

# QR-Code als Zwerg-QR-Code

Prof. Dr. Dörte Haftendorn Mai 2014 [www.mathematik-verstehen.de](http://www.mathematik-verstehen.de)

Erklärungen weitere Links und Impressum in der Webversion unten.

<http://haftendorn.uni-lueneburg.de/mathe-lehramt/codierung/qr/QR-Zwerg.htm>

<http://www.leuphana.de/masuv/03/codi/qr/QR-Zwerg.htm>

```
Manipulate[DynamicModule[{urz, Parity, hamming, qrmstart, qr, codew},
  [manipuliere [dynamisches Modul
    (* Definitionen der Funktionen *)
    urz[zahl_] := Partition[PadLeft[IntegerDigits[zahl, 2], 20, 0], 4];
      [partitioniere [fülle auf... [Ziffern einer ganzen Zahl
    Parity[uli_] := Mod[Plus@@uli, 2];
      [M... [addiere
    (* Der Kopf der Liste, also List, wird als Plus geändert *)
      [Liste [addiere
  hamming[u_] := Module[{uli = u, i, li},
    [Modul
    li = uli~Join~{Parity[{uli[[1]], uli[[2]], uli[[4]]}}];
      [verknüpfe
    li = li~Join~{Parity[{uli[[1]], uli[[3]], uli[[4]]}}];
      [verknüpfe
    li = li~Join~{Parity[{uli[[2]], uli[[3]], uli[[4]]}}];
      [verknüpfe
    li
  ];
  qrmstart = {{st, st, st, lg, st, lg, st},
    {st, lg, st, □, □, □, □}, {st, st, st, □, □, □, □}, {□, □, □, □, □, □, □},
    {□, □, □, □, □, □, □}, {□, □, □, □, □, □, □}, {□, □, □, □, □, □, st}};
  qr[cod_, qrmf_] := Module[{code = cod, qrm = qrmf, colorRegeln},
    [Modul
    (* Entsprechende Belegung*)
    colorRegeln = {st -> RGBColor[0, 0, 1], lg -> RGBColor[0.8, 0.8, 0.8],
      [RGB Farbe [RGB Farbe
    a1 -> RGBColor[1, 0, 1], a2 -> RGBColor[1, 0, 1],
      [RGB Farbe [RGB Farbe
    a3 -> RGBColor[1, 0, 1], a4 -> RGBColor[1, 0, 1],
      [RGB Farbe [RGB Farbe
    a5 -> RGBColor[1, 0, 1], a6 -> RGBColor[1, 0, 1], a7 -> RGBColor[1, 0, 1],
      [RGB Farbe [RGB Farbe [RGB Farbe
    b1 -> RGBColor[0, 1, 1], b2 -> RGBColor[0, 1, 1], b3 -> RGBColor[0, 1, 1], b4 -> RGBColor[0, 1, 1],
      [RGB Farbe [RGB Farbe [RGB Farbe [RGB Farbe
    b5 -> RGBColor[0, 1, 1], b6 -> RGBColor[0, 1, 1], b7 -> RGBColor[0, 1, 1],
      [RGB Farbe [RGB Farbe [RGB Farbe
    c1 -> RGBColor[1, 0, 0], c2 -> RGBColor[1, 0, 0], c3 -> RGBColor[1, 0, 0], c4 -> RGBColor[1, 0, 0],
      [RGB Farbe [RGB Farbe [RGB Farbe [RGB Farbe
    c5 -> RGBColor[1, 0, 0], c6 -> RGBColor[1, 0, 0], c7 -> RGBColor[1, 0, 0],
      [RGB Farbe [RGB Farbe [RGB Farbe
    d1 -> RGBColor[0, 1, 0], d2 -> RGBColor[0, 1, 0], d3 -> RGBColor[0, 1, 0], d4 -> RGBColor[0, 1, 0],
      [RGB Farbe [RGB Farbe [RGB Farbe [RGB Farbe
    d5 -> RGBColor[0, 1, 0], d6 -> RGBColor[0, 1, 0], d7 -> RGBColor[0, 1, 0],
      [RGB Farbe [RGB Farbe [RGB Farbe
```

```

e1 -> RGBColor[1, 1, 0], e2 -> RGBColor[1, 1, 0], e3 -> RGBColor[1, 1, 0], e4 -> RGBColor[1, 1, 0],
  RGB Farbe      RGB Farbe      RGB Farbe      RGB Farbe
e5 -> RGBColor[1, 1, 0], e6 -> RGBColor[1, 1, 0], e7 -> RGBColor[1, 1, 0];
  RGB Farbe      RGB Farbe      RGB Farbe

{qrm[[2, 4]], qrm[[2, 5]], qrm[[2, 6]], qrm[[2, 7]],
  qrm[[3, 4]], qrm[[3, 5]], qrm[[3, 6]], qrm[[3, 7]]} =
{a1 code[[1, 1]], b1 code[[2, 1]], c1 code[[3, 1]],
  d1 code[[4, 1]], e1 code[[5, 1]], a2 code[[1, 2]], b2 code[[2, 2]], c2 code[[3, 2]]};
{qrm[[4, 1]], qrm[[4, 2]], qrm[[4, 3]], qrm[[4, 4]], qrm[[4, 5]], qrm[[4, 6]], qrm[[4, 7]]} =
{d2 code[[4, 2]], e2 code[[5, 2]],
  a3 code[[1, 3]], b3 code[[2, 3]], c3 code[[3, 3]], d3 code[[4, 3]], e3 code[[5, 3]]};
{qrm[[5, 1]], qrm[[5, 2]], qrm[[5, 3]], qrm[[5, 4]], qrm[[5, 5]], qrm[[5, 6]], qrm[[5, 7]]} =
{a4 code[[1, 4]], b4 code[[2, 4]],
  c4 code[[3, 4]], d4 code[[4, 4]], e4 code[[5, 4]], a5 code[[1, 5]], b5 code[[2, 5]]};
{qrm[[6, 1]], qrm[[6, 2]], qrm[[6, 3]], qrm[[6, 4]], qrm[[6, 5]], qrm[[6, 6]], qrm[[6, 7]]} =
{c5 code[[3, 5]], d5 code[[4, 5]],
  e5 code[[5, 5]], a6 code[[1, 6]], b6 code[[2, 6]], c6 code[[3, 6]], d6 code[[4, 6]]};
{qrm[[7, 1]], qrm[[7, 2]], qrm[[7, 3]], qrm[[7, 4]], qrm[[7, 5]], qrm[[7, 6]], qrm[[7, 7]]} =
{e6 code[[5, 6]], a7 code[[1, 7]], b7 code[[2, 7]], c7 code[[3, 7]], d7 code[[4, 7]], e7 code[[5, 7]], st};
(* qrm//MatrixForm;*)
  Matrizenform

ArrayPlot[qrm, ColorRules -> colorRegeln];
  stelle Array dar  Farbgeregeln

(* eigentliche Manipulate-Befehle im Modul *)
  manipuliere

codew = hamming /@ urz[zahl];
qr[codew, qrmstart]
] (* Module-ende *)
  Modul

,
(* ***** Mein Standardblock ***** *)
Style["Trage eine sechsstellige Zahl in dem Kasten ein: ", 12],
  Stil
{{zahl, 250348, Zahl}, 1, 1000000, 1, Appearance -> {"Open"}},
  Darstellung  öffne
Style["Das ist der Zwerg-QR-Code deiner Zahl: ", 14, Bold],
  fett
(*Style["(Angabe ist jeder Primfaktor und sein Exponent) ", 12], *)
  Stil  Exponent

FrameLabel -> {{None, "Info\nunten"},
  Rahmenbeschriftung  keine
  {"www.mathematik-verstehen.de", "Zwerg-QR-Code erstellen"}},
LabelStyle -> Directive[RGBColor[0.7, 0, 0], Large]
  Anweisung  RGB Farbe  groß
]

```

## Zwerg-QR-Code erstellen

Trage eine sechsstellige Zahl in dem Kasten ein:

Zahl

Das ist der Zwerg-QR-Code deiner Zahl:

Info unten

[www.mathematik-verstehen.de](http://www.mathematik-verstehen.de)