



# Inhaltsverzeichnis

---

|  |    |
|--|----|
| 1. Modul:ComplForColorModules/Doku ..... | 2  |
| 2. Modul:BrewerColors .....              | 3  |
| 3. Modul:ChartColors .....               | 11 |
| 4. Modul:ComplForColorModules .....      | 15 |



## Modul:ComplForColorModules/Doku

---

**Dies ist die Dokumentationsseite für Modul:ComplForColorModules**

This module is used by **Module:BrewerColors** and **Module:ChartColors**.

To translate or review the translation of the module to your language, edit carefully [Data:l18n/ComplForColorModules.tab](#).

## Modul:BrewerColors

Used by **Vorlage:TI** and **Vorlage:TI**

To translate or review the translation of the module to your language, edit [Data:I18n/BrewerColors.tab](#) carefully.

### Function ShowColors

| Code  | Result  |
|---|---|
| <code>{{#invoke: BrewerColors ShowColors Oranges 5}}</code>   | <b>Lua-Fehler: bad argument #1 to "get" (not a valid title)</b> |
| <code>{{#invoke: BrewerColors ShowColors Oranges_i 5}}</code> | <b>Lua-Fehler: bad argument #1 to "get" (not a valid title)</b> |
| <code>{{BrewerColors Greens 9}}</code>                        | <b>Vorlage:BrewerColors</b>                                     |
| <code>{{#invoke: BrewerColors ShowColors YlGn 3 -}}</code>    | <b>Lua-Fehler: bad argument #1 to "get" (not a valid title)</b> |
| <code>{{#invoke: BrewerColors ShowColors YlGn 3 a}}</code>    | <b>Lua-Fehler: bad argument #1 to "get" (not a valid title)</b> |

### Function Legends

```
{{#invoke:BrewerColors|Legends|Oranges|5|30em}}
```

produces:

**Lua-Fehler: expandTemplate: template "div col" does not exist**

```
{{#invoke:BrewerColors|Legends|Greens|3||-Without data|One|Two|Three}}
```

produces:

**Lua-Fehler: bad argument #1 to "get" (not a valid title)**

```
local p = {}

local CFCM = require "Module:ComplForColorModules"
local TNTT = require "Module:TNTTools"
local ARG = require "Module:Arguments"
local I18n = 'BrewerColors'

--[[
TRANSLATION, in :Commons:Data:I18n/BrewerColors.tab
```



```
local RS_ColorNumInvalid = 'El nombre de colors ($1) està fora del rang'
local RS_ColorNameInvalid = 'El nom del color ($1) no és vàlid'
--]]
local RS_ColorNumInvalid = 'ColorNumInvalid'
local RS_ColorNameInvalid = 'ColorNameInvalid'

local function I18nStr (S, ...)
    return TTT.GetMsgP (I18n, S, {...})
end

-- This product includes ColorName specifications and designs developed by Cynthia
local ColorYlGn = {
    {"f7fcb9", "addd8e", "31a354"},
    {"ffffcc", "c2e699", "78c679", "238443"},
    {"ffffcc", "c2e699", "78c679", "31a354", "006837"},
    {"ffffcc", "d9f0a3", "addd8e", "78c679", "31a354", "006837"},
    {"ffffcc", "d9f0a3", "addd8e", "78c679", "41ab5d", "238443", "005a32"},
    {"ffffe5", "f7fcb9", "d9f0a3", "addd8e", "78c679", "41ab5d", "238443", "005a32"},
    {"ffffe5", "f7fcb9", "d9f0a3", "addd8e", "78c679", "41ab5d", "238443", "006837", "004
}
local ColorYlGnBu = {
    {"edf8b1", "7fcdbb", "2c7fb8"},
    {"ffffcc", "aldab4", "41b6c4", "225ea8"},
    {"ffffcc", "aldab4", "41b6c4", "2c7fb8", "253494"},
    {"ffffcc", "c7e9b4", "7fcdbb", "41b6c4", "2c7fb8", "253494"},
    {"ffffcc", "c7e9b4", "7fcdbb", "41b6c4", "1d91c0", "225ea8", "0c2c84"},
    {"ffffd9", "edf8b1", "c7e9b4", "7fcdbb", "41b6c4", "1d91c0", "225ea8", "0c2c84"},
    {"ffffd9", "edf8b1", "c7e9b4", "7fcdbb", "41b6c4", "1d91c0", "225ea8", "253494", "081
}
local ColorGnBu = {
    {"e0f3db", "a8ddb5", "43a2ca"},
    {"f0f9e8", "bae4bc", "7bcc4", "2b8cbe"},
    {"f0f9e8", "bae4bc", "7bcc4", "43a2ca", "0868ac"},
    {"f0f9e8", "cceb5", "a8ddb5", "7bcc4", "43a2ca", "0868ac"},
    {"f0f9e8", "cceb5", "a8ddb5", "7bcc4", "4eb3d3", "2b8cbe", "08589e"},
    {"f7fcf0", "e0f3db", "cceb5", "a8ddb5", "7bcc4", "4eb3d3", "2b8cbe", "08589e"},
    {"f7fcf0", "e0f3db", "cceb5", "a8ddb5", "7bcc4", "4eb3d3", "2b8cbe", "0868ac", "084
}
local ColorBuGn = {
    {"e5f5f9", "99d8c9", "2ca25f"},
    {"edf8fb", "b2e2e2", "66c2a4", "238b45"},
    {"edf8fb", "b2e2e2", "66c2a4", "2ca25f", "006d2c"},
    {"edf8fb", "ccece6", "99d8c9", "66c2a4", "2ca25f", "006d2c"},
    {"edf8fb", "ccece6", "99d8c9", "66c2a4", "41ae76", "238b45", "005824"},
    {"f7fcfd", "e5f5f9", "ccece6", "99d8c9", "66c2a4", "41ae76", "238b45", "005824"},
    {"f7fcfd", "e5f5f9", "ccece6", "99d8c9", "66c2a4", "41ae76", "238b45", "006d2c", "004
}
local ColorPuBuGn = {
    {"ece2f0", "a6bddb", "1c9099"},
    {"f6eff7", "bdc9e1", "67a9cf", "02818a"},
    {"f6eff7", "bdc9e1", "67a9cf", "1c9099", "016c59"},
    {"f6eff7", "d0d1e6", "a6bddb", "67a9cf", "1c9099", "016c59"},
    {"f6eff7", "d0d1e6", "a6bddb", "67a9cf", "3690c0", "02818a", "016450"},
    {"fff7fb", "ece2f0", "d0d1e6", "a6bddb", "67a9cf", "3690c0", "02818a", "016450"},
    {"fff7fb", "ece2f0", "d0d1e6", "a6bddb", "67a9cf", "3690c0", "02818a", "016c59", "014
}
local ColorPuBu = {
    {"ece7f2", "a6bddb", "2b8cbe"},
    {"fleef6", "bdc9e1", "74a9cf", "0570b0"},
    {"fleef6", "bdc9e1", "74a9cf", "2b8cbe", "045a8d"},
    {"fleef6", "d0d1e6", "a6bddb", "74a9cf", "2b8cbe", "045a8d"},
    {"fleef6", "d0d1e6", "a6bddb", "74a9cf", "3690c0", "0570b0", "034e7b"},
    {"fff7fb", "ece7f2", "d0d1e6", "a6bddb", "74a9cf", "3690c0", "0570b0", "034e7b"},
    {"fff7fb", "ece7f2", "d0d1e6", "a6bddb", "74a9cf", "3690c0", "0570b0", "045a8d", "024
```



```
}
local ColorBuPu = {
  {"e0ecf4", "9ebcda", "8856a7"},
  {"edf8fb", "b3cde3", "8c96c6", "88419d"},
  {"edf8fb", "b3cde3", "8c96c6", "8856a7", "810f7c"},
  {"edf8fb", "bfd3e6", "9ebcda", "8c96c6", "8856a7", "810f7c"},
  {"edf8fb", "bfd3e6", "9ebcda", "8c96c6", "8c6bb1", "88419d", "6e016b"},
  {"f7fcfd", "e0ecf4", "bfd3e6", "9ebcda", "8c96c6", "8c6bb1", "88419d", "6e016b"},
  {"f7fcfd", "e0ecf4", "bfd3e6", "9ebcda", "8c96c6", "8c6bb1", "88419d", "810f7c", "4d0
}
local ColorRdPu = {
  {"fde0dd", "fa9fb5", "c51b8a"},
  {"feebe2", "fbb4b9", "f768a1", "ae017e"},
  {"feebe2", "fbb4b9", "f768a1", "c51b8a", "7a0177"},
  {"feebe2", "fcc5c0", "fa9fb5", "f768a1", "c51b8a", "7a0177"},
  {"feebe2", "fcc5c0", "fa9fb5", "f768a1", "dd3497", "ae017e", "7a0177"},
  {"fff7f3", "fde0dd", "fcc5c0", "fa9fb5", "f768a1", "dd3497", "ae017e", "7a0177"},
  {"fff7f3", "fde0dd", "fcc5c0", "fa9fb5", "f768a1", "dd3497", "ae017e", "7a0177", "490
}
local ColorPuRd = {
  {"e7e1ef", "c994c7", "dd1c77"},
  {"fleef6", "d7b5d8", "df65b0", "ce1256"},
  {"fleef6", "d7b5d8", "df65b0", "dd1c77", "980043"},
  {"fleef6", "d4b9da", "c994c7", "df65b0", "dd1c77", "980043"},
  {"fleef6", "d4b9da", "c994c7", "df65b0", "e7298a", "ce1256", "91003f"},
  {"f7f4f9", "e7e1ef", "d4b9da", "c994c7", "df65b0", "e7298a", "ce1256", "91003f"},
  {"f7f4f9", "e7e1ef", "d4b9da", "c994c7", "df65b0", "e7298a", "ce1256", "980043", "670
}
local ColorOrRd = {
  {"fee8c8", "fdbb84", "e34a33"},
  {"fef0d9", "fdcc8a", "fc8d59", "d7301f"},
  {"fef0d9", "fdcc8a", "fc8d59", "e34a33", "b30000"},
  {"fef0d9", "fdd49e", "fdbb84", "fc8d59", "e34a33", "b30000"},
  {"fef0d9", "fdd49e", "fdbb84", "fc8d59", "ef6548", "d7301f", "990000"},
  {"fff7ec", "fee8c8", "fdd49e", "fdbb84", "fc8d59", "ef6548", "d7301f", "990000"},
  {"fff7ec", "fee8c8", "fdd49e", "fdbb84", "fc8d59", "ef6548", "d7301f", "b30000", "7f0
}
local ColorYlOrRd = {
  {"ffeda0", "feb24c", "f03b20"},
  {"ffffb2", "fecc5c", "fd8d3c", "e31a1c"},
  {"ffffb2", "fecc5c", "fd8d3c", "f03b20", "bd0026"},
  {"ffffb2", "fed976", "feb24c", "fd8d3c", "f03b20", "bd0026"},
  {"ffffb2", "fed976", "feb24c", "fd8d3c", "fc4e2a", "e31a1c", "b10026"},
  {"ffffcc", "ffeda0", "fed976", "feb24c", "fd8d3c", "fc4e2a", "e31a1c", "b10026"},
  {"ffffcc", "ffeda0", "fed976", "feb24c", "fd8d3c", "fc4e2a", "e31a1c", "bd0026", "800
}
local ColorYlOrBr = {
  {"fff7bc", "fec44f", "d95f0e"},
  {"ffffd4", "fed98e", "fe9929", "cc4c02"},
  {"ffffd4", "fed98e", "fe9929", "d95f0e", "993404"},
  {"ffffd4", "fee391", "fec44f", "fe9929", "d95f0e", "993404"},
  {"ffffd4", "fee391", "fec44f", "fe9929", "ec7014", "cc4c02", "8c2d04"},
  {"ffffe5", "fff7bc", "fee391", "fec44f", "fe9929", "ec7014", "cc4c02", "8c2d04"},
  {"ffffe5", "fff7bc", "fee391", "fec44f", "fe9929", "ec7014", "cc4c02", "993404", "662
}
local ColorPurples = {
  {"efedf5", "bcbddc", "756bb1"},
  {"f2f0f7", "cbc9e2", "9e9ac8", "6a51a3"},
  {"f2f0f7", "cbc9e2", "9e9ac8", "756bb1", "54278f"},
  {"f2f0f7", "dadaeb", "bcbddc", "9e9ac8", "756bb1", "54278f"},
  {"f2f0f7", "dadaeb", "bcbddc", "9e9ac8", "807dba", "6a51a3", "4a1486"},
  {"fcfbfd", "efedf5", "dadaeb", "bcbddc", "9e9ac8", "807dba", "6a51a3", "4a1486"},
  {"fcfbfd", "efedf5", "dadaeb", "bcbddc", "9e9ac8", "807dba", "6a51a3", "54278f", "3f0
}
```



```
local ColorBlues = {
  {"deebf7", "9ecae1", "3182bd"},
  {"eff3ff", "bdd7e7", "6baed6", "2171b5"},
  {"eff3ff", "bdd7e7", "6baed6", "3182bd", "08519c"},
  {"eff3ff", "c6dbef", "9ecae1", "6baed6", "3182bd", "08519c"},
  {"eff3ff", "c6dbef", "9ecae1", "6baed6", "4292c6", "2171b5", "084594"},
  {"f7fbff", "deebf7", "c6dbef", "9ecae1", "6baed6", "4292c6", "2171b5", "084594"},
  {"f7fbff", "deebf7", "c6dbef", "9ecae1", "6baed6", "4292c6", "2171b5", "08519c", "08519c"}
}

local ColorGreens = {
  {"e5f5e0", "ald99b", "31a354"},
  {"edf8e9", "bae4b3", "74c476", "238b45"},
  {"edf8e9", "bae4b3", "74c476", "31a354", "006d2c"},
  {"edf8e9", "c7e9c0", "ald99b", "74c476", "31a354", "006d2c"},
  {"edf8e9", "c7e9c0", "ald99b", "74c476", "41ab5d", "238b45", "005a32"},
  {"f7fcf5", "e5f5e0", "c7e9c0", "ald99b", "74c476", "41ab5d", "238b45", "005a32"},
  {"f7fcf5", "e5f5e0", "c7e9c0", "ald99b", "74c476", "41ab5d", "238b45", "006d2c", "006d2c"}
}

local ColorOranges = {
  {"fee6ce", "fdae6b", "e6550d"},
  {"feedde", "fdb8e5", "fd8d3c", "d94701"},
  {"feedde", "fdb8e5", "fd8d3c", "e6550d", "a63603"},
  {"feedde", "fdd0a2", "fdae6b", "fd8d3c", "e6550d", "a63603"},
  {"feedde", "fdd0a2", "fdae6b", "fd8d3c", "f16913", "d94801", "8c2d04"},
  {"fff5eb", "fee6ce", "fdd0a2", "fdae6b", "fd8d3c", "f16913", "d94801", "8c2d04"},
  {"fff5eb", "fee6ce", "fdd0a2", "fdae6b", "fd8d3c", "f16913", "d94801", "a63603", "7f7f7f"}
}

local ColorReds = {
  {"fee0d2", "fc9272", "de2d26"},
  {"fee5d9", "fcae91", "fb6a4a", "cb181d"},
  {"fee5d9", "fcae91", "fb6a4a", "de2d26", "a50f15"},
  {"fee5d9", "fcbb1a", "fc9272", "fb6a4a", "de2d26", "a50f15"},
  {"fee5d9", "fcbb1a", "fc9272", "fb6a4a", "ef3b2c", "cb181d", "99000d"},
  {"fff5f0", "fee0d2", "fcbb1a", "fc9272", "fb6a4a", "ef3b2c", "cb181d", "99000d"},
  {"fff5f0", "fee0d2", "fcbb1a", "fc9272", "fb6a4a", "ef3b2c", "cb181d", "a50f15", "676767"}
}

local ColorGreys = {
  {"f0f0f0", "bdbdbd", "636363"},
  {"f7f7f7", "cccccc", "969696", "525252"},
  {"f7f7f7", "cccccc", "969696", "636363", "252525"},
  {"f7f7f7", "d9d9d9", "bdbdbd", "969696", "636363", "252525"},
  {"f7f7f7", "d9d9d9", "bdbdbd", "969696", "737373", "525252", "252525"},
  {"ffffff", "f0f0f0", "d9d9d9", "bdbdbd", "969696", "737373", "525252", "252525"},
  {"ffffff", "f0f0f0", "d9d9d9", "bdbdbd", "969696", "737373", "525252", "252525", "000000"}
}

local ColorPuOr = {
  {"f1a340", "f7f7f7", "998ec3"},
  {"e66101", "fdb863", "b2abd2", "5e3c99"},
  {"e66101", "fdb863", "f7f7f7", "b2abd2", "5e3c99"},
  {"b35806", "f1a340", "fee0b6", "d8daeb", "998ec3", "542788"},
  {"b35806", "f1a340", "fee0b6", "f7f7f7", "d8daeb", "998ec3", "542788"},
  {"b35806", "e08214", "fdb863", "fee0b6", "d8daeb", "b2abd2", "8073ac", "542788"},
  {"b35806", "e08214", "fdb863", "fee0b6", "f7f7f7", "d8daeb", "b2abd2", "8073ac", "542788"},
  {"f73b08", "b35806", "e08214", "fdb863", "fee0b6", "d8daeb", "b2abd2", "8073ac", "542788"},
  {"f73b08", "b35806", "e08214", "fdb863", "fee0b6", "f7f7f7", "d8daeb", "b2abd2", "8073ac"}
}

local ColorBrBG = {
  {"d8b365", "f5f5f5", "5ab4ac"},
  {"a6611a", "dfc27d", "80cdc1", "018571"},
  {"a6611a", "dfc27d", "f5f5f5", "80cdc1", "018571"},
  {"8c510a", "d8b365", "f6e8c3", "c7eae5", "5ab4ac", "01665e"},
  {"8c510a", "d8b365", "f6e8c3", "f5f5f5", "c7eae5", "5ab4ac", "01665e"},
  {"8c510a", "bf812d", "dfc27d", "f6e8c3", "c7eae5", "80cdc1", "35978f", "01665e"},
  {"8c510a", "bf812d", "dfc27d", "f6e8c3", "f5f5f5", "c7eae5", "80cdc1", "35978f", "01665e"}
}
```



```
{ "543005", "8c510a", "bf812d", "dfc27d", "f6e8c3", "c7eae5", "80cdc1", "35978f", "010000"},  
{ "543005", "8c510a", "bf812d", "dfc27d", "f6e8c3", "f5f5f5", "c7eae5", "80cdc1", "35978f", "010000"},  
}  
local ColorPRGn = {  
  {"af8dc3", "f7f7f7", "7fbf7b"},  
  {"7b3294", "c2a5cf", "a6dba0", "008837"},  
  {"7b3294", "c2a5cf", "f7f7f7", "a6dba0", "008837"},  
  {"762a83", "af8dc3", "e7d4e8", "d9f0d3", "7fbf7b", "1b7837"},  
  {"762a83", "af8dc3", "e7d4e8", "f7f7f7", "d9f0d3", "7fbf7b", "1b7837"},  
  {"762a83", "9970ab", "c2a5cf", "e7d4e8", "d9f0d3", "a6dba0", "5aae61", "1b7837"},  
  {"762a83", "9970ab", "c2a5cf", "e7d4e8", "f7f7f7", "d9f0d3", "a6dba0", "5aae61", "1b7837"},  
  {"40004b", "762a83", "9970ab", "c2a5cf", "e7d4e8", "d9f0d3", "a6dba0", "5aae61", "1b7837"},  
  {"40004b", "762a83", "9970ab", "c2a5cf", "e7d4e8", "f7f7f7", "d9f0d3", "a6dba0", "5aae61", "1b7837"},  
}  
local ColorPiYG = {  
  {"e9a3c9", "f7f7f7", "a1d76a"},  
  {"d01c8b", "f1b6da", "b8e186", "4dac26"},  
  {"d01c8b", "f1b6da", "f7f7f7", "b8e186", "4dac26"},  
  {"c51b7d", "e9a3c9", "fde0ef", "e6f5d0", "a1d76a", "4d9221"},  
  {"c51b7d", "e9a3c9", "fde0ef", "f7f7f7", "e6f5d0", "a1d76a", "4d9221"},  
  {"c51b7d", "de77ae", "f1b6da", "fde0ef", "e6f5d0", "b8e186", "7fbc41", "4d9221"},  
  {"c51b7d", "de77ae", "f1b6da", "fde0ef", "f7f7f7", "e6f5d0", "b8e186", "7fbc41", "4d9221"},  
  {"8e0152", "c51b7d", "de77ae", "f1b6da", "fde0ef", "e6f5d0", "b8e186", "7fbc41", "4d9221"},  
  {"8e0152", "c51b7d", "de77ae", "f1b6da", "fde0ef", "f7f7f7", "e6f5d0", "b8e186", "7fbc41", "4d9221"},  
}  
local ColorRdBu = {  
  {"ef8a62", "f7f7f7", "67a9cf"},  
  {"ca0020", "f4a582", "92c5de", "0571b0"},  
  {"ca0020", "f4a582", "f7f7f7", "92c5de", "0571b0"},  
  {"b2182b", "ef8a62", "fddbc7", "d1e5f0", "67a9cf", "2166ac"},  
  {"b2182b", "ef8a62", "fddbc7", "f7f7f7", "d1e5f0", "67a9cf", "2166ac"},  
  {"b2182b", "d6604d", "f4a582", "fddbc7", "d1e5f0", "92c5de", "4393c3", "2166ac"},  
  {"b2182b", "d6604d", "f4a582", "fddbc7", "f7f7f7", "d1e5f0", "92c5de", "4393c3", "2166ac"},  
  {"67001f", "b2182b", "d6604d", "f4a582", "fddbc7", "d1e5f0", "92c5de", "4393c3", "2166ac"},  
  {"67001f", "b2182b", "d6604d", "f4a582", "fddbc7", "f7f7f7", "d1e5f0", "92c5de", "4393c3", "2166ac"},  
}  
local ColorRdGy = {  
  {"ef8a62", "ffffff", "999999"},  
  {"ca0020", "f4a582", "bababa", "404040"},  
  {"ca0020", "f4a582", "ffffff", "bababa", "404040"},  
  {"b2182b", "ef8a62", "fddbc7", "e0e0e0", "999999", "4d4d4d"},  
  {"b2182b", "ef8a62", "fddbc7", "ffffff", "e0e0e0", "999999", "4d4d4d"},  
  {"b2182b", "d6604d", "f4a582", "fddbc7", "e0e0e0", "bababa", "878787", "4d4d4d"},  
  {"b2182b", "d6604d", "f4a582", "fddbc7", "ffffff", "e0e0e0", "bababa", "878787", "4d4d4d"},  
  {"67001f", "b2182b", "d6604d", "f4a582", "fddbc7", "e0e0e0", "bababa", "878787", "4d4d4d"},  
  {"67001f", "b2182b", "d6604d", "f4a582", "fddbc7", "ffffff", "e0e0e0", "bababa", "878787", "4d4d4d"},  
}  
local ColorRdYlBu = {  
  {"fc8d59", "ffffbf", "91bafb"},  
  {"d7191c", "fdae61", "abd9e9", "2c7bb6"},  
  {"d7191c", "fdae61", "ffffbf", "abd9e9", "2c7bb6"},  
  {"d73027", "fc8d59", "fee090", "e0f3f8", "91bafb", "4575b4"},  
  {"d73027", "fc8d59", "fee090", "ffffbf", "e0f3f8", "91bafb", "4575b4"},  
  {"d73027", "f46d43", "fdae61", "fee090", "e0f3f8", "abd9e9", "74add1", "4575b4"},  
  {"d73027", "f46d43", "fdae61", "fee090", "ffffbf", "e0f3f8", "abd9e9", "74add1", "4575b4"},  
  {"a50026", "d73027", "f46d43", "fdae61", "fee090", "e0f3f8", "abd9e9", "74add1", "4575b4"},  
  {"a50026", "d73027", "f46d43", "fdae61", "fee090", "ffffbf", "e0f3f8", "abd9e9", "74add1", "4575b4"},  
}  
local ColorSpectral = {  
  {"fc8d59", "ffffbf", "99d594"},  
  {"d7191c", "fdae61", "abdda4", "2b83ba"},  
  {"d7191c", "fdae61", "ffffbf", "abdda4", "2b83ba"},  
  {"d53e4f", "fc8d59", "fee08b", "e6f598", "99d594", "3288bd"},  
  {"d53e4f", "fc8d59", "fee08b", "ffffbf", "e6f598", "99d594", "3288bd"},  
}
```



```
{ "d53e4f", "f46d43", "fdae61", "fee08b", "e6f598", "abdda4", "66c2a5", "3288bd" },
{ "d53e4f", "f46d43", "fdae61", "fee08b", "ffffbf", "e6f598", "abdda4", "66c2a5", "3288bd" },
{ "9e0142", "d53e4f", "f46d43", "fdae61", "fee08b", "e6f598", "abdda4", "66c2a5", "3288bd" },
{ "9e0142", "d53e4f", "f46d43", "fdae61", "fee08b", "ffffbf", "e6f598", "abdda4", "66c2a5" }
}
local ColorRdYlGn = {
  {"fc8d59", "ffffbf", "91cf60"},
  {"d7191c", "fdae61", "a6d96a", "1a9641"},
  {"d7191c", "fdae61", "ffffbf", "a6d96a", "1a9641"},
  {"d73027", "fc8d59", "fee08b", "d9ef8b", "91cf60", "1a9850"},
  {"d73027", "fc8d59", "fee08b", "ffffbf", "d9ef8b", "91cf60", "1a9850"},
  {"d73027", "f46d43", "fdae61", "fee08b", "d9ef8b", "a6d96a", "66bd63", "1a9850"},
  {"d73027", "f46d43", "fdae61", "fee08b", "ffffbf", "d9ef8b", "a6d96a", "66bd63", "1a9850"},
  {"a50026", "d73027", "f46d43", "fdae61", "fee08b", "d9ef8b", "a6d96a", "66bd63", "1a9850"},
  {"a50026", "d73027", "f46d43", "fdae61", "fee08b", "ffffbf", "d9ef8b", "a6d96a", "66bd63"}
}
local ColorAccent = {
  {"7fc97f", "beaed4", "fdc086"},
  {"7fc97f", "beaed4", "fdc086", "ffff99"},
  {"7fc97f", "beaed4", "fdc086", "ffff99", "386cb0"},
  {"7fc97f", "beaed4", "fdc086", "ffff99", "386cb0", "f0027f"},
  {"7fc97f", "beaed4", "fdc086", "ffff99", "386cb0", "f0027f", "bf5b17"},
  {"7fc97f", "beaed4", "fdc086", "ffff99", "386cb0", "f0027f", "bf5b17", "666666"},
}
local ColorDark2 = {
  {"1b9e77", "d95f02", "7570b3"},
  {"1b9e77", "d95f02", "7570b3", "e7298a"},
  {"1b9e77", "d95f02", "7570b3", "e7298a", "66a61e"},
  {"1b9e77", "d95f02", "7570b3", "e7298a", "66a61e", "e6ab02"},
  {"1b9e77", "d95f02", "7570b3", "e7298a", "66a61e", "e6ab02", "a6761d"},
  {"1b9e77", "d95f02", "7570b3", "e7298a", "66a61e", "e6ab02", "a6761d", "666666"},
}
local ColorPaired = {
  {"a6cee3", "1f78b4", "b2df8a"},
  {"a6cee3", "1f78b4", "b2df8a", "33a02c"},
  {"a6cee3", "1f78b4", "b2df8a", "33a02c", "fb9a99"},
  {"a6cee3", "1f78b4", "b2df8a", "33a02c", "fb9a99", "e31a1c"},
  {"a6cee3", "1f78b4", "b2df8a", "33a02c", "fb9a99", "e31a1c", "fdbf6f"},
  {"a6cee3", "1f78b4", "b2df8a", "33a02c", "fb9a99", "e31a1c", "fdbf6f", "ff7f00"},
  {"a6cee3", "1f78b4", "b2df8a", "33a02c", "fb9a99", "e31a1c", "fdbf6f", "ff7f00", "cab2d6"},
  {"a6cee3", "1f78b4", "b2df8a", "33a02c", "fb9a99", "e31a1c", "fdbf6f", "ff7f00", "cab2d6", "fdbf6f"},
  {"a6cee3", "1f78b4", "b2df8a", "33a02c", "fb9a99", "e31a1c", "fdbf6f", "ff7f00", "cab2d6", "fdbf6f", "ff7f00"},
}
local ColorPastel1 = {
  {"fbb4ae", "b3cde3", "cceb5"},
  {"fbb4ae", "b3cde3", "cceb5", "decbe4"},
  {"fbb4ae", "b3cde3", "cceb5", "decbe4", "fed9a6"},
  {"fbb4ae", "b3cde3", "cceb5", "decbe4", "fed9a6", "ffffcc"},
  {"fbb4ae", "b3cde3", "cceb5", "decbe4", "fed9a6", "ffffcc", "e5d8bd"},
  {"fbb4ae", "b3cde3", "cceb5", "decbe4", "fed9a6", "ffffcc", "e5d8bd", "fddaec"},
  {"fbb4ae", "b3cde3", "cceb5", "decbe4", "fed9a6", "ffffcc", "e5d8bd", "fddaec", "f28e2b"}
}
local ColorPastel2 = {
  {"b3e2cd", "fcdac", "cbd5e8"},
  {"b3e2cd", "fcdac", "cbd5e8", "f4cae4"},
  {"b3e2cd", "fcdac", "cbd5e8", "f4cae4", "e6f5c9"},
  {"b3e2cd", "fcdac", "cbd5e8", "f4cae4", "e6f5c9", "fff2ae"},
  {"b3e2cd", "fcdac", "cbd5e8", "f4cae4", "e6f5c9", "fff2ae", "f1e2cc"},
  {"b3e2cd", "fcdac", "cbd5e8", "f4cae4", "e6f5c9", "fff2ae", "f1e2cc", "cccccc"},
}
local ColorSet1 = {
  {"e41a1c", "377eb8", "4daf4a"},
  {"e41a1c", "377eb8", "4daf4a", "984ea3"},
}
```



```
{ "e41a1c", "377eb8", "4daf4a", "984ea3", "ff7f00" },
{ "e41a1c", "377eb8", "4daf4a", "984ea3", "ff7f00", "ffff33" },
{ "e41a1c", "377eb8", "4daf4a", "984ea3", "ff7f00", "ffff33", "a65628" },
{ "e41a1c", "377eb8", "4daf4a", "984ea3", "ff7f00", "ffff33", "a65628", "f781bf" },
{ "e41a1c", "377eb8", "4daf4a", "984ea3", "ff7f00", "ffff33", "a65628", "f781bf", "996633" }
}
local ColorSet2 = {
  {"66c2a5", "fc8d62", "8da0cb"},
  {"66c2a5", "fc8d62", "8da0cb", "e78ac3"},
  {"66c2a5", "fc8d62", "8da0cb", "e78ac3", "a6d854"},
  {"66c2a5", "fc8d62", "8da0cb", "e78ac3", "a6d854", "ffd92f"},
  {"66c2a5", "fc8d62", "8da0cb", "e78ac3", "a6d854", "ffd92f", "e5c494"},
  {"66c2a5", "fc8d62", "8da0cb", "e78ac3", "a6d854", "ffd92f", "e5c494", "b3b3b3"},
}
local ColorSet3 = {
  {"8dd3c7", "ffffb3", "bebada"},
  {"8dd3c7", "ffffb3", "bebada", "fb8072"},
  {"8dd3c7", "ffffb3", "bebada", "fb8072", "80b1d3"},
  {"8dd3c7", "ffffb3", "bebada", "fb8072", "80b1d3", "fdb462"},
  {"8dd3c7", "ffffb3", "bebada", "fb8072", "80b1d3", "fdb462", "b3de69"},
  {"8dd3c7", "ffffb3", "bebada", "fb8072", "80b1d3", "fdb462", "b3de69", "fccde5"},
  {"8dd3c7", "ffffb3", "bebada", "fb8072", "80b1d3", "fdb462", "b3de69", "fccde5", "d9d9d9"},
  {"8dd3c7", "ffffb3", "bebada", "fb8072", "80b1d3", "fdb462", "b3de69", "fccde5", "d9d9d9"},
  {"8dd3c7", "ffffb3", "bebada", "fb8072", "80b1d3", "fdb462", "b3de69", "fccde5", "d9d9d9"},
  {"8dd3c7", "ffffb3", "bebada", "fb8072", "80b1d3", "fdb462", "b3de69", "fccde5", "d9d9d9"}
}

function p.GetColors(ColorName, IsInv, N, CallError)
  local ColorFound = {}
  function GetC(ColorSet)
    if (N < 3) or (N > (table.getn(ColorSet)+2)) then
      if CallError then
        error (I18nStr(RS_ColorNumInvalid,tostring(N)), 0)
      end
    elseif IsInv then
      for i=table.getn(ColorSet[N-2]), 1, -1 do
        table.insert (ColorFound, '#'..ColorSet[N-2][i])
      end
    else
      for i=1, table.getn(ColorSet[N-2]) do
        table.insert (ColorFound, '#'..ColorSet[N-2][i])
      end
    end
  end
  local ColorNameL = string.lower (ColorName)
  if ColorNameL == 'ylgn' then GetC(ColorYlGn)
  elseif ColorNameL == 'ylgnbu' then GetC(ColorYlGnBu)
  elseif ColorNameL == 'gnbu' then GetC(ColorGnBu)
  elseif ColorNameL == 'bugn' then GetC(ColorBuGn)
  elseif ColorNameL == 'pubugn' then GetC(ColorPuBuGn)
  elseif ColorNameL == 'pubu' then GetC(ColorPuBu)
  elseif ColorNameL == 'bupu' then GetC(ColorBuPu)
  elseif ColorNameL == 'rdpu' then GetC(ColorRdPu)
  elseif ColorNameL == 'purd' then GetC(ColorPuRd)
  elseif ColorNameL == 'orrd' then GetC(ColorOrRd)
  elseif ColorNameL == 'ylorrd' then GetC(ColorYlOrRd)
  elseif ColorNameL == 'ylorbr' then GetC(ColorYlOrBr)
  elseif ColorNameL == 'purples' then GetC(ColorPurples)
  elseif ColorNameL == 'blues' then GetC(ColorBlues)
  elseif ColorNameL == 'greens' then GetC(ColorGreens)
  elseif ColorNameL == 'oranges' then GetC(ColorOranges)
  elseif ColorNameL == 'reds' then GetC(ColorReds)
  elseif ColorNameL == 'greys' then GetC(ColorGreys)
  elseif ColorNameL == 'puor' then GetC(ColorPuOr)
```



```
elseif ColorNameL == 'brbg' then GetC(ColorBrBG)
elseif ColorNameL == 'prgn' then GetC(ColorPRGn)
elseif ColorNameL == 'piyg' then GetC(ColorPiYG)
elseif ColorNameL == 'rdbu' then GetC(ColorRdBu)
elseif ColorNameL == 'rdgy' then GetC(ColorRdGy)
elseif ColorNameL == 'rdylbu' then GetC(ColorRdYlBu)
elseif ColorNameL == 'spectral' then GetC(ColorSpectral)
elseif ColorNameL == 'rdylgn' then GetC(ColorRdYlGn)
elseif ColorNameL == 'accent' then GetC(ColorAccent)
elseif ColorNameL == 'dark2' then GetC(ColorDark2)
elseif ColorNameL == 'paired' then GetC(ColorPaired)
elseif ColorNameL == 'pastel1' then GetC(ColorPastel1)
elseif ColorNameL == 'pastel2' then GetC(ColorPastel2)
elseif ColorNameL == 'set1' then GetC(ColorSet1)
elseif ColorNameL == 'set2' then GetC(ColorSet2)
elseif ColorNameL == 'set3' then GetC(ColorSet3)
else
    if CallError then
        error (I18nStr (RS_ColorNameInvalid,ColorName), 0)
    end
end
return ColorFound
end

function p.ShowColors(frame)
    local args = ARG.getArgs(frame, {
        removeBlanks = false
    })
    local ColorName, IsInv = CFCM.ColorNameInv (args)
    local N = tonumber(args[2])
    local WriteColor = args[3] or ''
    local ColorFound = {}
    ColorFound = p.GetColors (ColorName, IsInv, N, true)
    local boxes = {}
    for i=1, table.getn(ColorFound) do
        table.insert(boxes, CFCM.Box(ColorFound[i],WriteColor))
    end
    return table.concat(boxes, " ")
end

function p.Legends(frame)
    local args = ARG.getArgs(frame, {
        removeBlanks = false
    })
    local ColorName, IsInv = CFCM.ColorNameInv (args)
    local N = tonumber(args[2])
    local ColWidth = args[3]
    local ColorFound = {}
    local Labels = {}
    local NLabels = 0
    local IsTemplate = true
    ColorFound = p.GetColors (ColorName, IsInv, N, true)
    Labels, NLabels, OutlineColor = CFCM.GetLabels (args, N, 4)
    return CFCM.LegendText (ColorFound, Labels, NLabels, ColWidth, IsTemplate)
end

return p
```

## Modul:ChartColors

Used by **Vorlage:TI** and **Vorlage:TI**

To translate or review the translation of the module to your language, edit carefully [Data:118n/ChartColors.tab](#).

### Function ShowColors

| Code  | Result  |
|---|---|
| <code>{{#invoke:ChartColors ShowColors Categ20 5}}</code>   | <b>Lua-Fehler: bad argument #1 to "get" (not a valid title)</b> |
| <code>{{ChartColors Categ20 9}}</code>                      | <b>Lua-Fehler: bad argument #1 to "get" (not a valid title)</b> |
| <code>{{#invoke:ChartColors ShowColors Categ20 3 -}}</code> | <b>Lua-Fehler: bad argument #1 to "get" (not a valid title)</b> |
| <code>{{#invoke:ChartColors ShowColors Categ20 3 a}}</code> | <b>Lua-Fehler: bad argument #1 to "get" (not a valid title)</b> |
| <code>{{#invoke:ChartColors ShowColors Plotter 5}}</code>   | <b>Lua-Fehler: bad argument #1 to "get" (not a valid title)</b> |
| <code>{{ChartColors Plotter 20}}</code>                     | <b>Lua-Fehler: bad argument #1 to "get" (not a valid title)</b> |

### Function Legends

```
{{#invoke:ChartColors|Legends|Categ20|5|2}}
```

produces:

**Lua-Fehler: expandTemplate: template "div col" does not exist**

```
{{#invoke:ChartColors|Legends|Categ20|3|1|One|Two|Three}}
```

produces:

**Lua-Fehler: expandTemplate: template "div col" does not exist**

```
local p = {}  
  
local ARG = require "Module:Arguments"  
local CFCM = require "Module:ComplForColorModules"
```

```
local TNTT = require "Module:TNTTools"
--local SD = require "Module:SimpleDebug"

local I18n = 'ChartColors'

local function I18nStr (S, ...)
    return TNTT.GetMsgP (I18n, S, {...})
end

--local RS_ColorNameInvalid = 'El nom del color (%s) no és vàlid (ha de ser Categ
--local RS_ColorNumInvalid = 'El nombre de colors hauria d'estar entre 1 i %d (a
--local RS_StartNumInvalid = 'El número del primer color hauria d'estar entre 1 i

local category20 = {
    '#1f77b4', '#aec7e8', '#ff7f0e', '#ffbb78', '#2ca02c', '#98df8a', '#d62728',
    '#8c564b', '#c49c94', '#e377c2', '#f7b6d2', '#7f7f7f', '#c7c7c7', '#bcbd22'
}
local plotter = {--from Module:Plotter/DefaultColors
    'red', 'blue', 'green', 'yellow', 'fuchsia', 'aqua', 'brown', 'orange',
    '#F0A3FF', '#191919', '#005C31', 'honeydew', '#808080', 'khaki', 'lime',
    '#5EF1F2', 'turquoise', '#E0FF66', 'violet', '#FFFF80', '#FF5005',
}

function p.GetColors (ColorName, IsInv, StartN, N, CallError)
    local SelColors = {}
    local ColorNameL = string.lower(ColorName)
    local Palet0 = {}
    if ColorNameL == 'categ20' then
        Palet0 = category20
    elseif ColorNameL == 'plotter' then
        Palet0 = plotter
    elseif CallError then
        error (I18nStr ('ColorNameInvalid',ColorName))
    end
    MaxN = table.getn(Palet0)
    if (N < 0) or (N > MaxN) then
        error (I18nStr ('ColorNumInvalid',tostring(MaxN),tostring(N)))
    else
        Palet = {}
        if IsInv then
            for i=MaxN, 1, -1 do
                table.insert(Palet, Palet0[i])
            end
        else
            Palet = Palet0
        end
        if StartN > MaxN then
            error (I18nStr ('StartNumInvalid',tostring(MaxN),tostring(StartN)))
        end
        local Step = 1
        local NEnd = N
        if (ColorNameL == 'categ20') and (N<=10) then
            Step = 2
            if StartN == 1 then
                NEnd = N*2
            end
        end
        if StartN == 1 then
            for i=1, NEnd, Step do
                table.insert(SelColors, Palet[i])
            end
        else
            Count = 0
            for i=StartN, MaxN, Step do
                Count = Count + 1
                table.insert(SelColors, Palet[i])
            end
        end
    end
end
```

```
        table.insert(SelColors, Palet[i])
        Count = Count + 1
        if Count == NEnd then
            break
        end
    end
    if Count < NEnd then
        for i=1, MaxN, Step do
            table.insert(SelColors, Palet[i])
            Count = Count + 1
            if Count == NEnd then
                break
            end
        end
    end
    end
    return SelColors
end --GetColors

function p.ColorNameInvStartFromS (S)
    local StartN = 1
    local ParamsA = {}
    local ColorName, IsInv
    ColorName, IsInv, ParamsA = CFCM.ColorNameInvFromS0 (S)
    local PN = table.getn(ParamsA)
    if PN > 1 then
        if ParamsA[2]=='i' then
            IsInv = true
        else
            StartN = tonumber (ParamsA[2])
        end
        if PN == 3 then
            StartN = tonumber (ParamsA[3])
        end
    end
    end
    return ColorName, IsInv, StartN
end --ColorNameInvStartFromS

function ColorNameInv (args)
    local S = args[1] or ''
    local ColorName, IsInv, ParamsA = CFCM.ColorNameInvFromS0 (S)
    local StartN = 1
    ColorName, IsInv, StartN = p.ColorNameInvStartFromS (S)
    return ColorName, IsInv, StartN
end --ColorNameInv

function p.ShowColors(frame)
    local args = ARG.getArgs(frame,{
        removeBlanks = false
    });
    local ColorName, IsInv, StartN = ColorNameInv (args)
    local N = tonumber(args[2])
    local WriteColor = args[3] or ""
    local ColorFound = {}
    ColorFound = p.GetColors (ColorName, IsInv, StartN, N, true)
    local boxes = {}
    for i=1, table.getn(ColorFound) do
        table.insert(boxes, CFCM.Box(ColorFound[i],WriteColor))
    end
    return table.concat(boxes, " ")
end --ShowColors

function p.Legends(frame)
```



```
    local args = ARG.getArgs(frame,{
        removeBlanks = false
    })
    local Nargs = require("Module:TableTools").length(args)
    local ColorName, IsInv, StartN = ColorNameInv (args)
    local N = tonumber(args[2])
    local ColWidth = args[3]
    local ColorFound = {}
    local Labels = {}
    local NLabels = 0
    local IsTemplate = true
    ColorFound = p.GetColors (ColorName, IsInv, StartN, N, true)
    Labels, NLabels, OutlineColor = CFCM.GetLabels (args, N, 4)
    return CFCM.LegendText (ColorFound, Labels, NLabels, ColWidth, IsTemplate)
end

return p
```

## Modul:ComplForColorModules

This module is used by **Module:BrewerColors** and **Module:ChartColors**.

To translate or review the translation of the module to your language, edit carefully [Data:I18n/ComplForColorModules.tab](#).

```
local p = {}

local TNTT = require "Module:TNTTools"

local I18n = 'ComplForColorModules'

--local MoreOneNoData = "Found 2 legends of \"Value not assigned\" or \"Data not
--local FoundNLegendsExpectedN = "Found $1 legends, expected $2"
--local Color = "Color"
--local Legend = "legend"

local function I18nStr (S, ...)
    return TNTT.GetMsgP (I18n, S, {...})
end

function p.ColorNameInvFromS0 (S)
    local IsInv = false
    local ColorName = ''
    local Params = {}
    if S ~= '' then
        for w in S:gmatch("[^_]+") do
            table.insert(Params, w)
        end
        ColorName = Params[1]
    end
    local PN = table.getn(Params)
    if (PN == 2) and (Params[2]=='i') then
        IsInv = true
    end
    return ColorName, IsInv, Params
end

function p.ColorNameInv (args)
    local ColorName, IsInv, Params = p.ColorNameInvFromS0(args[1])
    return ColorName, IsInv
end

function p.GetLabels(Args, N, Pos)
    local Labels = {}
    local index = Pos
    while Args[index] do
        Labels[#Labels+1] = Args[index]
        index = index + 1
    end
    local NLabels = #Labels
    local OutlineColor = Args['outline'] or ''
    if (NLabels ~= 0) and (NLabels ~= N) then
        local StartUnknown = (string.sub(Labels[1],1,2) == "--")
        local EndUnknown = (string.sub(Labels[NLabels],1,2) == "--")
        if StartUnknown and EndUnknown then
            error (I18nStr ('MoreOneNoData'))
        elseif StartUnknown or EndUnknown then
```

```
        N = N + 1
    end
    if (NLabels ~= 0) and (NLabels ~= N) then
        error (I18nStr ('FoundNLegendsExpectedN', tostring(NLabels)),
    end
end
return Labels, NLabels, OutlineColor
end

function SColor (Color)
    if string.sub(Color,1,1) == '#' then --the usual
        return string.sub(Color,2,100)
    else
        return Color
    end
end

function p.Box(Color,WriteColor)
    if WriteColor == '-' then
        WriteColor = Color
    elseif WriteColor == 'a' then
        WriteColor = Color..'ff' --Alpha channel
    end
    local TheBox = '<span style="background-color: '..Color..'"; border:1px so
    if WriteColor ~= '' then
        TheBox = TheBox..'&nbsp; '..WriteColor..' &nbsp;';
    end
    return TheBox
end

function p.TextWithTooltip (Text, Tooltip)
    if Tooltip ~= '' then
        return '<span title=" '..Tooltip..' ">'..Text..'</span>'
    else
        return Text
    end
end

function p.LegendColor(Color, Text, Tooltip)
    if Text == '' then
        Text = SColor(Color)
    end
    return '<li style="list-style-type: none; list-style-image: none;"><span
end

function LegendCode(Color, Text, OutlineColor)
    local SOutlineColor = ''
    if OutlineColor ~= '' then
        SOutlineColor = '|outline='..OutlineColor
    end
    return '{{ '..TNTT.GetStrP(I18n, 'Legend')..' | '..Color..' | '..Text..' SOutline
end

function p.LegendText (AColors, Labels, NLabels, ColWidth, IsTemplate, OutlineCo
    if ColWidth then
        ColWidth = mw.text.trim(ColWidth)
        if #ColWidth == 0 then
            ColWidth = nil
        end
    end
    local Show, Codes = {}, {}
    local Text = ''
    local WithLabels = NLabels > 0
    if WithLabels then
```



```
        local Gray = '#b3b3b3' -- A 40% gray
        if string.sub(Labels[1],1,2) == '--' then
            table.insert(AColors, 1, Gray)
            Labels[1] = string.sub(Labels[1], 3, 1000)
        elseif string.sub(Labels[NLabels],1,2) == '--' then
            table.insert(AColors, Gray)
            Labels[NLabels] = string.sub(Labels[NLabels], 3, 1000)
        end
    end
end
for i=1, table.getn(AColors) do
    if WithLabels then Text = Labels[i] end
    table.insert(Show, p.LegendColor(AColors[i],Text, ''))
end
SShow = table.concat(Show, "\n")
if ColWidth then
    local frame = mw.getCurrentFrame()
    SShow = frame:expandTemplate{title="div col",args={colwidth=ColWidth}}
    SShow = SShow .. frame:expandTemplate{title="div col end"}
end
for i=1, table.getn(AColors) do
    if WithLabels then Text = Labels[i] end
    table.insert(Codes, ' ' .. LegendCode(AColors[i],Text,OutlineColor))
end
local SCodes = '<pre>\n'
if ColWidth then
    SCodes = SCodes .. require("Module:Template invocation").invocation
end
SCodes = SCodes .. table.concat(Codes, "\n")
if ColWidth and #ColWidth ~= 0 then
    SCodes = SCodes .. "\n{{div col end}}"
end
SCodes = SCodes .. "\n</pre>"
return SShow..' \n' .. SCodes
end
return p
```