4
2 Example 5

Usage

First, load the module.

```
local newBuffer = require('Module:OutputBuffer')
```

Then, create the buffer and the functions that act on it.

```
local getBuffer, print, printf = newBuffer()
```

getBuffer

```
local text = getBuffer(sep)
```

Returns the contents of the buffer, with an optional separator string *sep*.

print

```
print(s)
```

Adds the string *s* to the buffer.

printf

```
printf(s, ...)
```

Adds the string *s* to the buffer. The string is formatted with any subsequent arguments, following the rules used for string.format.

Example

```
local newBuffer = require('Module:OutputBuffer')
local p = {}

function p.main()
local getBuffer, print, printf = newBuffer()
print('Welcome to the English Wikipedia.')
printf('There are currently %d articles on this site.', mw.site.stats.articles)
return getBuffer(' ')
end

return p

-- Assuming this module's name is "TestModule", and that mw.site.stats.articles r
-- {{#invoke:TestModule|main}} would output:
-- "Welcome to the English Wikipedia. There are currently 4500000 articles on thi
```

```
return function()
        local buffer = {}
        return function(sep)
                local b = buffer
                buffer = {}
                return table.concat(b, sep)
        end,
        function(text)
                buffer[#buffer + 1] = text
        end,
        function(...)
                buffer[#buffer + 1] = string.format(...)
        end
end
```