

Binärdarstellung der RGB-Werte

Bislang hast du die RGB-Werte als Zahlen zwischen 0 und 255 dargestellt. Für die Speicherung und Verarbeitung im Rechner müssen auch diese Werte in einen binären Code aus Nullen und Einsen umgewandelt werden. Für die einzelnen Farbanteile stehen dabei jeweils 1 Byte, also 8 Bit zur Verfügung. Die Zahlen zwischen 0 und 255 werden dabei in das binäre Zahlensystem umgerechnet.

Vielleicht kennst du Binärzahlen aus dem Mathematikunterricht oder hast dich bei der Untersuchung des ASCII-Codes bereits damit beschäftigt. Eine Erläuterung findest du ansonsten in der Datei „Exkurs_Binaersystem“.

Aufgabe 1:

- Um welchen Farbton handelt es sich bei dem binären RGB-Wert:
1100 1000 1100 1000 1100 1000
- Gib die binäre Codierung für ein helles und ein dunkles Magenta an.
- Ergänze in Tabelle 2 jeweils die fehlenden Angaben.





R(otanteil)		G(rünanteil)		B(lauanteil)		Farbe	
Dezimal	Binär	Dezimal	Binär	Dezimal	Binär	Name	Aussehen
						Rot	
	0000 0000		1011 1001		1011 1001		
112			0011 0000	160		Lila	
	11111010	178			0110 0000	Orange	

Tabelle 1: Dezimale und binäre RGB-Werte verschiedener Farben

Aufgabe 2: Begründe mithilfe der Binärdarstellung der RGB-Werte, warum für jeden Farbanteil genau die Zahlen zwischen 0 und 255 verwendet werden können. Beachte dabei, dass für jeden Farbanteil 1 Byte zur Verfügung steht.

Aufgabe 3: Wie viele Farben könnten wir unterscheiden, wenn für wir jeden Farbanteil nur ein Bit zur Verfügung hätten? Ordne jeder Farbe die entsprechende Codierung zu.

Beispiel: 1 0 0 codiert die Farbe Rot.

Dieses Werk und die beiliegenden Quelltexte sind lizenziert unter einer [Creative Commons Namensnennung - Nicht-kommerziell - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz](https://creativecommons.org/licenses/by-nc-sa/4.0/). Von der Lizenz ausgenommen ist das InfSI-Logo.