

Programmier- und Skriptsprachen

Tom Gries



Dokumenten URL:

<http://docs.tx7.de/TT-PP5>

Autor:

Tom Gries <TT-PP5@tx7.de>
@tomo@chaos.social

Lizenz:

Creative Commons BY-NC-ND

Version:

7.2.3 vom 16.01.2025



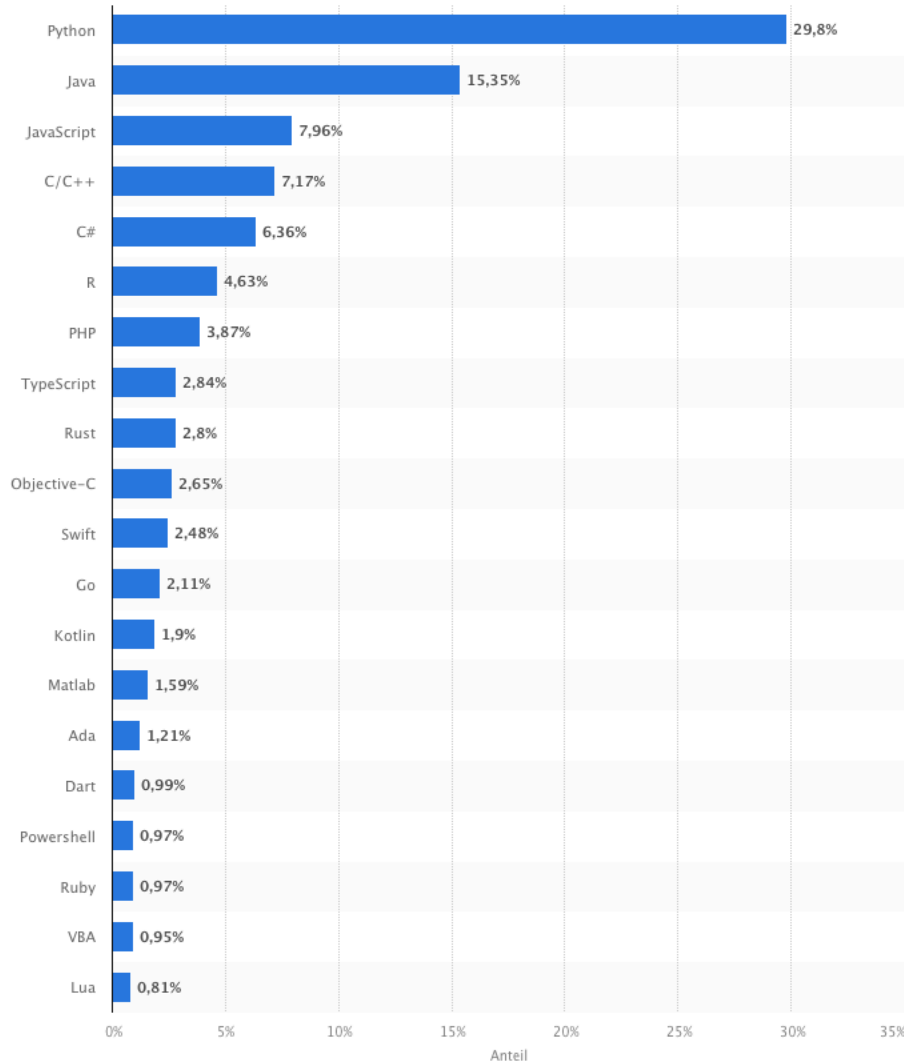


Programmier- und Skriptsprachen

(X)HTML/CSS	Der "Großvater" (statisch) im Internet. Bei (X)HTML handelt es sich nicht um eine Programmiersprache.
CGI/PHP/ASP	Für dynamische Inhalte (auf der Server Seite).
JavaScript/TypeScript	Für dynamische und responsive Inhalte (auf der Client Seite).
Python	Universell Skriptsprache mit Frameworks für Website Entwicklung (Flask, Django und weitere).
SQL	Die meistverwendete Sprache für Datenbanken.
Scratch	Visuelle Programmiersprache. In Raspberry OS bereits enthalten. Andere Implementierung auf OpenRoberta .
C/C++	C++ (1985) ist eine Erweiterung von C (1972).
Go	Von Google entwickelt. Soll Nachteile von C/C++ beheben.
Bash	Skriptsprache für die Linux Shell BASH (Burne again Shell).



Programmier- und Skriptsprachen



Python ist seit Jahren auf Platz 1. Es ist kostenlos, leicht zu erlernen, auf allen wichtigen Plattformen verfügbar, funktioniert auch plattformübergreifend und hat eine sehr große Community.

(X)HTML und CSS sind keine Programmiersprachen und daher hier auch nicht aufgeführt.



(X)HTML Beispiel zum Ausprobieren

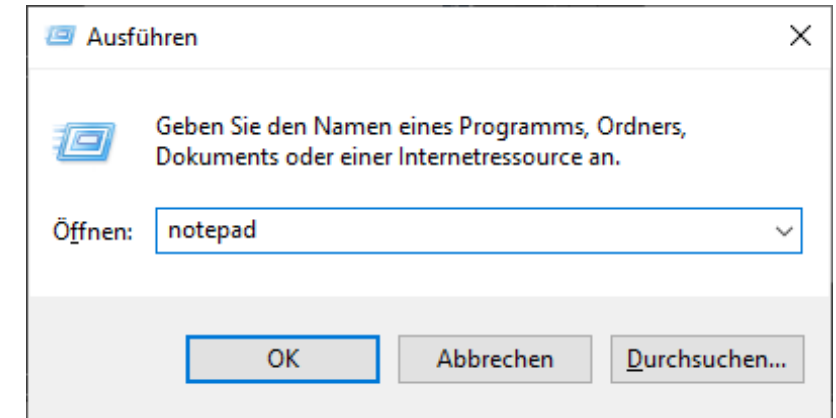
```
<html>
  <head>
    <title>A basic HTML document</title>
  </head>

  <body>
    This is a <!-- very --> simple HTML document.
  </body>
</html>
```



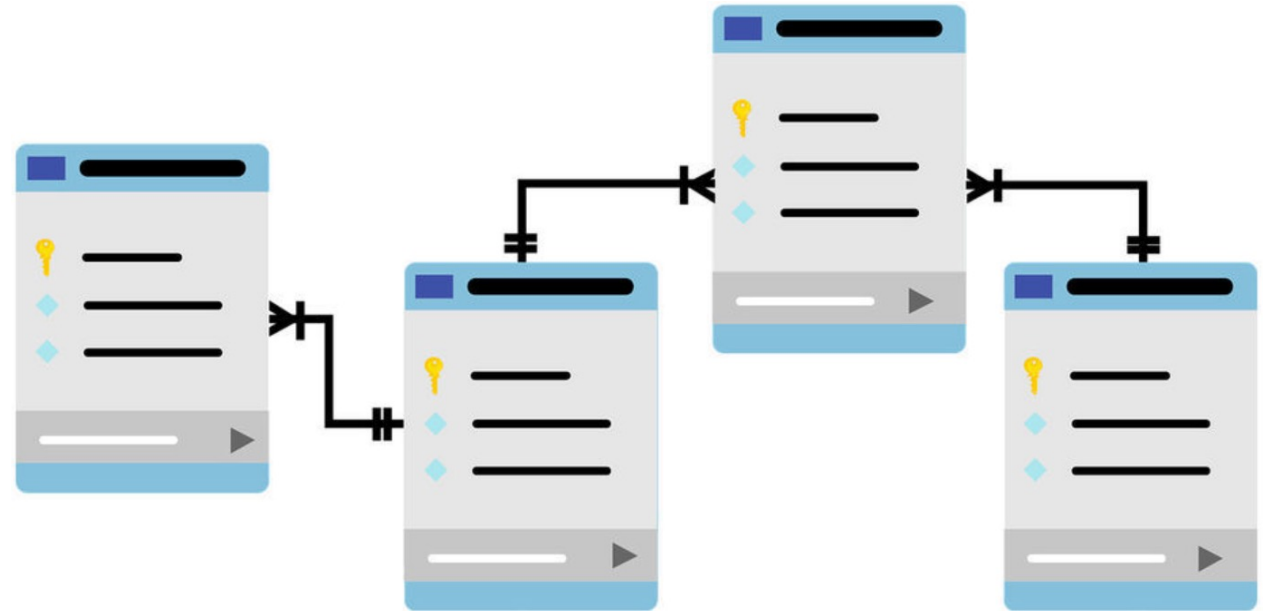
(X)HTML Beispiel zum Ausprobieren

- Editor öffnen (WIN + R: notepad)
- (X)HTML Grundgerüst der vorherigen Seite einfügen
- Datei auf dem Desktop unter dem Namen "index.html" speichern und durch Doppelklick öffnen





```
SELECT
    username, password
FROM
    user
WHERE
    username = "Armin";
```



	username	password
1	Armin	Frikadelle



```
<html>
  <head>
    <title>Hello World with PHP</title>
  </head>

  <body>
    <?php echo "Hello World!"; ?>
  </body>
</html>
```



Oder ein QR-Code Generator ...

```
# Define input variables
```

```
n1 = input("First number: ")
```

```
n2 = input("\nSecond number: ")
```

```
# Adding two integers
```

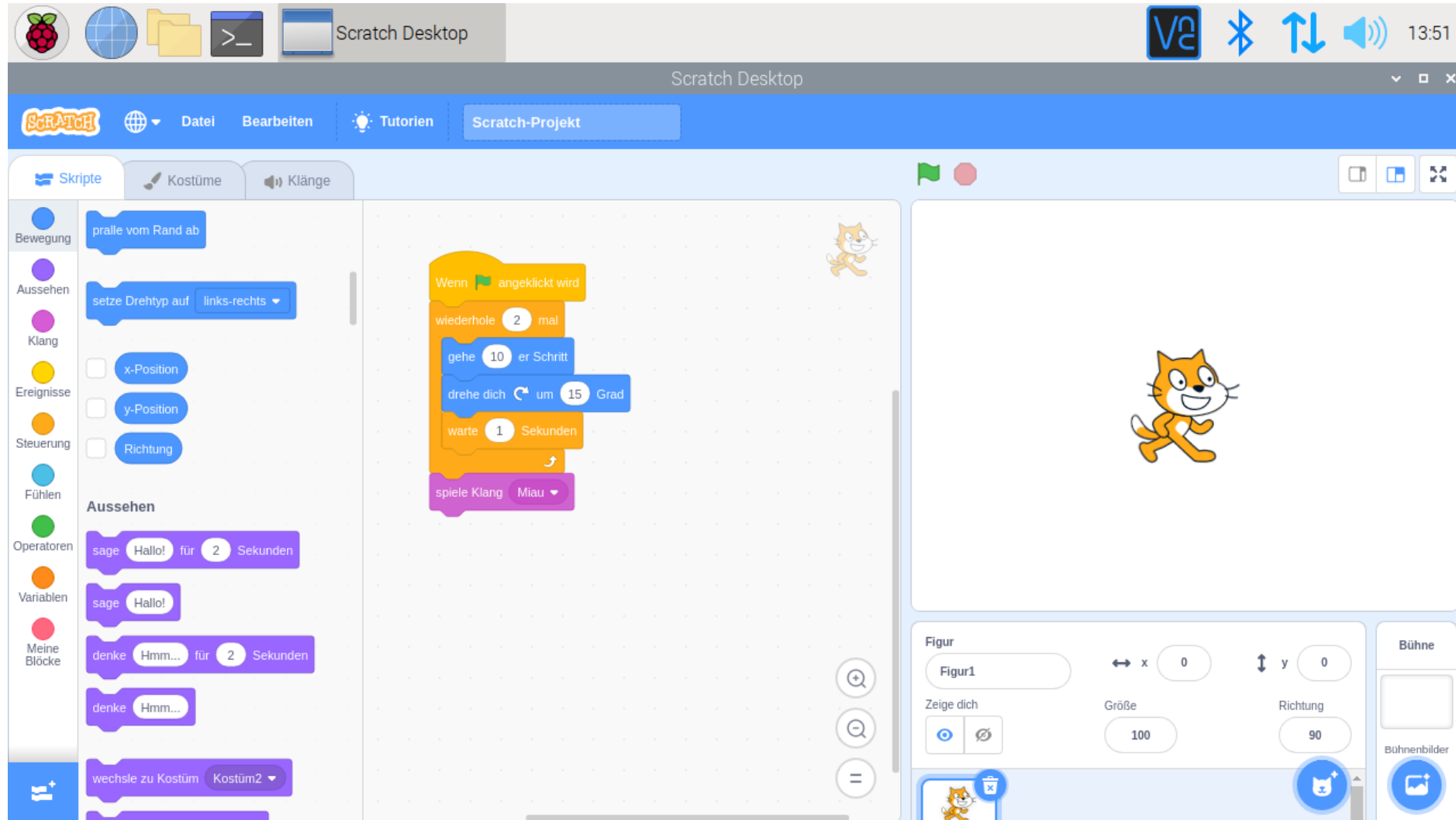
```
sum = int(n1) + int(n2)
```

```
# Display the sum
```

```
print("{0} and {1} is {2}".format(n1, n2, sum))
```



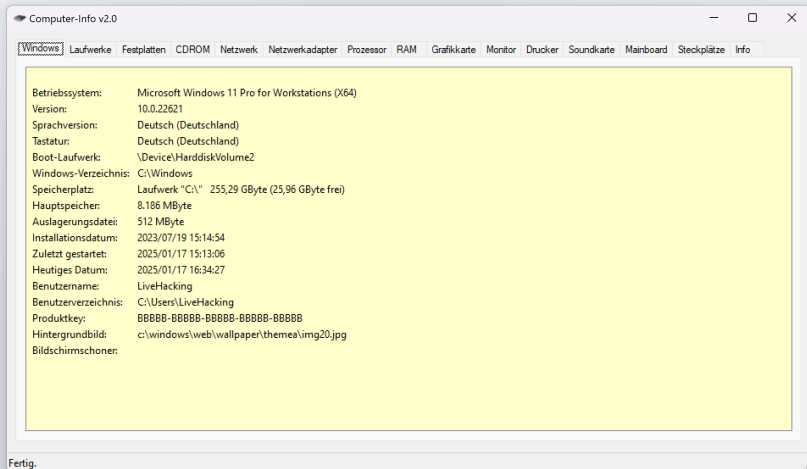

Scratch 3 auf dem Raspberry Pi - Für Kids und Einsteiger





```
C:\Users\LiveHacking\OneDrive\LiveHacking\Tools\AutoIt3\EigeneWerke\Computer-Info.au3 - SciTE
File Edit Search View Tools Options Language Buffers Help

1 #Region ;**** Directives created by AutoIt3Wrapper_GUI ****
2 #AutoIt3Wrapper_Ico=prog.ico
3 #AutoIt3Wrapper_Outfile=Computer-Info.exe
4 #AutoIt3Wrapper_UseUpX=n
5 #EndRegion ;**** Directives created by AutoIt3Wrapper_GUI ****
6
7 #NoTrayIcon
8 #Region ;**** Includes created by OrganizeIncludes ****
9 #include <Constants.au3>
10 #include <Date.au3>
11 #include <Inet.au3>
12 #include <GuiListView.au3>
13 #include <ListViewConstants.au3>
14 #include <GUIConstantsEx.au3>
15 #include <GuiStatusBar.au3>
16 #include <WindowsConstants.au3>
17 #EndRegion ;**** Includes created by OrganizeIncludes ****
18
19 Opt('GUIOnEventMode', 1)
20 Opt('MustDeclareVars', 1)
21
22 Global $sTitle = 'Computer-Info v2.0'
23 Global $hGUI, $hTab, $hStatus, $iWidth
24 Global $aTabNames[15] = [ _
25     'Windows', 'Laufwerke', 'Festplatten', 'CDROM', 'Netzwerk', _
26     'Netzwerkadapter', 'Prozessor', 'RAM', 'Grafikkarte', 'Monitor', _
27     'Drucker', 'Soundkarte', 'Mainboard', 'Steckplätze', 'Info' _
28 ]
29 Global $aTabIds[UBound($aTabNames)], $aContent[UBound($aTabNames)]
30
31 If @OSType <> 'WIN32_NT' Then Exit MsgBox(48, $sTitle, 'Das Programm funktioniert erst ab Windows 2000')
32
33 If (Not @Compiled) And (@AutoItVersion < '3.3.0.0') Then
34     Global $Msg = 'Dieses Script benötigt min. die AutoIt-Version: 3.3.0.0 ' & @CRLF & @CRLF
35     $Msg &= 'Gestartet wurde es mit AutoIt-Version: ' & @AutoItVersion
36     Exit MsgBox(48, $sTitle, $Msg)
37 EndIf
38
39 $iWidth = @DesktopWidth - 20
40 If $iWidth > 950 Then $iWidth = 950
41
42 $hGUI = GUICreate($sTitle, $iWidth, 520, Default, 120, BitOR($GUI_SS_DEFAULT_GUI, $WS_MAXIMIZEBOX, $WS_SIZEBOX))
43 GUISetOnEvent($GUI_EVENT_CLOSE, '_End')
44 If FileExists(@ScriptDir & '\prog.ico') Then
45     GUISetIcon(@ScriptDir & '\prog.ico', -1)
46 Else
47     GUISetIcon('shell32.dll', -13)
48 EndIf
49 $hTab = GUICtrlCreateTab(10, 10, $iWidth - 20, 475)
50 GUICtrlSetOnEvent(-1, '_TabClick')
51 GUICtrlSetResizing(-1, 2 + 4 + 32 + 64)
52 For $i = 0 To UBound($aTabNames) - 1
53     $aTabIds[$i] = GUICtrlCreateTabItem($aTabNames[$i])
54     $aContent[$i] = GUICtrlCreateListView(' | ', 22, 42, $iWidth - 45, 430, $LVS_NOCOLUMNHEADER, $LVS_EX_DOUBLEBUFFER + $WS_EX_CLIENTEDGE)
55     GUICtrlSetResizing(-1, 2 + 4 + 32 + 64)
56     GUICtrlSetBkColor(-1, 0xffffcc)
57 Next
58 GUICtrlCreateTabItem('')
59 $hStatus = _GUICtrlStatusBar_Create($hGUI)
60 GUIRegisterMsg($WM_SIZE, 'WM_SIZE')
61 GUIRegisterMsg($WM_GETMINMAXINFO, 'MY_WM_GETMINMAXINFO')
62 _CallFunction(0)
63 GUISetState()
64
65 While True
66     Sleep(1000)
67
68 >+16:34:27 AU3Check ended.rc:0
69 >Running: (3.3.16.1):C:\Users\LiveHacking\OneDrive\LiveHacking\Tools\AutoIt3\autoit3_x64.exe "C:\Users\LiveHacking\OneDrive\LiveHacking\Tools\AutoIt3\EigeneWerke\Computer-Info.au3"
70 >Setting Hotkeys...--> Press Ctrl+Alt+Break to Restart or Ctrl+BREAK to Stop.
71
72 It=76 con=1 INS (CR+LF)
```





Jupyter Notebook / Lab

The screenshot displays a Jupyter Notebook environment with a file explorer on the left and a code editor on the right. The file explorer shows a directory structure with files like 'qr-generator.ipynb' and 'qr-generator.ini'. The code editor contains three code cells with the following content:

```
[1]: code_text = "http://docs.tx7.de/TT-C11"
     file_name = "TT-C11"
```

Set Text and Filename

```
[2]: import configparser
     ini = configparser.ConfigParser()
     ini.read('qr-generator.ini', encoding='utf-8')

     color_scheme = "TOM0 blaugrau"
     fill_color = ini.get(color_scheme, "FillColor")
     back_color = ini.get(color_scheme, "BackColor")
```

Select Colorscheme

```
[3]: import qrcode
     ## pip install qrcode[pil] --user

     qr = qrcode.QRCode(border=1, box_size=40, error_correction=qrcode.constants.ERROR_CORRECT_M)
     qr.add_data(f"{code_text}")
     qr.make(fit=True)

     img = qr.make_image(fill_color=fill_color, back_color=back_color)
     img.save(f"QR-Codes/{file_name}.png")
```

Generate QR-Code

The status bar at the bottom indicates the notebook is running on Python 3 (ipykernel) and is in Command mode.

Anmerkungen oder Fragen?