

Zwerg-QR-Code lesen

Prof. Dr. Dörte Haftendorn Juli 2014 [www.mahematik – verstehen.de](http://www.mahematik-verstehen.de)

Erklärungen weitere Links und Impressum in der Webversion unten.

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

Manipulate [(*

[manipuliere](#)

DynamicModule [{colorRegeln,codeLesen,Parity,hamming,hammingKorr,

[dynamisches Modul](#)

reparieren, hexraus,codeheil}, *)

(* Definitionen der Funktionen*)

colorRegeln =

{st -> RGBColor[0, 0, 1], lg -> LightGray, a1 -> RGBColor[1, 0, 1], a2 -> RGBColor[1, 0, 1], a3 -> RGBColor[1, 0, 1],
[\[RGB Farbe\]](#) [\[hellgrau\]](#) [\[RGB Farbe\]](#) [\[RGB Farbe\]](#) [\[RGB Farbe\]](#)

a4 -> RGBColor[1, 0, 1], a5 -> RGBColor[1, 0, 1], a6 -> RGBColor[1, 0, 1], a7 -> RGBColor[1, 0, 1],
[\[RGB Farbe\]](#) [\[RGB Farbe\]](#) [\[RGB Farbe\]](#) [\[RGB Farbe\]](#)

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];
[\[RGB Farbe\]](#) [\[RGB Farbe\]](#) [\[RGB Farbe\]](#)

codeLesen[qrles_] := Module[{code, qrm = qrles}, (* Entsprechende Belegung*)

[\[Modul\]](#)

code =
$$\begin{pmatrix} \square & \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square & \square \end{pmatrix};$$

{code[[1, 1]], code[[2, 1]], code[[3, 1]], code[[4, 1]], code[[5, 1]], code[[1, 2]], code[[2, 2]], code[[3, 2]]} =

{qrm[[2, 4]], qrm[[2, 5]], qrm[[2, 6]], qrm[[2, 7]],

qrm[[3, 4]], qrm[[3, 5]], qrm[[3, 6]], qrm[[3, 7]]};

{code[[4, 2]], code[[5, 2]], code[[1, 3]], code[[2, 3]], code[[3, 3]], code[[4, 3]], code[[5, 3]]} =

{qrm[[4, 1]], qrm[[4, 2]], qrm[[4, 3]], qrm[[4, 4]], qrm[[4, 5]], qrm[[4, 6]], qrm[[4, 7]]};

{code[[1, 4]], code[[2, 4]], code[[3, 4]], code[[4, 4]], code[[5, 4]], code[[1, 5]], code[[2, 5]]} =

{qrm[[5, 1]], qrm[[5, 2]], qrm[[5, 3]], qrm[[5, 4]], qrm[[5, 5]], qrm[[5, 6]], qrm[[5, 7]]};

{code[[3, 5]], code[[4, 5]], code[[5, 5]], code[[1, 6]], code[[2, 6]], code[[3, 6]], code[[4, 6]]} =

{qrm[[6, 1]], qrm[[6, 2]], qrm[[6, 3]], qrm[[6, 4]], qrm[[6, 5]], qrm[[6, 6]], qrm[[6, 7]]};

{code[[5, 6]], code[[1, 7]], code[[2, 7]], code[[3, 7]], code[[4, 7]], code[[5, 7]], st} =

{qrm[[7, 1]], qrm[[7, 2]], qrm[[7, 3]], qrm[[7, 4]], qrm[[7, 5]], qrm[[7, 6]], qrm[[7, 7]]};

code

];

```

Parity[uli_] := Mod[Plus@@uli, 2] (* Der Kopf der Liste, also List, wird als Plus geändert *);
    ... |addiere |Liste |addiere
hamming[u_] := Module[{uli = u, i, li},
    |Modul
    li = uli ~ Join ~ {Parity[{uli[[1]], uli[[2]], uli[[4]]]}};
    |verknüpfte
    li = li ~ Join ~ {Parity[{uli[[1]], uli[[3]], uli[[4]]]}};
    |verknüpfte
    li = li ~ Join ~ {Parity[{uli[[2]], uli[[3]], uli[[4]]]}};
    |verknüpfte
    li
];
hammingKorr[codeWortFalsch_] := Module[{cf = codeWortFalsch, h, ok},
    |Modul
    h = hamming[{cf[[1]], cf[[2]], cf[[3]], cf[[4]]]};
    ok = cf;
    If[cf[[5]] == h[[5]] && cf[[6]] == h[[6]] && cf[[7]] == h[[7]], ok = cf];
    |wenn
    If[cf[[5]] != h[[5]] && cf[[6]] != h[[6]] && cf[[7]] == h[[7]], ok[[1]] = Mod[cf[[1]] + 1, 2];
    |Modulo
    If[cf[[5]] != h[[5]] && cf[[6]] == h[[6]] && cf[[7]] != h[[7]], ok[[2]] = Mod[cf[[2]] + 1, 2];
    |wenn |Modulo
    If[cf[[5]] == h[[5]] && cf[[6]] != h[[6]] && cf[[7]] != h[[7]], ok[[3]] = Mod[cf[[3]] + 1, 2];
    |Modulo
    If[cf[[5]] != h[[5]] && cf[[6]] != h[[6]] && cf[[7]] != h[[7]], ok[[4]] = Mod[cf[[4]] + 1, 2];
    |Modulo
    If[cf[[5]] != h[[5]] && cf[[6]] == h[[6]] && cf[[7]] == h[[7]], ok[[5]] = Mod[cf[[5]] + 1, 2];
    |Modulo
    If[cf[[5]] == h[[5]] && cf[[6]] != h[[6]] && cf[[7]] == h[[7]], ok[[6]] = Mod[cf[[6]] + 1, 2];
    |Modulo
    If[cf[[5]] == h[[5]] && cf[[6]] == h[[6]] && cf[[7]] != h[[7]], ok[[7]] = Mod[cf[[7]] + 1, 2];
    |Modulo
    ok
];
repariere[codeFalsch_] := hammingKorr /@ codeFalsch;
hexraus[codeheil_] := Module[{ch = codeheil, urr, i, j},
    |Modul
    urr = 
$$\begin{pmatrix} \square & \square & \square & \square \\ \square & \square & \square & \square \\ \square & \square & \square & \square \\ \square & \square & \square & \square \\ \square & \square & \square & \square \end{pmatrix};$$

    For[i = 1, i < 6, i++,
    |For-Schleife
    For[j = 1, j < 5, j++,
    |For-Schleife
    urr[[i, j]] = ch[[i, j]]];
qrmstart = {{(st, st, st, lg, st, lg, st), (st, lg, st, □, □, □), (st, st, st, □, □, □),
    {□, □, □, □, □, □}, {□, □, □, □, □, □}, {□, □, □, □, □, □}, {□, □, □, □, □, st}}};
qr[cod_, qrmf_] := DynamicModule[{code = cod, qrm = qrmf, colorRegeln},
    |dynamisches 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; *)
  Matritzenform
ArrayPlot[qrm, ColorRules -> colorRegeln];
  stelle Array dar  Farbregelein

```

(* Begin des eigentlichen Manipulate-Codes *)

```

  beginne Kontext      manipuliere
codeheil = repariere[codeLesen[daten]];
Grid[{{ArrayPlot[daten, ColorRules -> {st -> Blue, lg -> LightGray}},
  Gitter  stelle Array dar  Farbregelein  blau  hellgrau
  qr[codeheil, qrmstart]},
{Graphics[Text[Style["Pos.1.1 Dein Zwerg-QR-Code\nPos.2.1 Deine 5 Codewörter"
  Graphik  Text  Stil
  zu 7 Stellen\nPos.3.1 Hamming-korrigierte Codewörter", 10]]
(* ,ContentSelectable->True *) ], Graphics[Text[Style[
  auswählbarer Inhalt  wahr  Graphik  Text  Stil
  "Pos.1.2 Dein Zwerg-QR-Code wie 1.1\nPos.1.3 Korrigierter
  Zwerg-QR-Code\n\nPos.3.2 Die korrigierte
  Nachricht:\n 5 Binärwörter,\n5 Hexadezimalwörter,\n

```

```

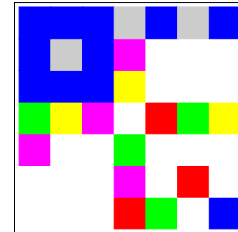
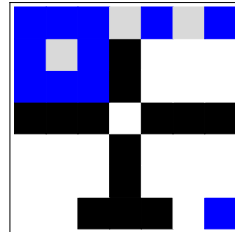
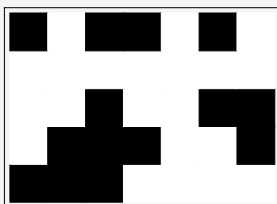
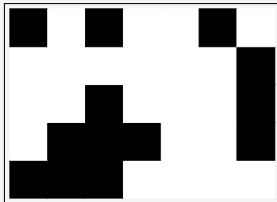
    eine Dezimalzahl als Nachricht", 10]]
  (* ,ContentSelectable→True] *) ]], {ExpressionCell[
    |auswählbarer Inhalt |wahr |Ausdruckszelle
  {MatrixForm[hexraus[codeheil]],
    |Matritzenform
  Transpose[{FromDigits[#, 2] & /@hexraus[codeheil]}] // MatrixForm,
    |transponiere |Zahl aus Ziffern |Matritzenform
  FromDigits[Flatten[hexraus[codeheil]], 2]}]}]]
  |Zahl aus Ziff... |ebene ein
,
Style["Klicke deinen Zwerg-QR-Code an", 12],
|Stil
  { {daten, 
$$\begin{pmatrix} \text{st} & \text{st} & \text{st} & \text{lg} & \text{st} & \text{lg} & \text{st} \\ \text{st} & \text{lg} & \text{st} & 0 & 0 & 0 & 0 \\ \text{st} & \text{st} & \text{st} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \text{st} \end{pmatrix}$$
 }, ControlType → None},
    |Art des Bediene... |keine
Dynamic[Panel[
  |dynamisch |graphisches Gestaltungsfeld
  Grid[Outer[Checkbox[Dynamic[daten[ [#1, #2]]], {0, 1}] &, Range[7], Range[7]]]],
    |Gitter |äuße... |Kontrollkä... |dynamisch |Liste aufei... |Liste aufeinanderfolg
  Dynamic[GraphicsGrid[{{ ArrayPlot[codeLesen[daten]]}, {ArrayPlot[codeheil]
    |dynamisch |Graphik in Gitteran... |stelle Array dar |stelle Array dar
  }]}], ControlPlacement → Left,
    |Platzierung des Bedien... |links
  FrameLabel →
    |Rahmenbeschriftung
    {{None, "Info\nunten"}, {"www.mathematik-verstehen.de", "Zwerg-QR-Code-lesen"}},
    |keine
  LabelStyle → Directive[RGBColor[0.7, 0, 0], Large]
    |Beschriftungsstil |Anweisung |RGB Farbe |groß
] (* Ende Manipulate*)
  |manipuliere

```

Zwerg-QR-Code-lesen

Klicke deinen Zwerg-QR-Code an

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>



1.1 Dein Zwerg-QR-Code
 1.2 Dein Zwerg-QR-Code w
 1.3 Korrigierter Zwerg-QR-
 1.4 Deine 5 Codewörter zu 7 s
 1.5 Hamming-korrigierte Code
 1.6 Die korrigierte Nachricht
 1.7 5 Binärwörter,
 1.8 5 Hexadezimalwörter,
 1.9 ihre Dezimalzahl als Nachric

$$\begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix},$$

$$\begin{pmatrix} 11 \\ 0 \\ 2 \\ 7 \\ 14 \end{pmatrix}, \{721\ 534\}$$

Info
unten

www.mathematik-verstehen.de

Bilder

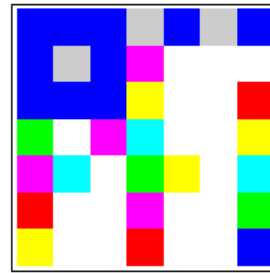
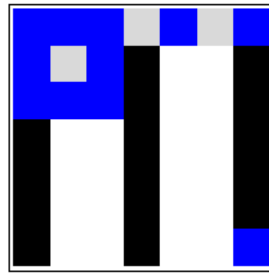
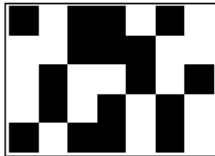
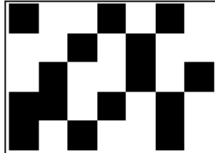
Quit

[beende Kernel](#)

Das Folgende sind nur zwei Bilder:

Zwerg-QR-Code-lesen

Klicke deinen Zwerg-QR-Code an



Pos.1.1 Dein Zwerg-QR-Code
 Pos.2.1 Deine 5 Codewörter zu 7 Stellen
 Pos.3.1 Hamming-korrigierte Codewörter

Pos.1.2 Dein Zwerg-QR-Code wie 1.1
 Pos.1.3 Korrigierter Zwerg-QR-Code

Pos.3.2 Die korrigierte Nachricht:
 5 Binärwörter,
 5 Hexadezimalwörter,
 eine Dezimalzahl als Nachricht

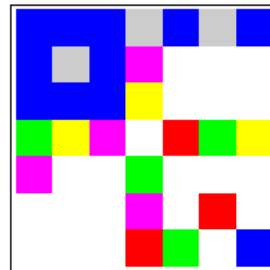
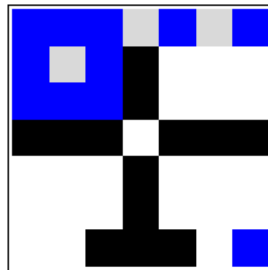
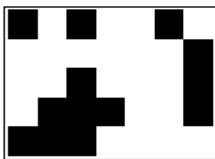
$$\left\{ \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}, \begin{pmatrix} 11 \\ 3 \\ 4 \\ 5 \\ 11 \end{pmatrix}, 734299 \right\}$$

Info unten

www.mathematik-verstehen.de

Zwerg-QR-Code-lesen

Klicke deinen Zwerg-QR-Code an



Pos.1.1 Dein Zwerg-QR-Code
 Pos.2.1 Deine 5 Codewörter zu 7 Stellen
 Pos.3.1 Hamming-korrigierte Codewörter

Pos.1.2 Dein Zwerg-QR-Code wie 1.1
 Pos.1.3 Korrigierter Zwerg-QR-Code

Pos.3.2 Die korrigierte Nachricht:
 5 Binärwörter,
 5 Hexadezimalwörter,
 eine Dezimalzahl als Nachricht

$$\left\{ \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 11 \\ 0 \\ 2 \\ 7 \\ 14 \end{pmatrix}, 721534 \right\}$$

Info unten

www.mathematik-verstehen.de