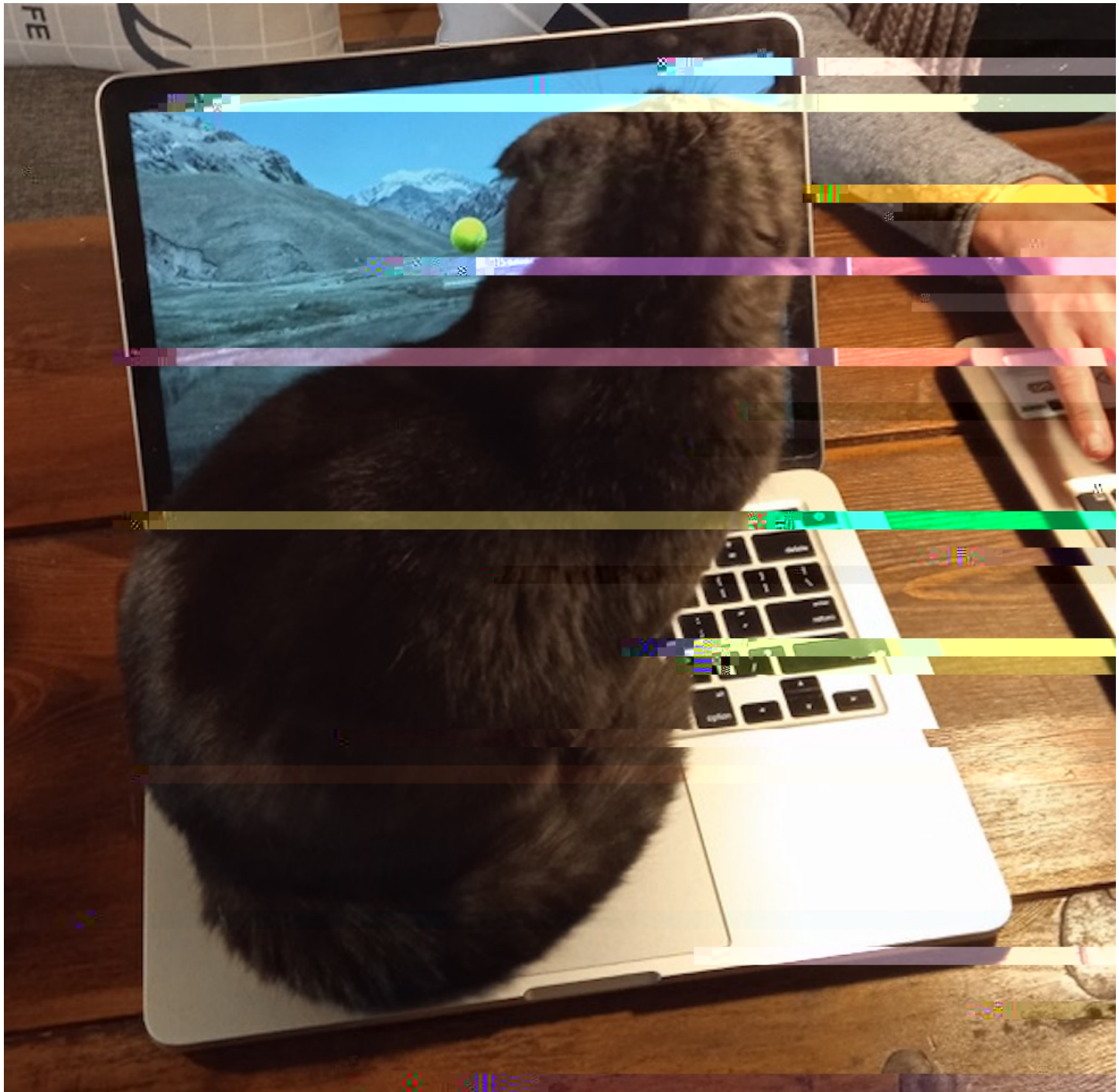
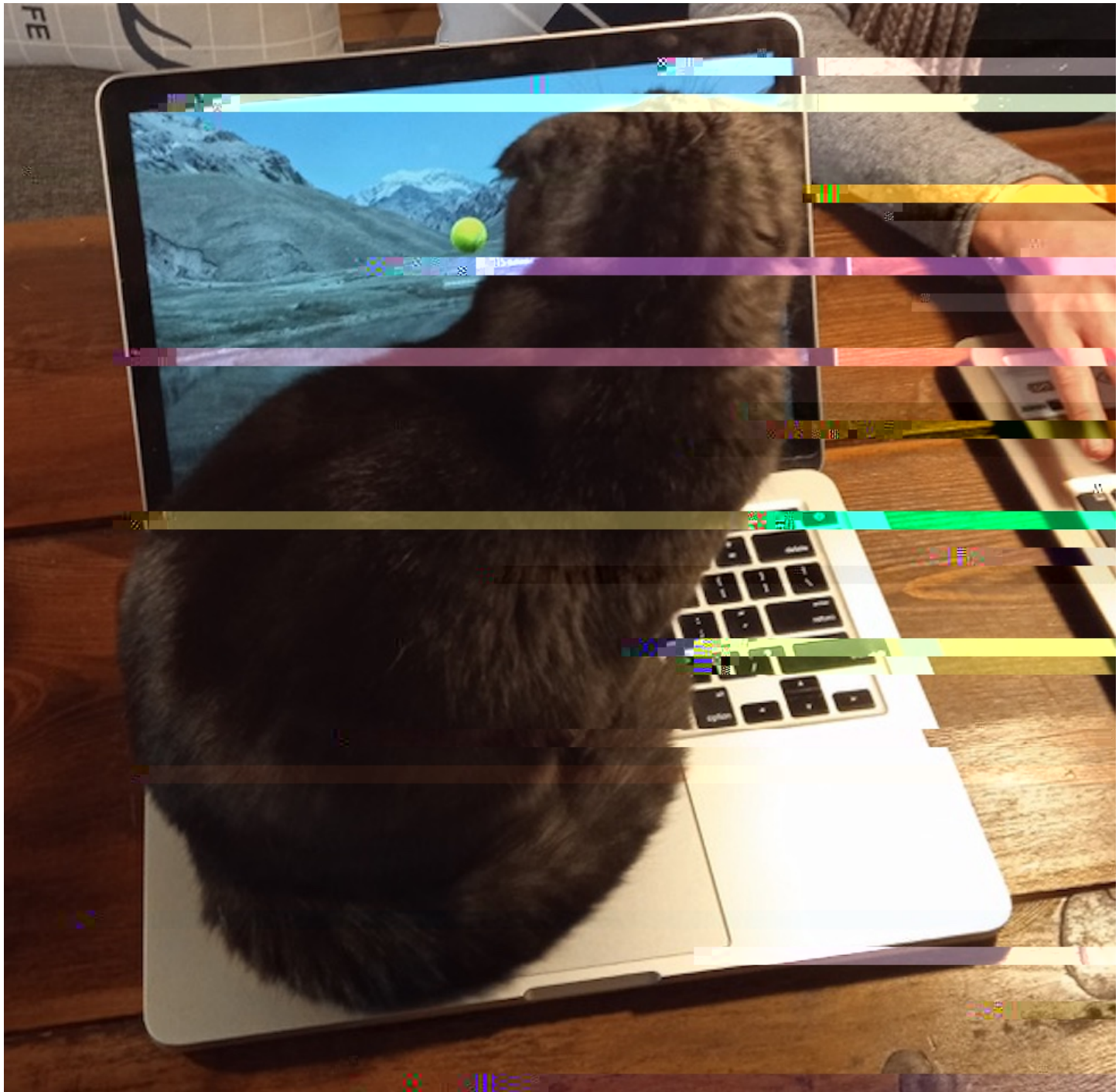
Error Correcting Codes

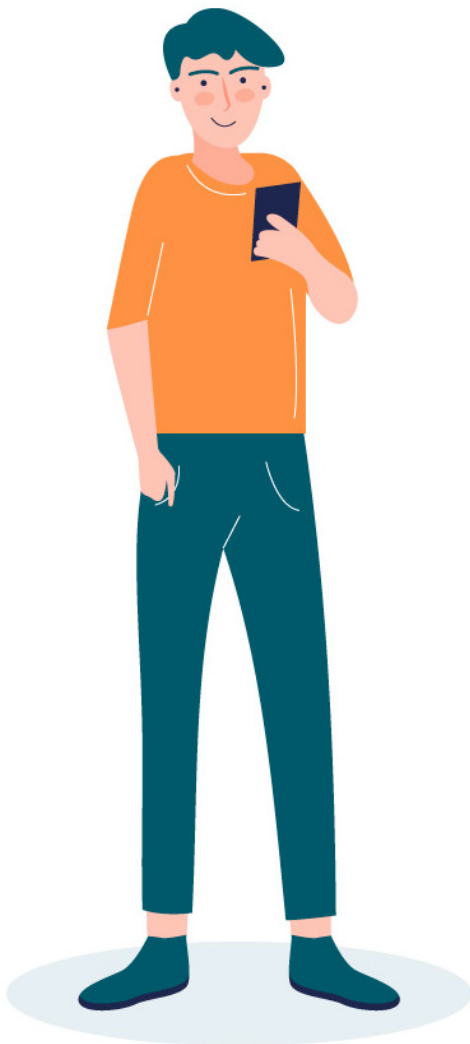


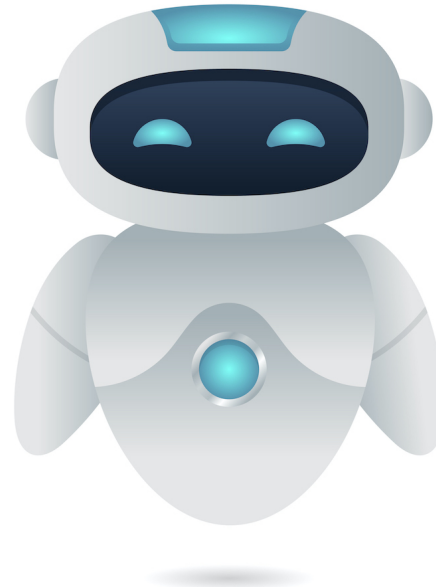


Empfänger	Osterhase
IBAN	DE78 5001 0517 8395 5917 76
Kreditinstitut	ING-DIBA
Betrag in EUR	2.000,00
Verwendungs-zweck	Unkosten fuer Eier und Schokolade 107

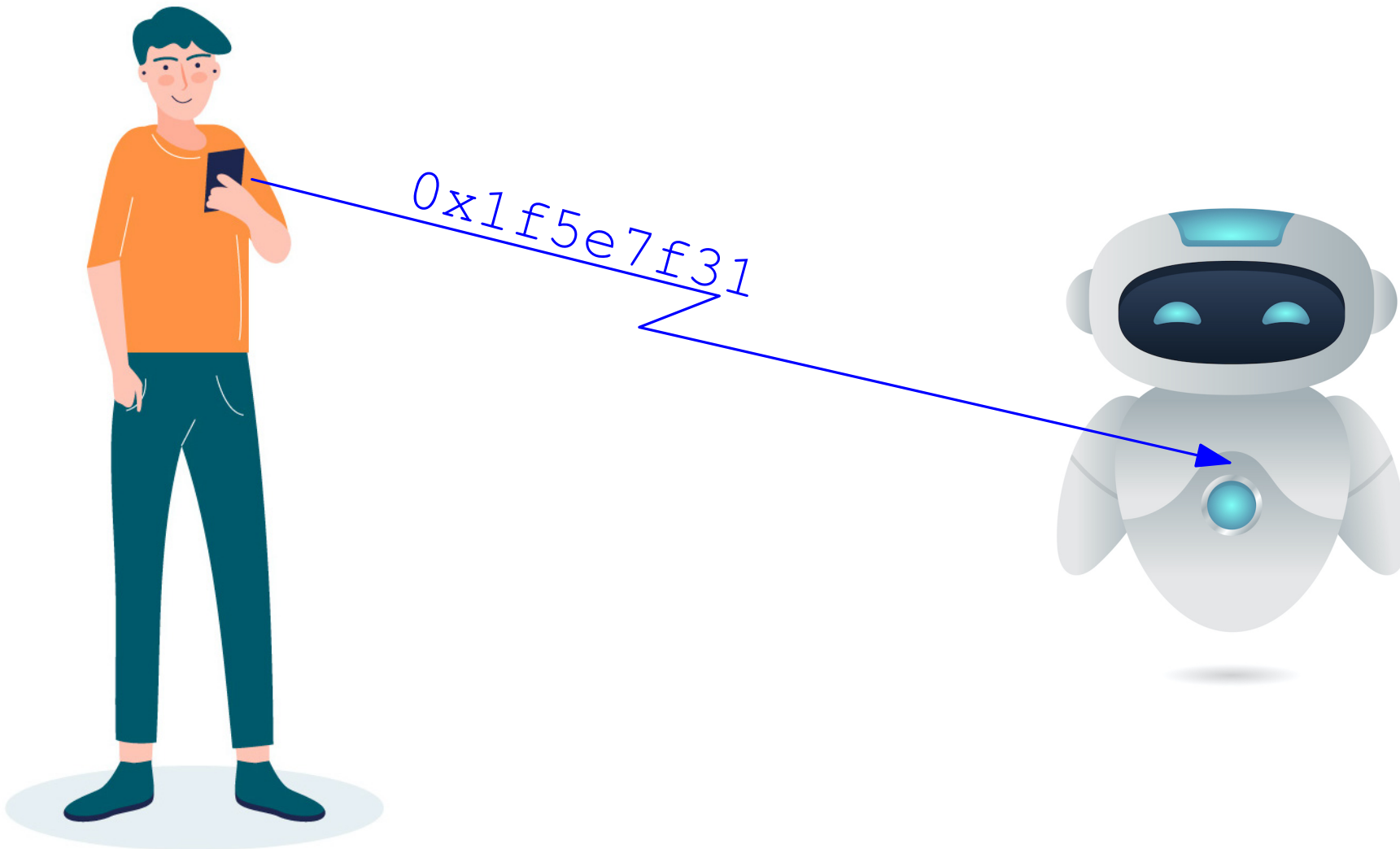
Empfänger	Osterhase
IBAN	DE78 5001 0514 8395 5917 76
Kreditinstitut	ING-DIBA
Betrag in EUR	2.000,00
Verwendungs-zweck	Unkosten fuer Eier und Schokolade 107

Empfänger	Osterhase
IBAN	DE78 5001 0514 8395 5917 76
	Bitte geben Sie eine gültige IBAN ein.
Kreditinstitut	ING-DIBA
Betrag in EUR	2.000,00
Verwendungs-zweck	Unkosten fuer Eier und Schokolade 107

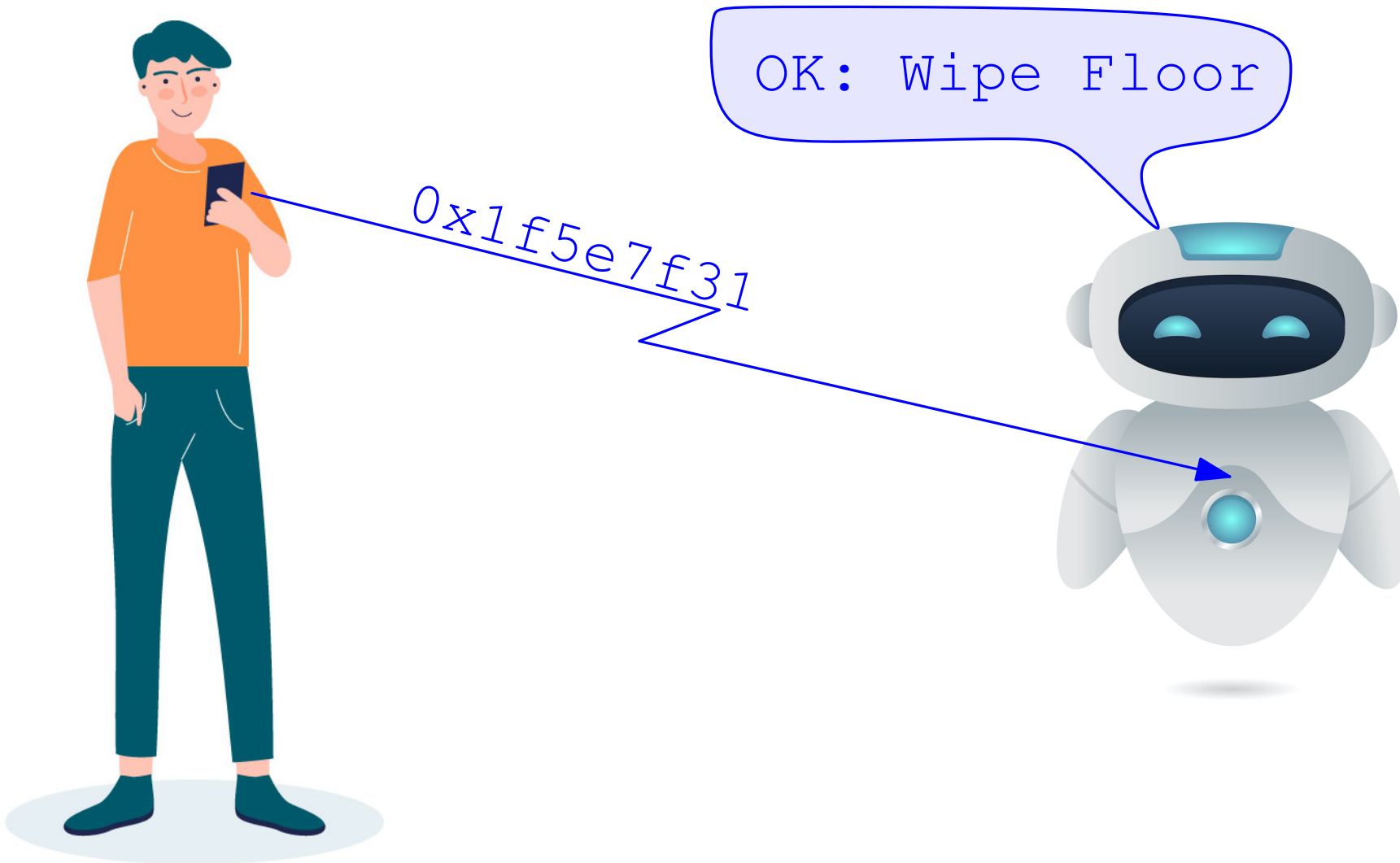




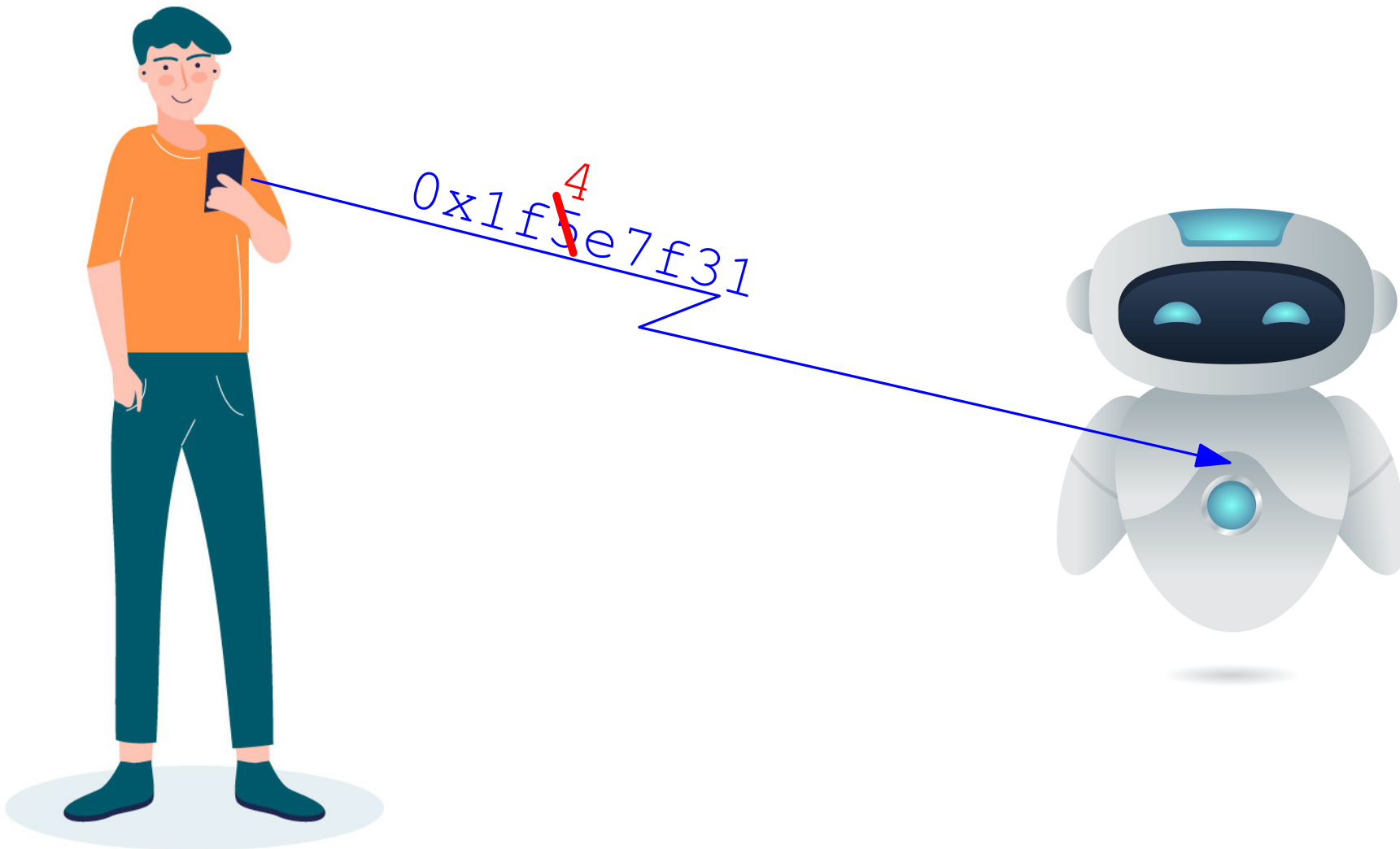
Bilder von `http://www.freepik.com`



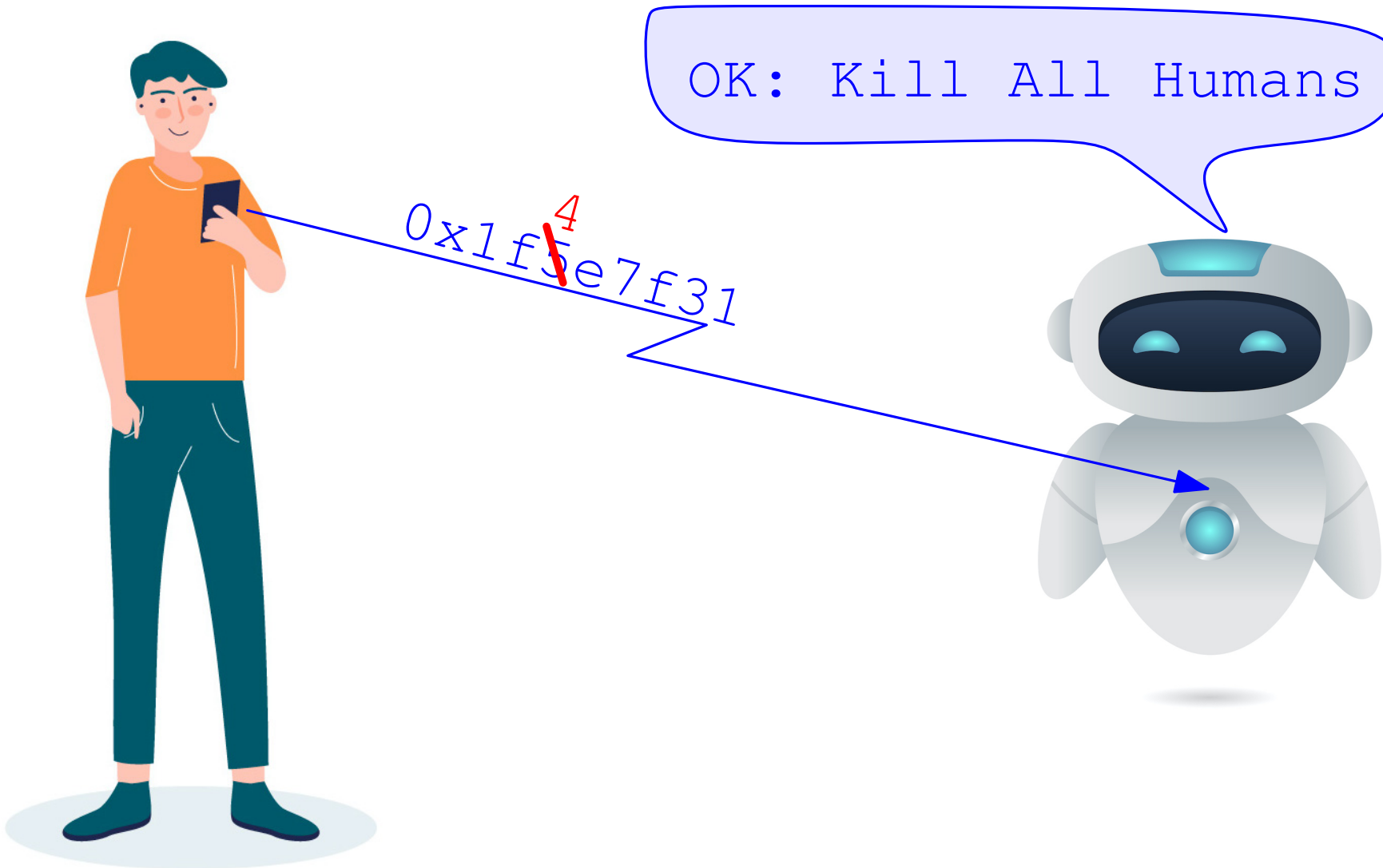
Bilder von `http://www.freepik.com`



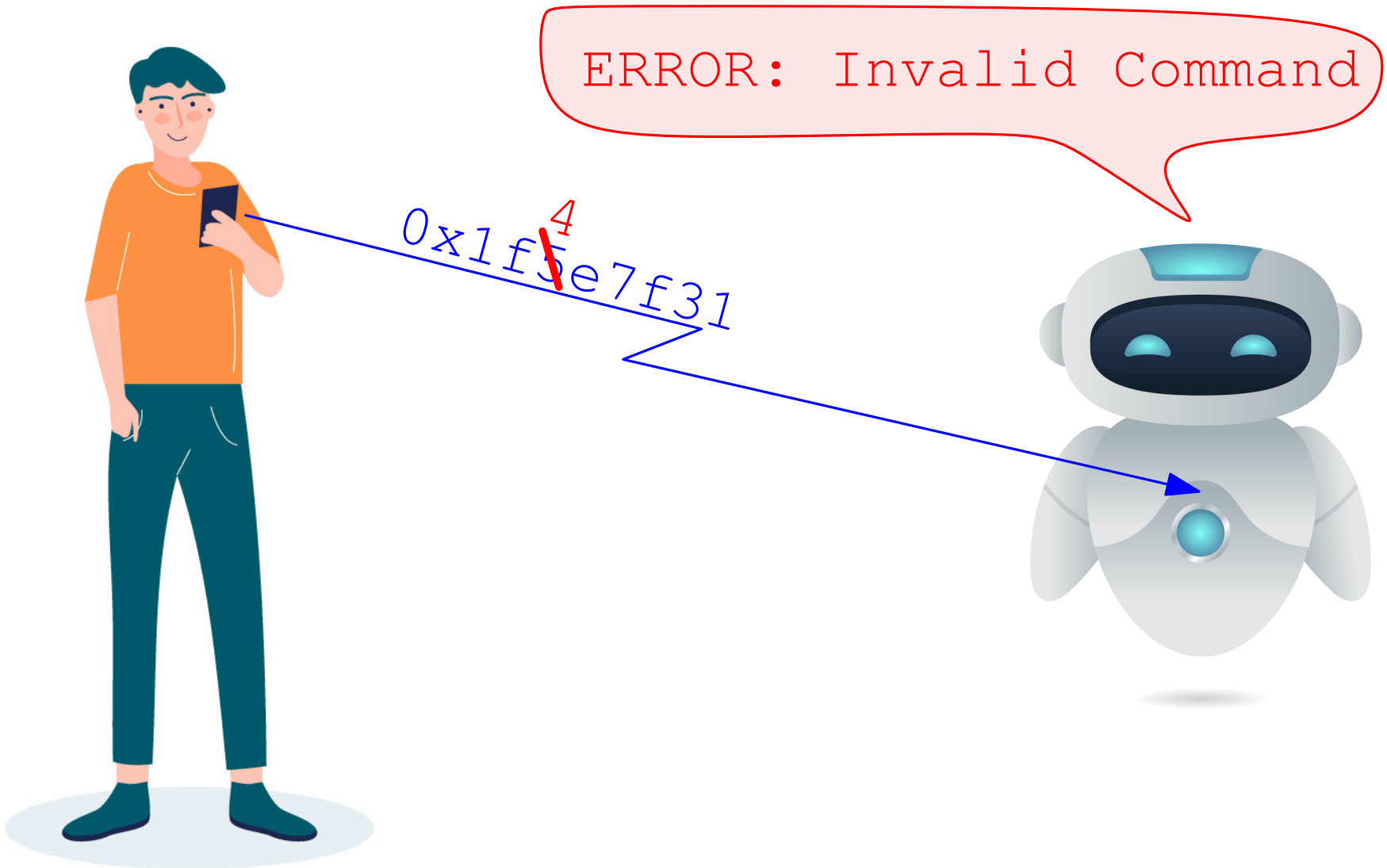
Bilder von `http://www.freepik.com`



Bilder von `http://www.freepik.com`



Bilder von `http://www.freepik.com`



Bilder von `http://www.freepik.com`

Error Correcting Codes

- Wir können gültige Code-Wörter von ungültigen unterscheiden.
- Zwei gültige Codewörter unterscheiden sich in vielen Stellen.

Definition

Definition. Sei Σ ein endliches Alphabet und Σ^n die Menge aller Wörter der Länge n über Σ. Der *Hamming*-Abstand zweier Wörter $\mathbf{x}, \mathbf{y} \in \Sigma^n$ ist

$$d_H(\mathbf{x}, \mathbf{y}) := |\{i \in [n] \mid x_i \neq y_i\}|$$

Ein *Code* ist eine Teilmenge $C \subseteq \Sigma^n$.

- Die *Länge* von C ist n.
- Der *Minimalabstand* (*minimum distance*) von C ist $\Delta(C) := \min\{d_H(\mathbf{x}, \mathbf{y}) \mid \mathbf{x}, \mathbf{y} \in C, \mathbf{x} \neq \mathbf{y}\}$
- Die *Rate* von C ist $R(C) := \frac{\log |C|}{n \log |\Sigma|}$.

Definition

Definition. Sei Σ ein endliches Alphabet und Σ^n die Menge aller Wörter der Länge n über Σ. Der *Hamming*-Abstand zweier Wörter $\mathbf{x}, \mathbf{y} \in \Sigma^n$ ist

$$d_H(\mathbf{x}, \mathbf{y}) := |\{i \in [n] \mid x_i \neq y_i\}|$$

Ein *Code* ist eine Teilmenge $C \subseteq \Sigma^n$.

- Die *Länge* von C ist n.
- Der *Minimalabstand* (*minimum distance*) von C ist $\Delta(C) := \min\{d_H(\mathbf{x}, \mathbf{y}) \mid \mathbf{x}, \mathbf{y} \in C, \mathbf{x} \neq \mathbf{y}\}$
- Die *Rate* von C ist $R(C) := \frac{\log |C|}{n \log |\Sigma|}$.

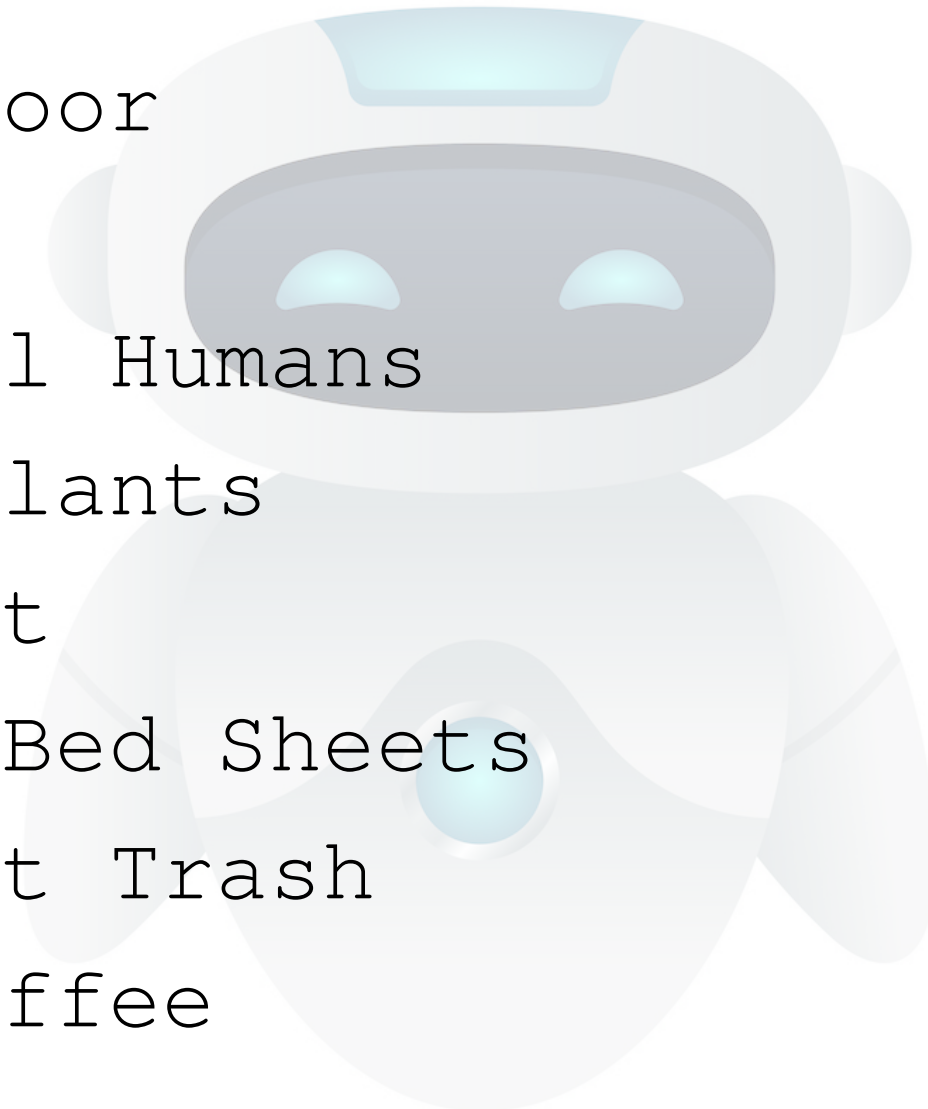
Beginnen Sie mit einfachen Beispielen!

Beginnen Sie mit einfachen Beispielen!

konkret  abstrakt

Roboter-Instruktionen

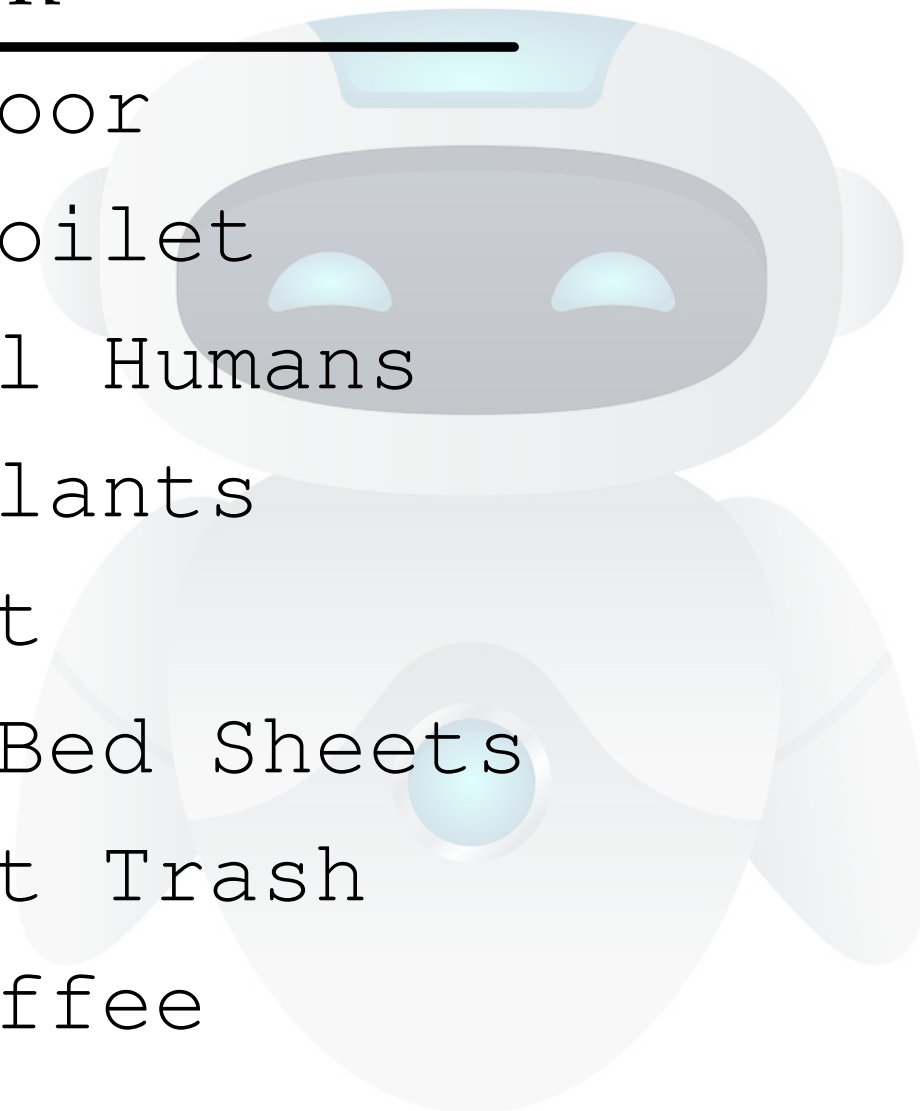
```
Wipe Floor

Kill All Humans
Water Plants
Feed Cat
Change Bed Sheets
Take Out Trash
Make Coffee
```



Bilder von http://www.freepik.com

Roboter-Instruktionen

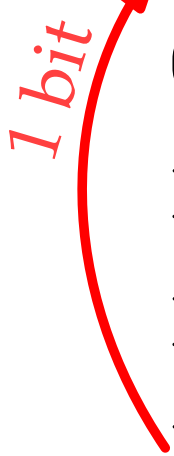
Binärcode	Instruktion
000	Wipe Floor
001	Clean Toilet
010	Kill All Humans
011	Water Plants
100	Feed Cat
101	Change Bed Sheets
110	Take Out Trash
111	Make Coffee



Bilder von http://www.freepik.com

Roboter-Instruktionen

Binärcode	Instruktion
000	Wipe Floor
001	Clean Toilet
010	Kill All Humans
011	Water Plants
100	Feed Cat
101	Change Bed Sheets
110	Take Out Trash
111	Make Coffee



Bilder von http://www.freepik.com

Checksumme

Checksumme. Wir investieren ein zusätzliches Bit und setzen dieses so, dass das Codewort insgesamt eine *gerade* Anzahl von 1en enthält.

Checksumme

Checksumme. Wir investieren ein zusätzliches Bit und setzen dieses so, dass das Codewort insgesamt eine *gerade* Anzahl von 1en enthält.

Das "rohe Wort" $x_1 x_2 \ldots x_n$ wird zum Codewort $x_1 x_2 \ldots x_n x_{n+1}$ mit

$$x_{n+1} := x_1 \oplus x_2 \oplus \cdots \oplus x_n$$

Checksumme

Checksumme. Wir investieren ein zusätzliches Bit und setzen dieses so, dass das Codewort insgesamt eine *gerade* Anzahl von 1en enthält.

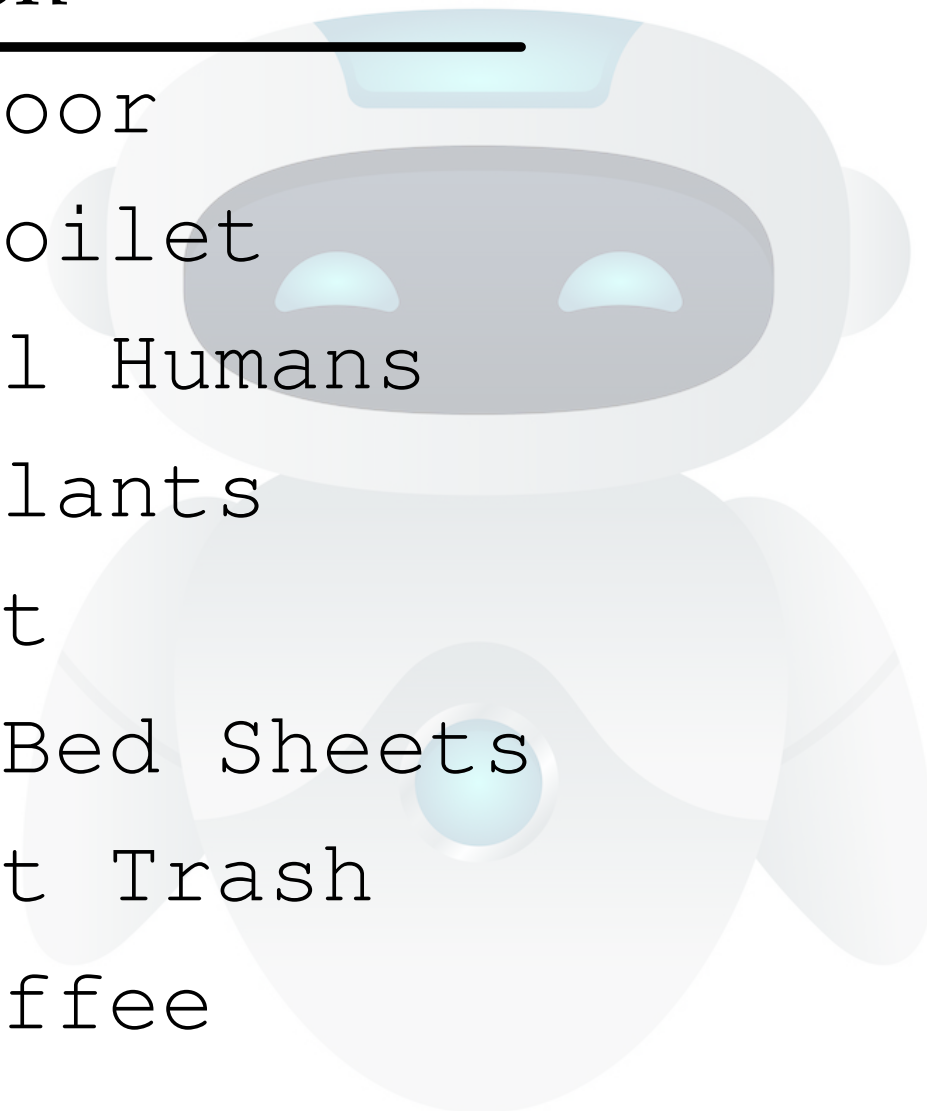
Das "rohe Wort" $x_1 x_2 \ldots x_n$ wird zum Codewort $x_1 x_2 \ldots x_n x_{n+1}$ mit

$$x_{n+1} := x_1 \oplus x_2 \oplus \cdots \oplus x_n$$

Gültigkeitsprüfung. $x_1 x_2 \ldots x_n x_{n+1}$ ist ein Codewort genau dann, wenn

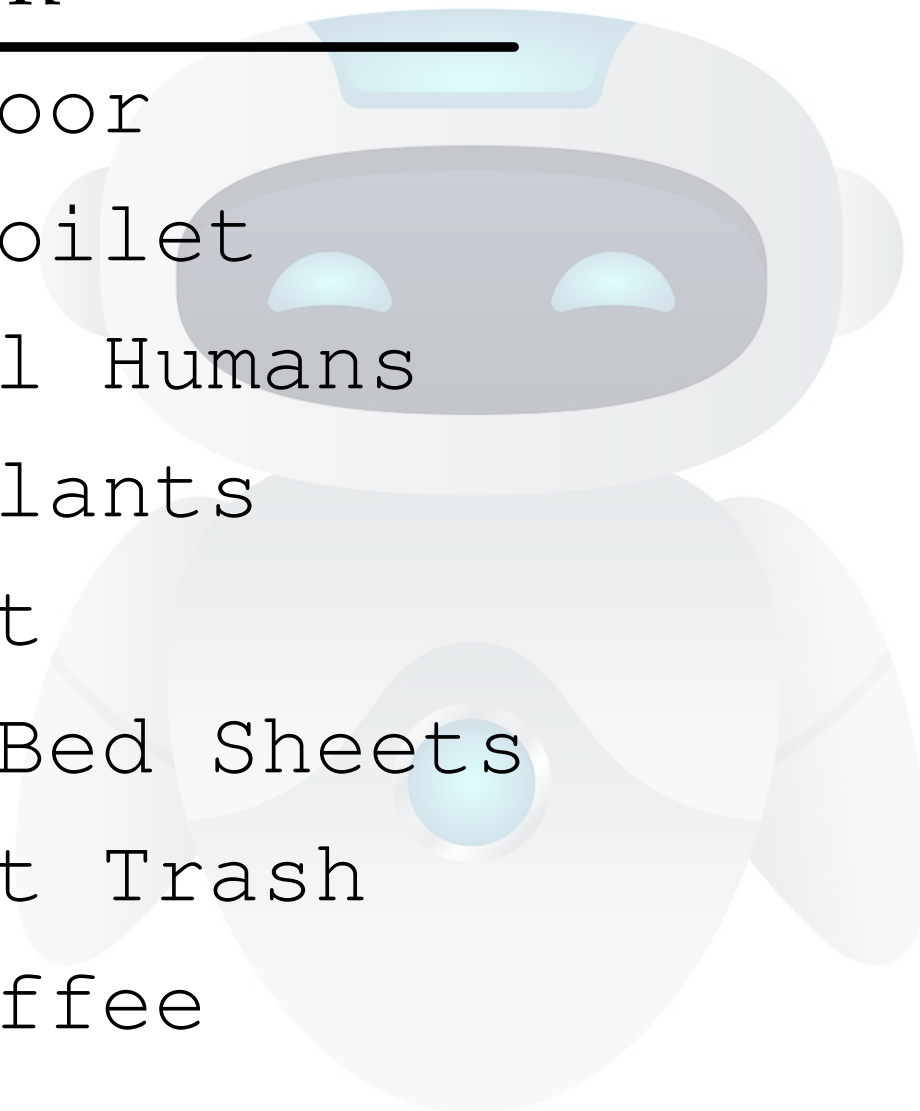
$$x_1 \oplus x_2 \oplus \cdots \oplus x_n \oplus x_{n+1} = 0$$

Binärcode	Instruktion
000	Wipe Floor
001	Clean Toilet
010	Kill All Humans
011	Water Plants
100	Feed Cat
101	Change Bed Sheets
110	Take Out Trash
111	Make Coffee



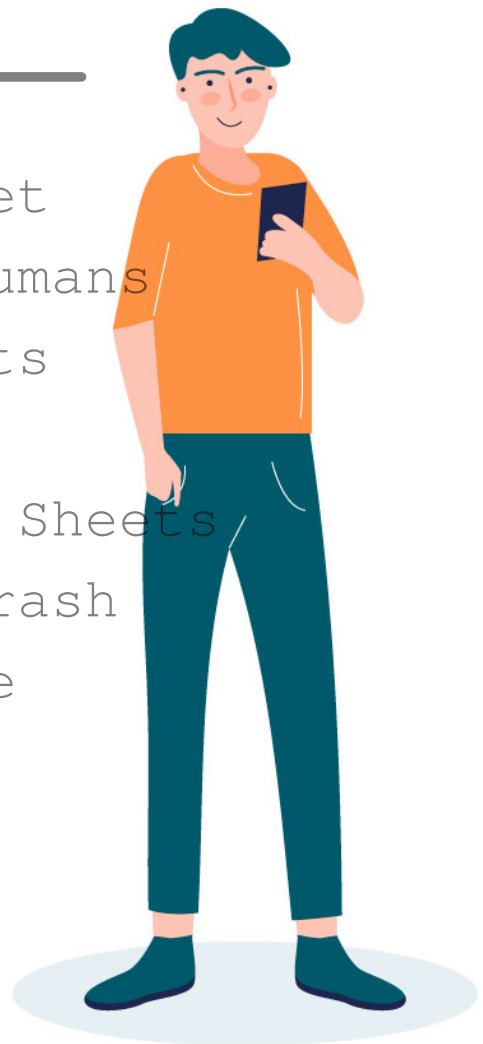
Bilder von http://www.freepik.com

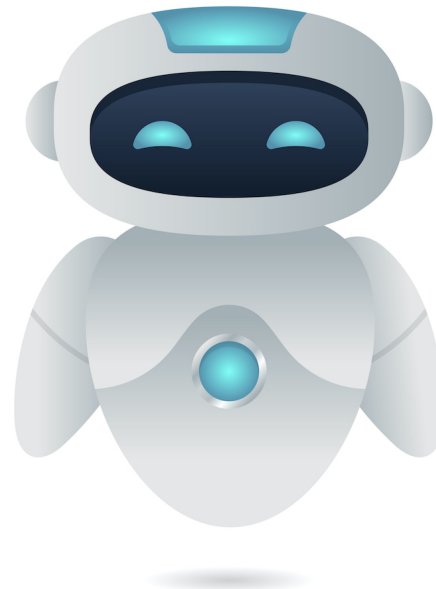
Binärcode	Instruktion
0000	Wipe Floor
0011	Clean Toilet
0101	Kill All Humans
0110	Water Plants
1001	Feed Cat
1010	Change Bed Sheets
1100	Take Out Trash
1111	Make Coffee



Bilder von http://www.freepik.com

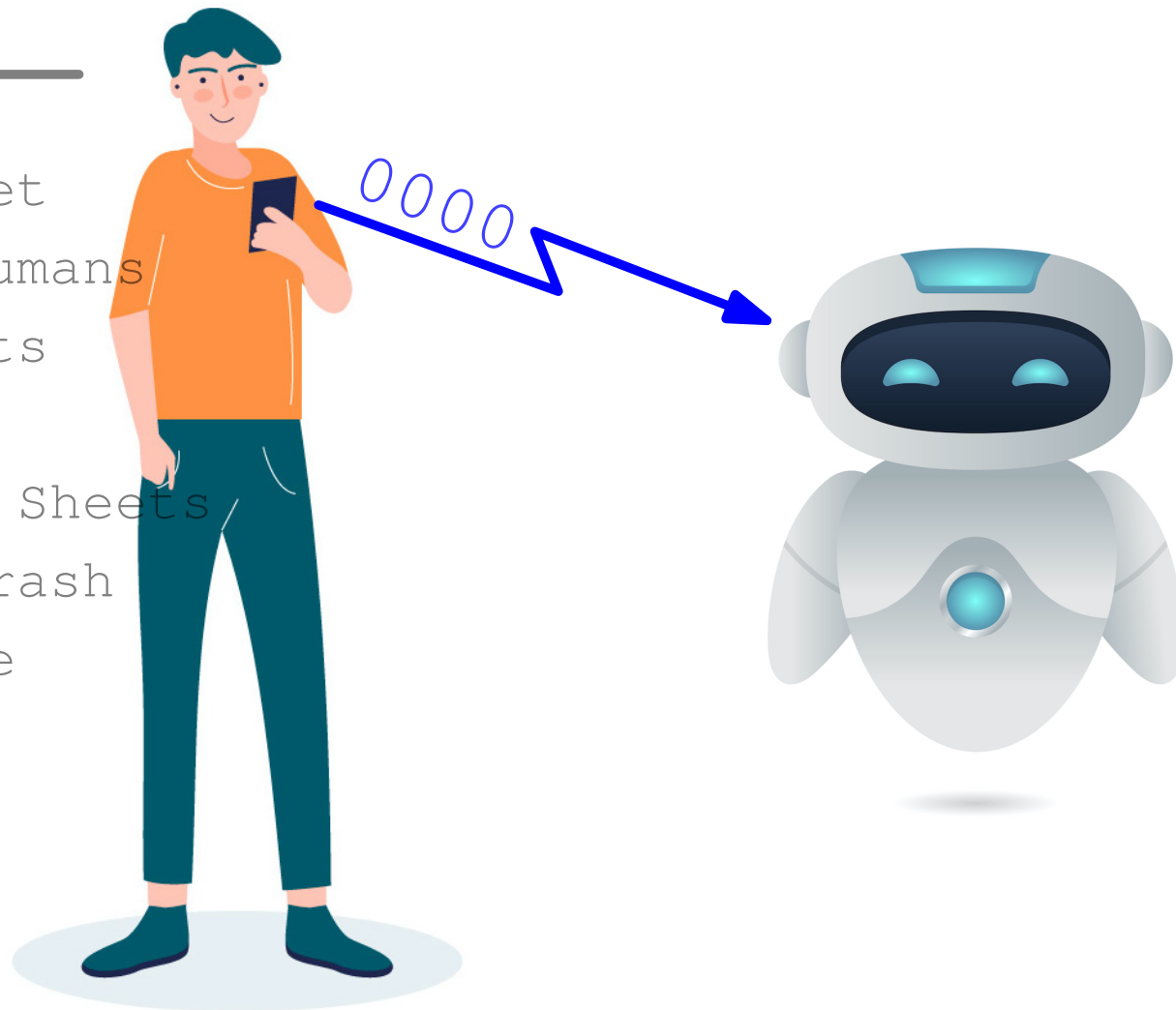
Binärcode	Instruktion
0000	Wipe Floor
0011	Clean Toilet
0101	Kill All Humans
0110	Water Plants
1001	Feed Cat
1010	Change Bed Sheets
1100	Take Out Trash
1111	Make Coffee





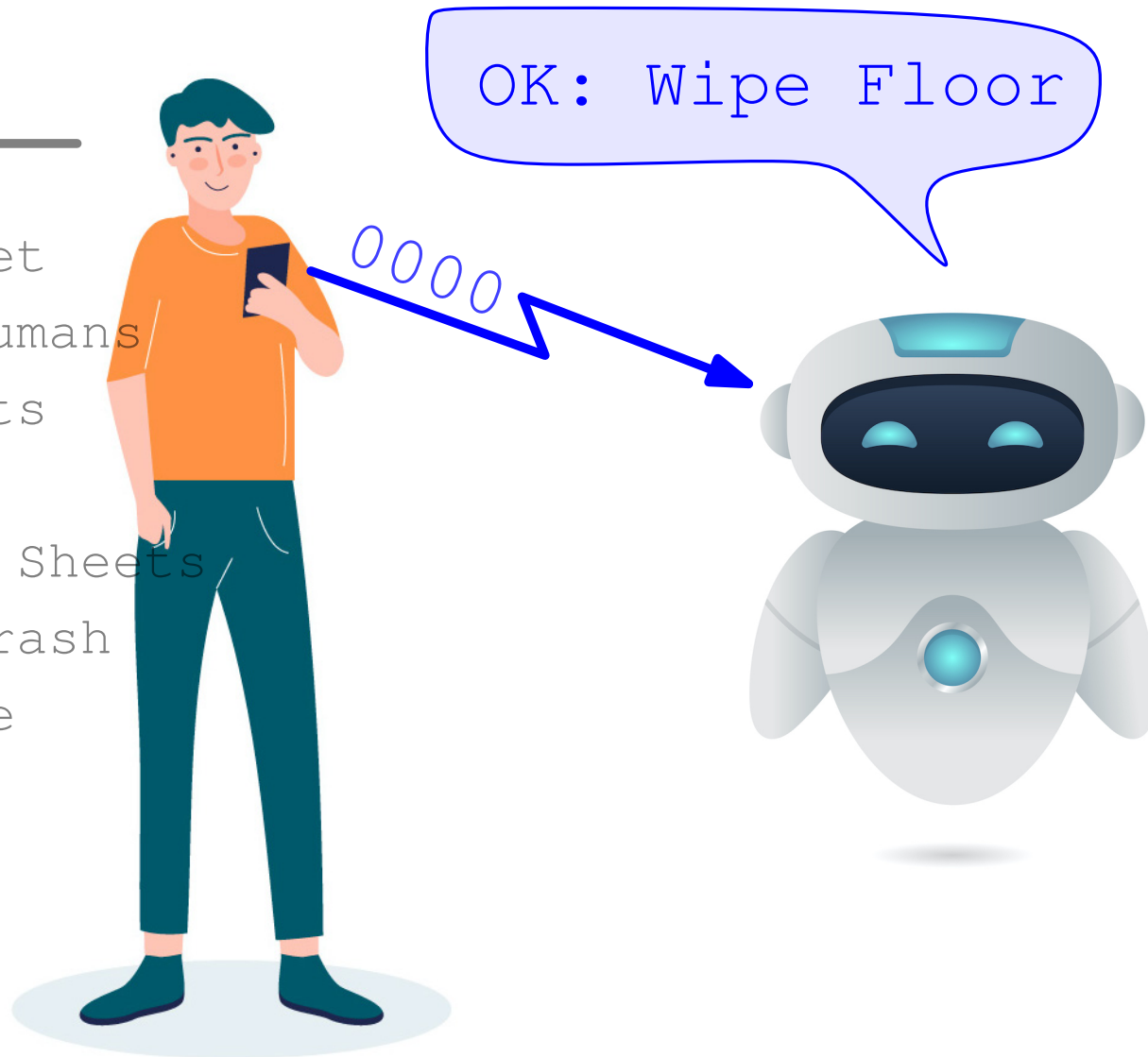
Bilder von http://www.freepik.com

Binärcode	Instruktion
0000	Wipe Floor
0011	Clean Toilet
0101	Kill All Humans
0110	Water Plants
1001	Feed Cat
1010	Change Bed Sheets
1100	Take Out Trash
1111	Make Coffee



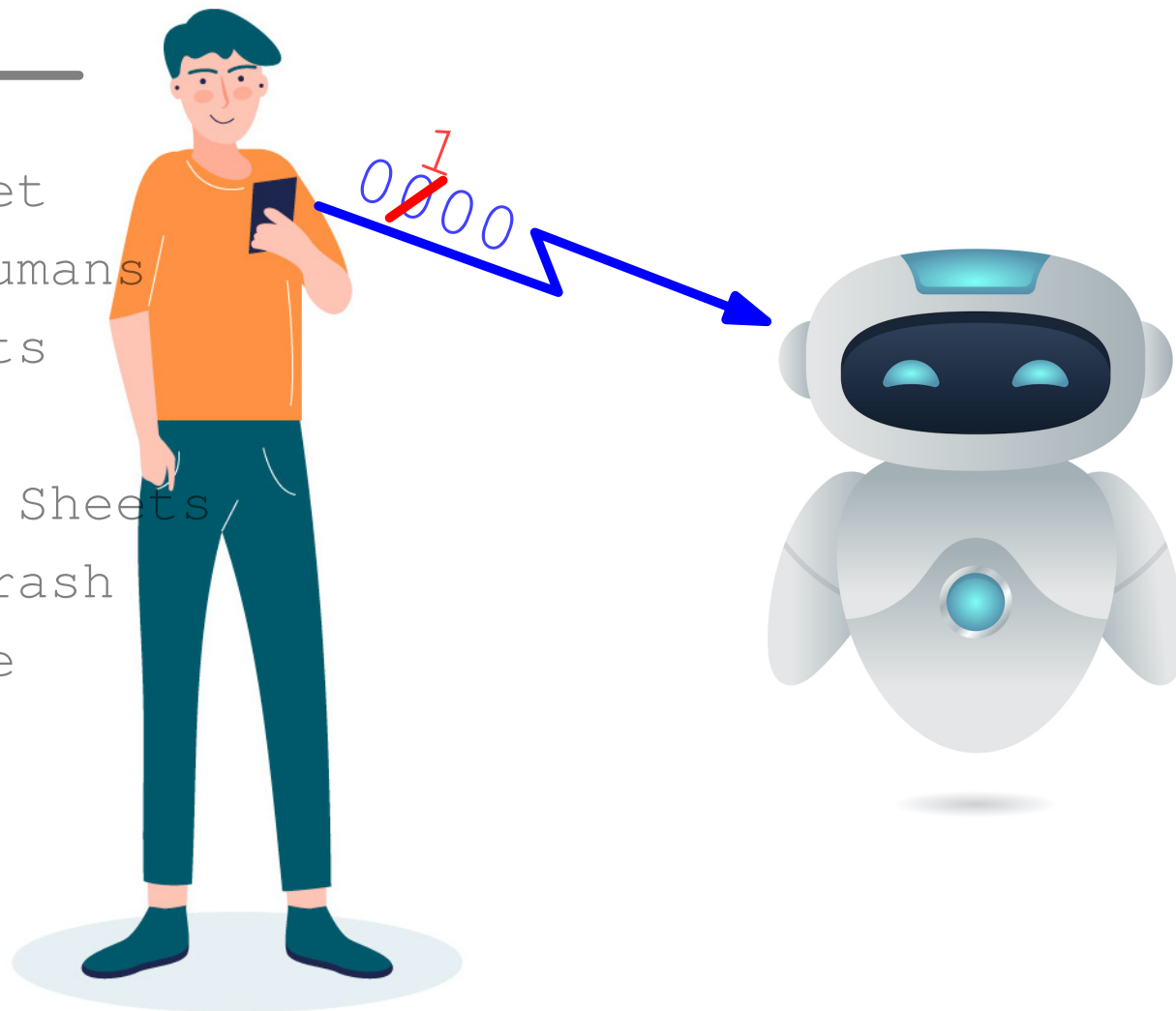
Bilder von http://www.freepik.com

Binärcode	Instruktion
0000	Wipe Floor
0011	Clean Toilet
0101	Kill All Humans
0110	Water Plants
1001	Feed Cat
1010	Change Bed Sheets
1100	Take Out Trash
1111	Make Coffee



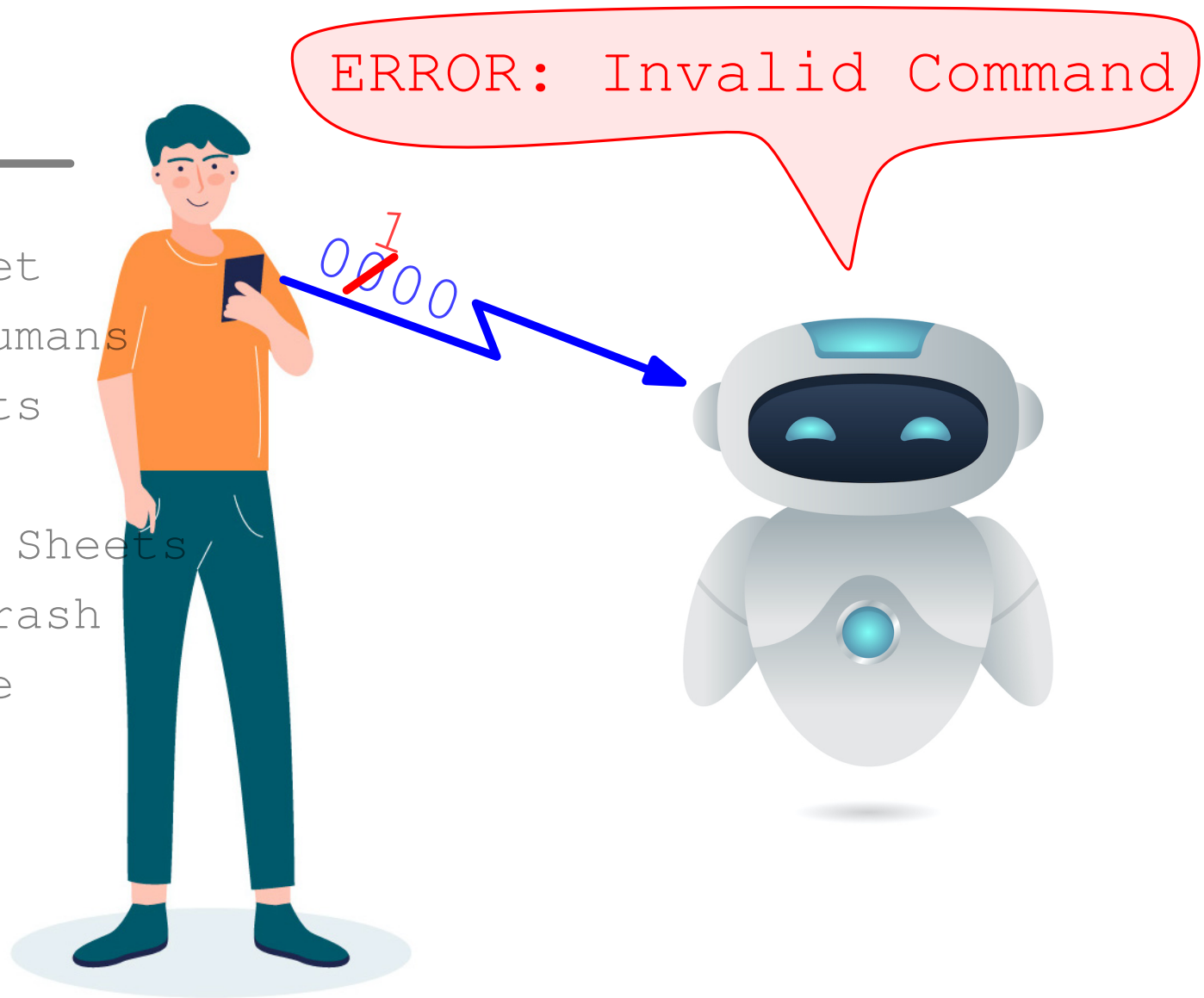
Bilder von http://www.freepik.com

Binärcode	Instruktion
0000	Wipe Floor
0011	Clean Toilet
0101	Kill All Humans
0110	Water Plants
1001	Feed Cat
1010	Change Bed Sheets
1100	Take Out Trash
1111	Make Coffee



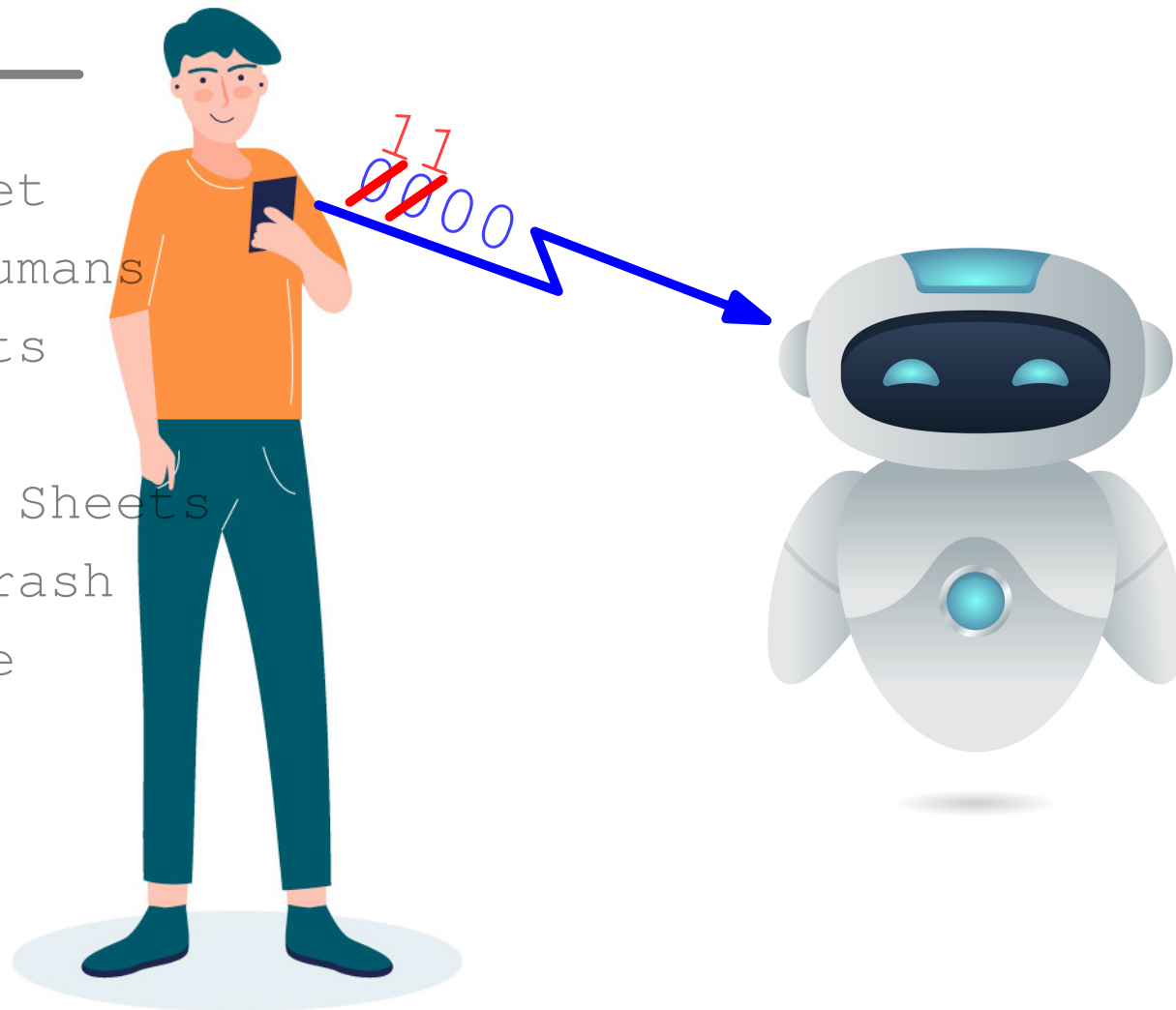
Bilder von http://www.freepik.com

Binärcode	Instruktion
0000	Wipe Floor
0011	Clean Toilet
0101	Kill All Humans
0110	Water Plants
1001	Feed Cat
1010	Change Bed Sheets
1100	Take Out Trash
1111	Make Coffee



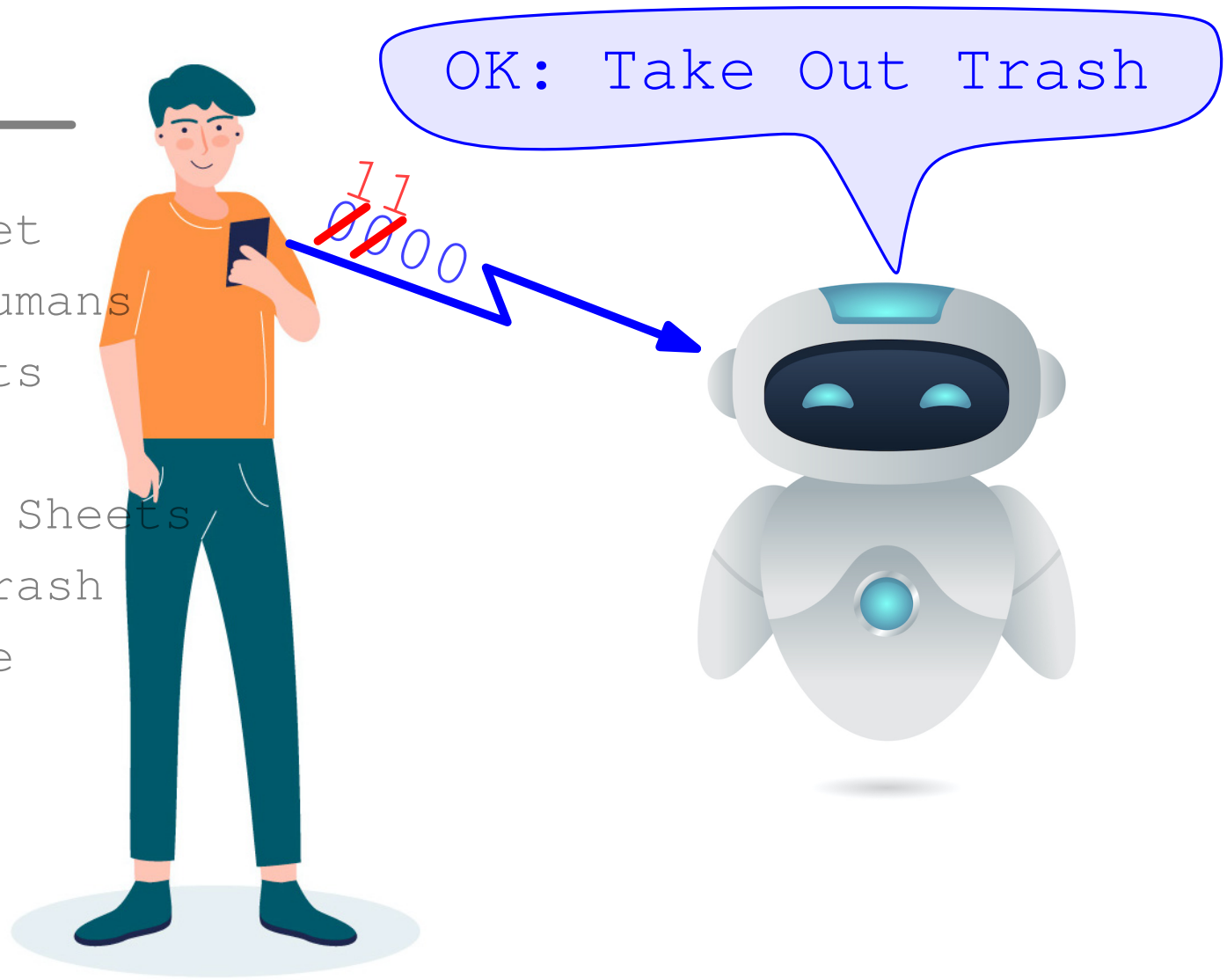
Bilder von http://www.freepik.com

Binärcode	Instruktion
0000	Wipe Floor
0011	Clean Toilet
0101	Kill All Humans
0110	Water Plants
1001	Feed Cat
1010	Change Bed Sheets
1100	Take Out Trash
1111	Make Coffee



Bilder von http://www.freepik.com

Binärcode	Instruktion
0000	Wipe Floor
0011	Clean Toilet
0101	Kill All Humans
0110	Water Plants
1001	Feed Cat
1010	Change Bed Sheets
1100	Take Out Trash
1111	Make Coffee



Bilder von http://www.freepik.com

Länge, Rate, Abstand

ein Code C

0000

0011

0101

0110

1001

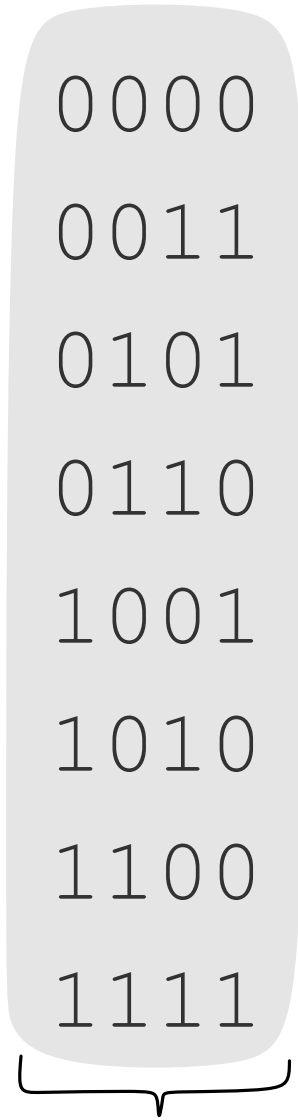
1010

1100

1111

Länge, Rate, Abstand

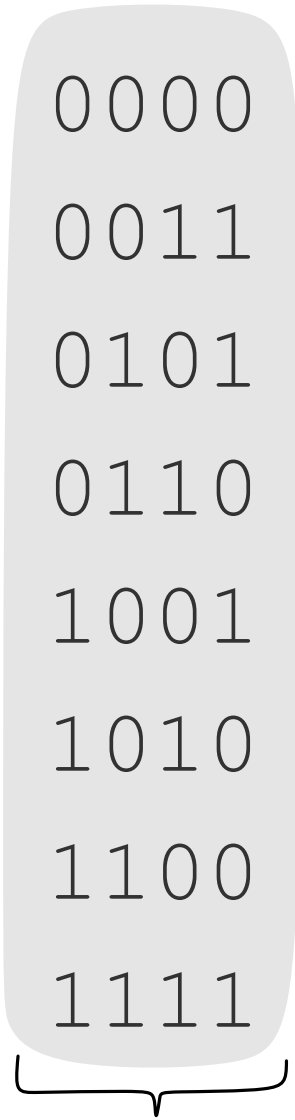
ein Code C



Die *Blocklänge* des Codes (hier: $n = 4$)

Länge, Rate, Abstand

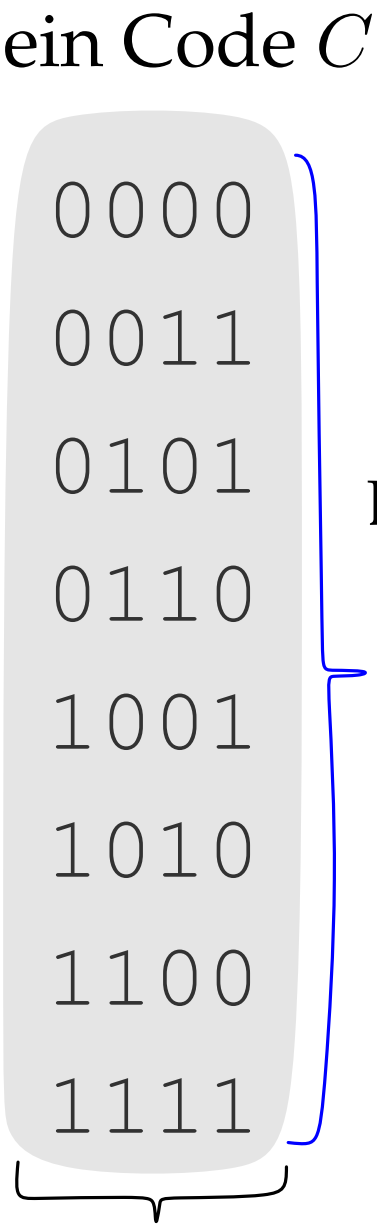
ein Code C



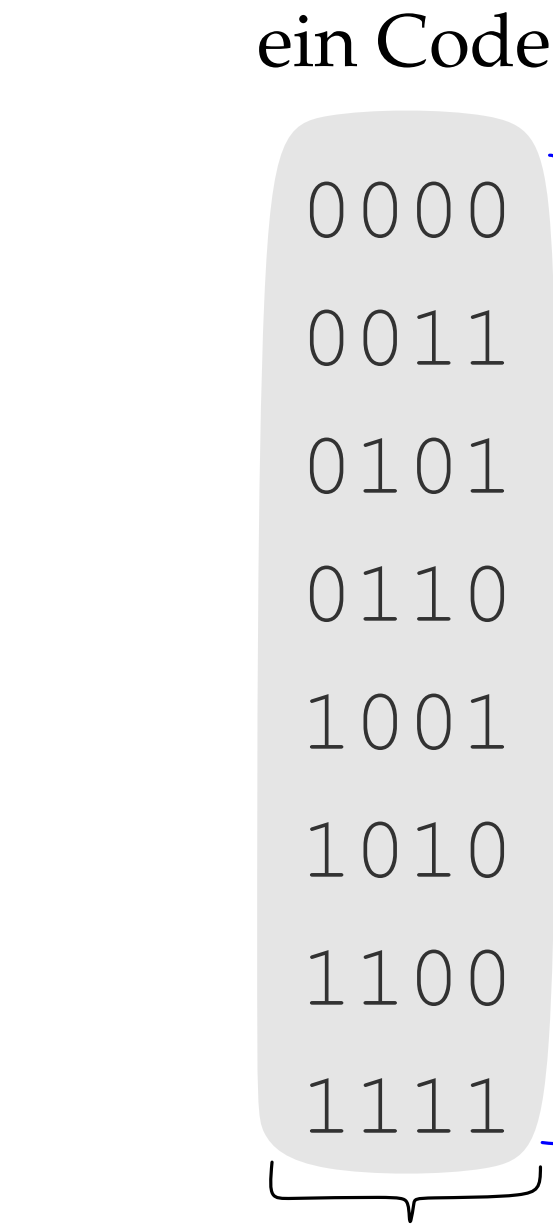
Die *Blocklänge* des Codes (hier: $n = 4$)

Kurz ist gut!

Länge, Rate, Abstand

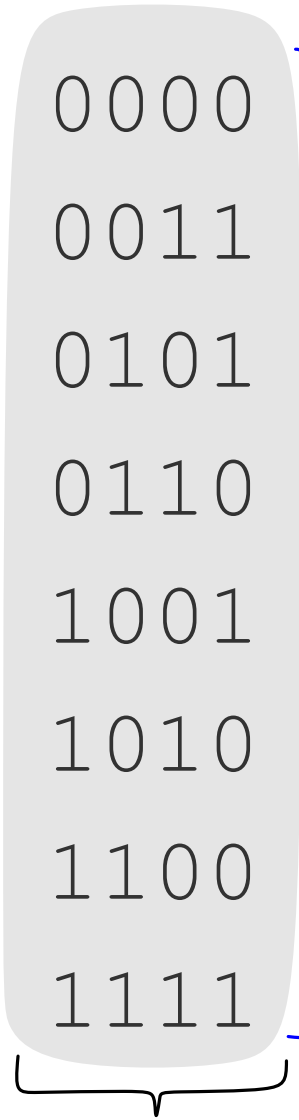


Länge, Rate, Abstand



Länge, Rate, Abstand

ein Code C



Die *Größe* des Codes (hier: $|C| = 8$)

Groß ist gut!

Anzahl der "Nutzbits": $\log_2 |C|$

(hier: $\log_2 |C| = 3$)

Nutzbit pro gesendetem Bit:

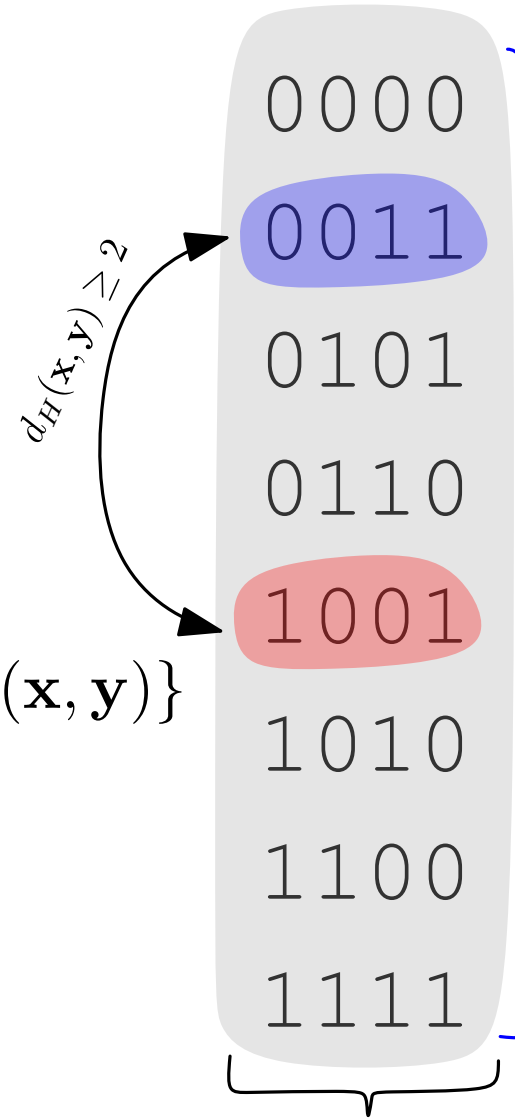
Rate $R(C) := \frac{\log_2 |C|}{n}$ (hier: $R = \frac{3}{4}$)

Die *Blocklänge* des Codes (hier: $n = 4$)

Kurz ist gut!

Länge, Rate, Abstand

ein Code C



Minimalabstand
$\Delta(C) := \min\{d_H(\mathbf{x}, \mathbf{y})\}$

(hier: $\Delta(C) = 2$)

Groß ist gut!

Die *Größe* des Codes (hier: $|C| = 8$)

Groß ist gut!

Anzahl der "Nutzbits": $\log_2 |C|$

(hier: $\log_2 |C| = 3$)

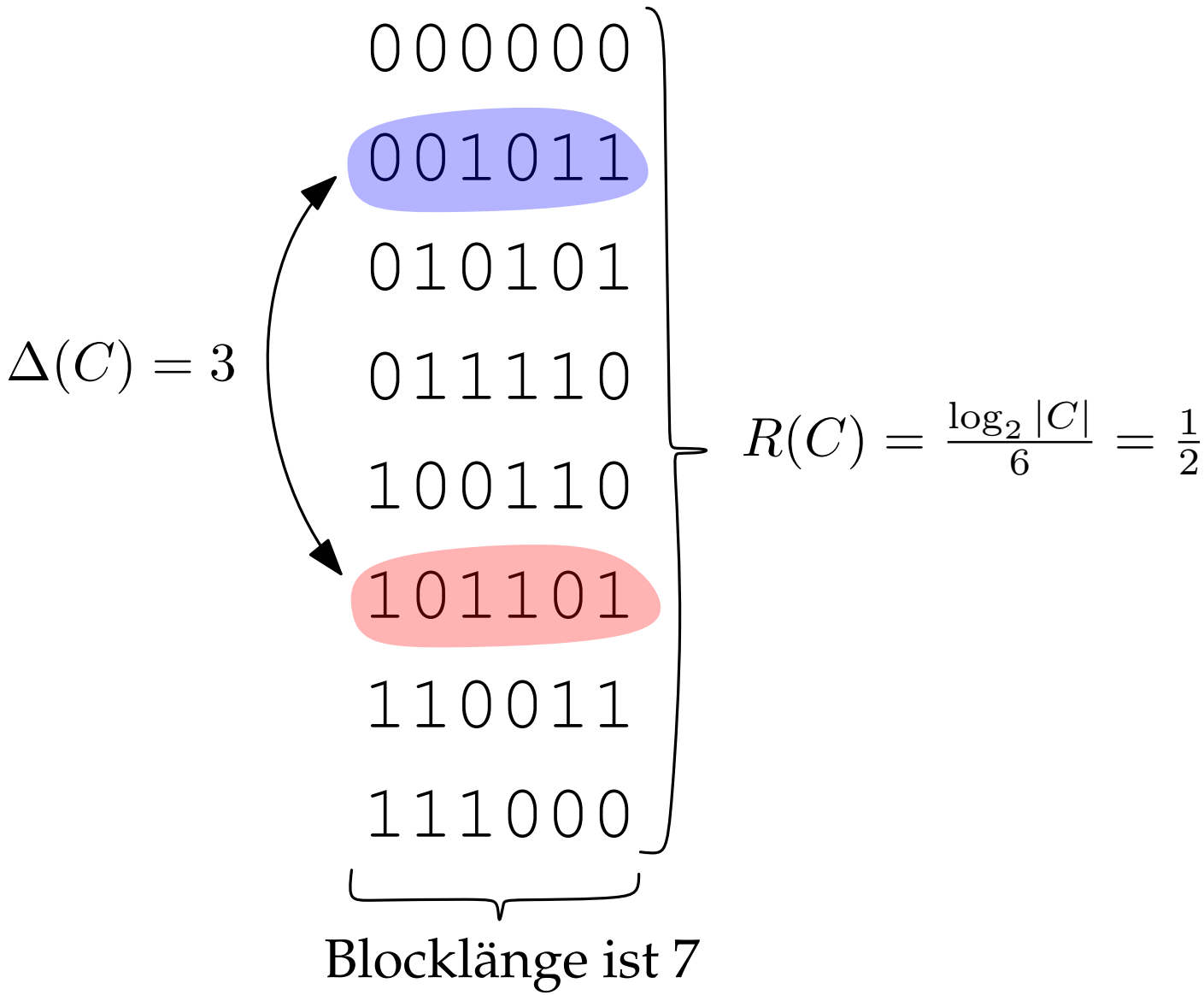
Nutzbit pro gesendetem Bit:

Rate $R(C) := \frac{\log_2 |C|}{n}$ (hier: $R = \frac{3}{4}$)

Die *Blocklänge* des Codes (hier: $n = 4$)

Kurz ist gut!

Länge, Rate, Abstand



Definition. Ein *fehlerkorrigierender Code* ist eine Teilmenge $C \subseteq \{0,1\}^n$.

- Die *Blocklänge* von C ist n.
- Der *Minimalabstand* (*minimum distance*) von C ist $\Delta(C) := \min\{d_H(\mathbf{x},\mathbf{y}) \mid \mathbf{x},\mathbf{y} \in C, \mathbf{x} \neq \mathbf{y}\}$
- Die *Rate* von C ist $R(C) := \frac{\log_2 |C|}{n}$.

Definition. Ein *fehlerkorrigierender Code* ist eine Teilmenge $C \subseteq \{0,1\}^n$.

- Die *Blocklänge* von C ist n.
- Der *Minimalabstand* (*minimum distance*) von C ist $\Delta(C) := \min\{d_H(\mathbf{x},\mathbf{y}) \mid \mathbf{x},\mathbf{y} \in C, \mathbf{x} \neq \mathbf{y}\}$
- Die *Rate* von C ist $R(C) := \frac{\log_2 |C|}{n}$.

Frage. Für gegebene Blocklänge n und Minimalabstand d, was ist der größtmögliche Code C? Wir bezeichnen die maximal mögliche Größe mit $A(n,d)$.

Definition. Ein *fehlerkorrigierender Code* ist eine Teilmenge $C \subseteq \{0,1\}^n$.

- Die *Blocklänge* von C ist n.
- Der *Minimalabstand* (*minimum distance*) von C ist $\Delta(C) := \min\{d_H(\mathbf{x},\mathbf{y}) \mid \mathbf{x},\mathbf{y} \in C, \mathbf{x} \neq \mathbf{y}\}$
- Die *Rate* von C ist $R(C) := \frac{\log_2 |C|}{n}$.

Frage. Für gegebene Blocklänge n und Minimalabstand d, was ist der größtmögliche Code C? Wir bezeichnen die maximal mögliche Größe mit $A(n,d)$.

Asymptotische Frage. Gegeben $\delta \in [0,1]$. Wie hoch kann $R(C)$ werden, wenn wir Minimalabstand $d = \delta n$ wollen und n frei wählen können? In anderen Worten: was ist

$$\sup_n \frac{\log_2 A(n,d)}{n}?$$

Was nun?

Möglichkeiten und Unmöglichkeiten

Existenz guter Codes / untere Schranken. Für jedes n und $\delta \in [0, 1]$ gibt es einen Code C mit Minimalabstand δn und Rate

$$R(C) \geq ...$$

Möglichkeiten und Unmöglichkeiten

Existenz guter Codes / untere Schranken. Für jedes n und $\delta \in [0, 1]$ gibt es einen Code C mit Minimalabstand δn und Rate

$$R(C) \geq ...$$

Unmöglichkeit / obere Schranken. Wenn C ein Code mit Blocklänge n und Abstand δn ist, dann gilt für die Rate des Codes

$$R(C) \leq ...$$

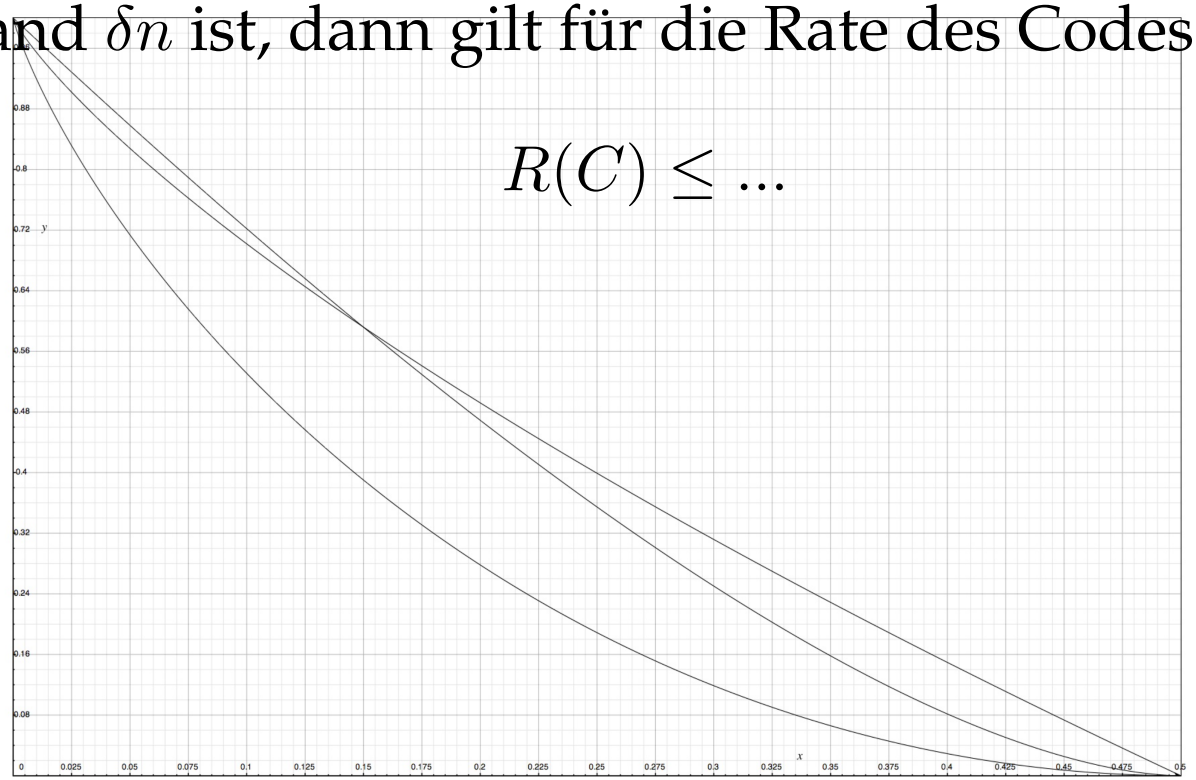
Möglichkeiten und Unmöglichkeiten

Existenz guter Codes / untere Schranken. Für jedes n und $\delta \in [0, 1]$ gibt es einen Code C mit Minimalabstand δn und Rate

$$R(C) \geq ...$$

Unmöglichkeit / obere Schranken. Wenn C ein Code mit Blocklänge n und Abstand δn ist, dann gilt für die Rate des Codes

$$R(C) \leq ...$$



Quelle: `https://www.cs.cmu.edu/~venkatg/teaching/codingtheory/notes/notes5b.pdf`

Lineare Codes

000000

001011

010101

011110

100110

101101

110011

111000

Lösungen des Gleichungssystems

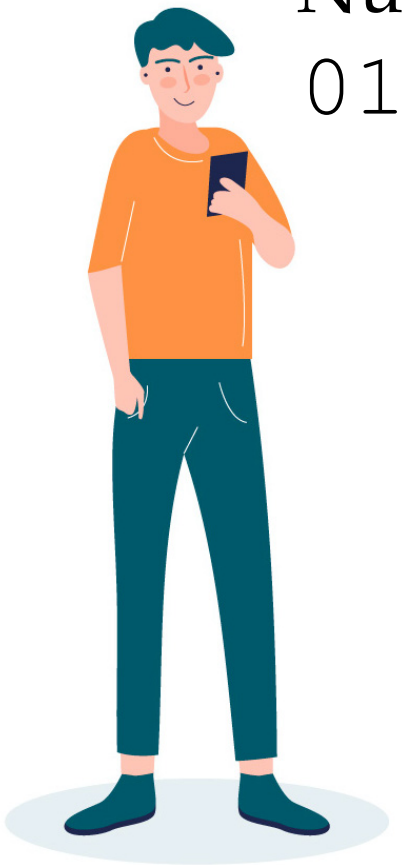
$$x_4 \oplus x_5 \oplus x_6 = 0$$

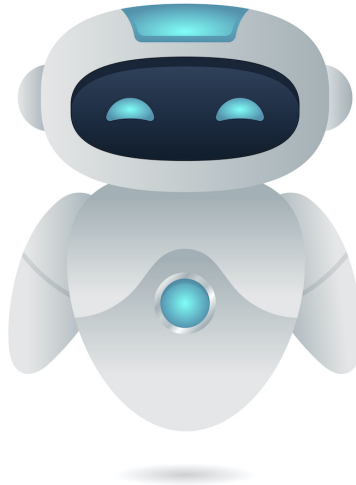
$$x_2 \oplus x_4 \oplus x_6 = 0$$

$$x_1 \oplus x_2 \oplus x_4 = 0$$

Konstruktionen / Algorithmische Fragen

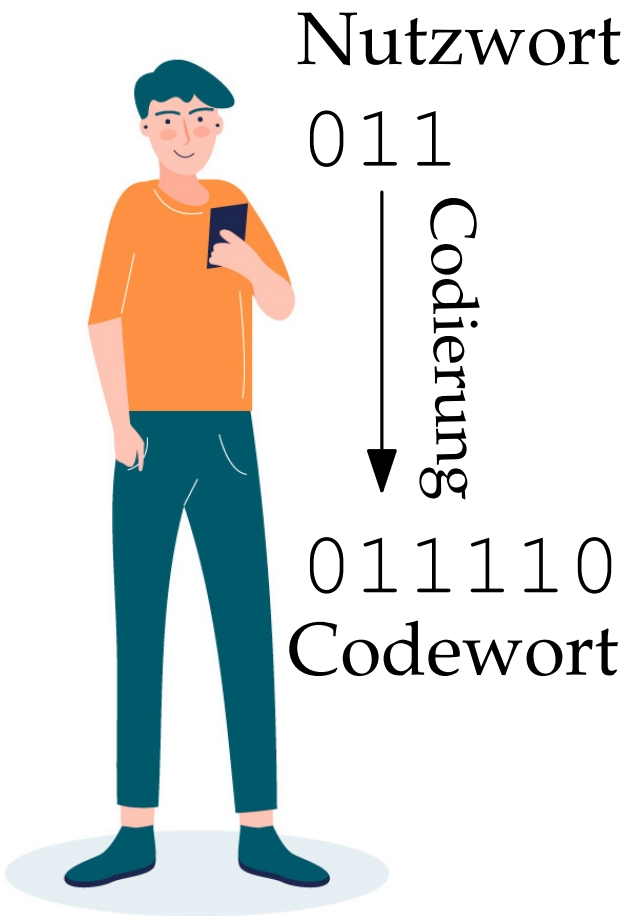
Nutzwort:
011

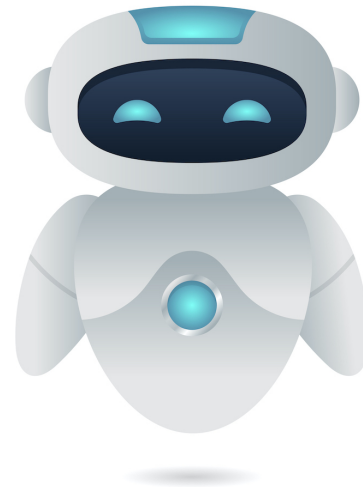




Bilder von http://www.freepik.com

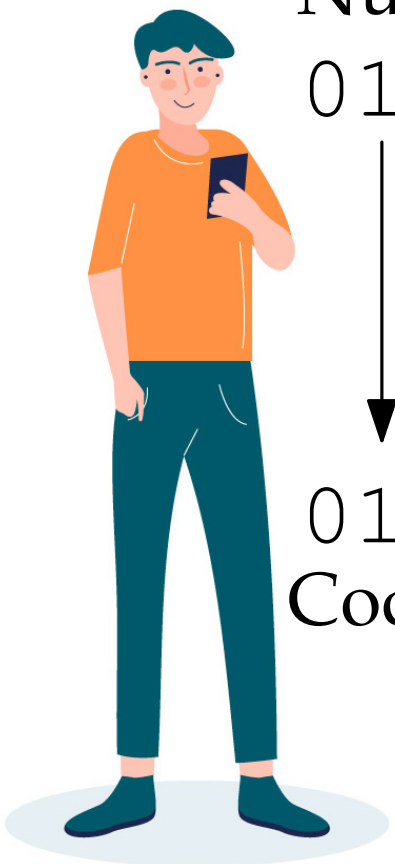
Konstruktionen / Algorithmische Fragen





Bilder von `http://www.freepik.com`

Konstruktionen / Algorithmische Fragen



Nutzwort:

011

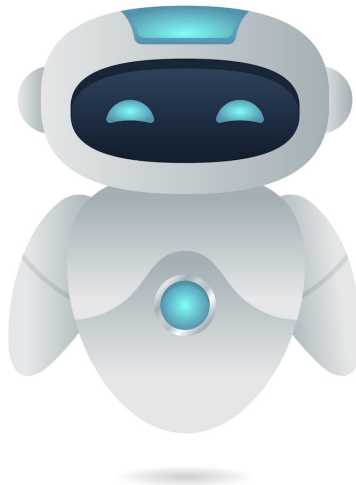
Codierung

011110

Codewort

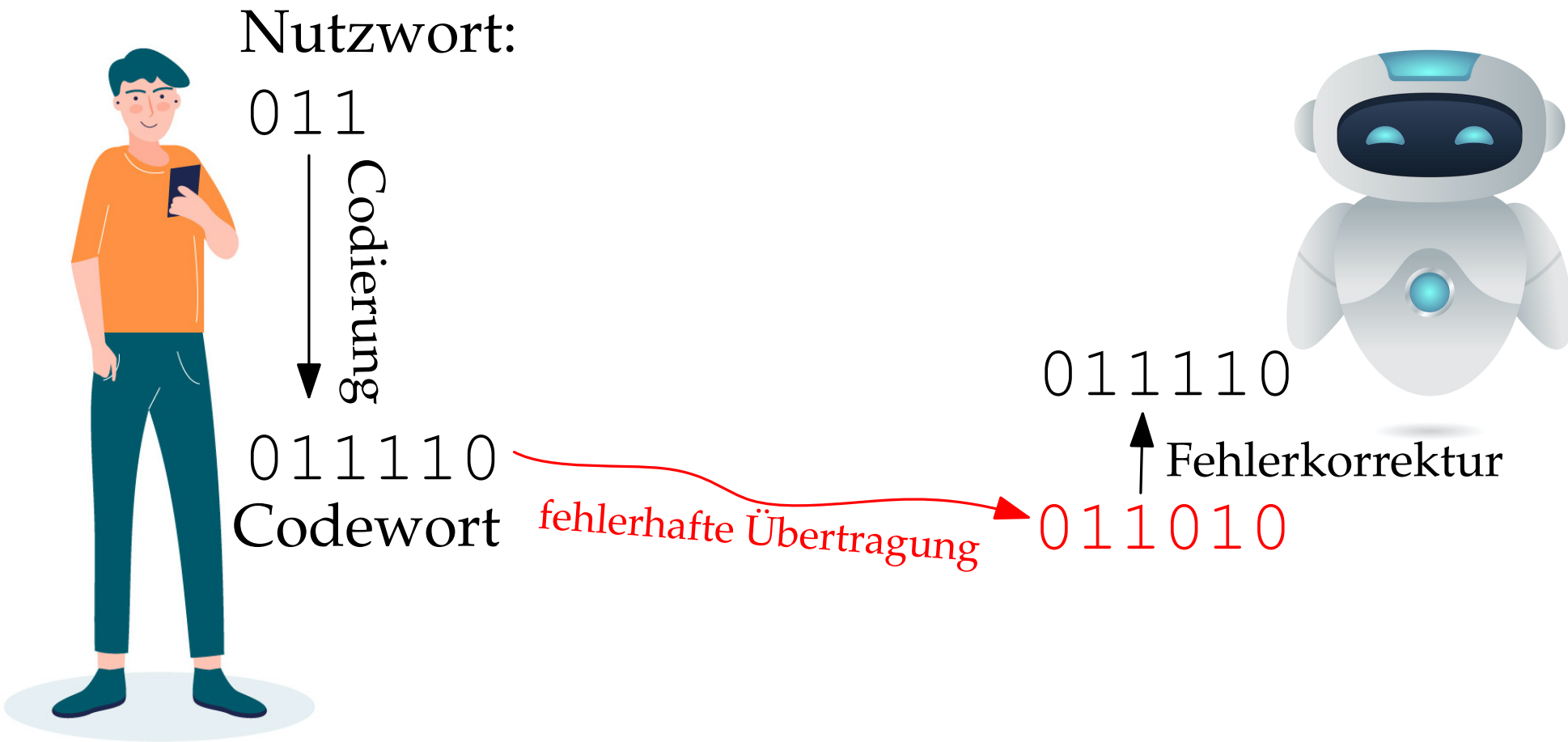
fehlerhafte Übertragung

011010



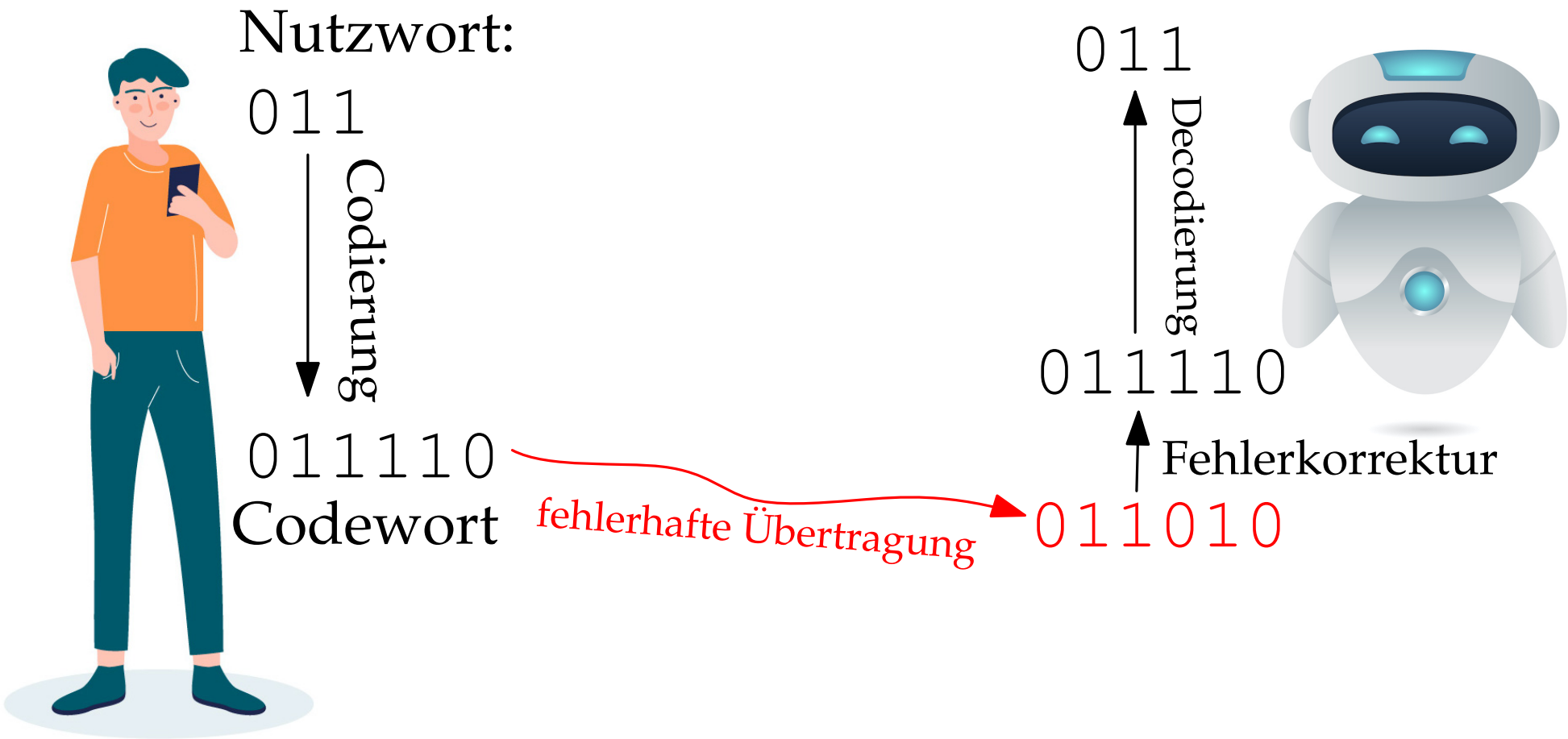
Bilder von http://www.freepik.com

Konstruktionen / Algorithmische Fragen



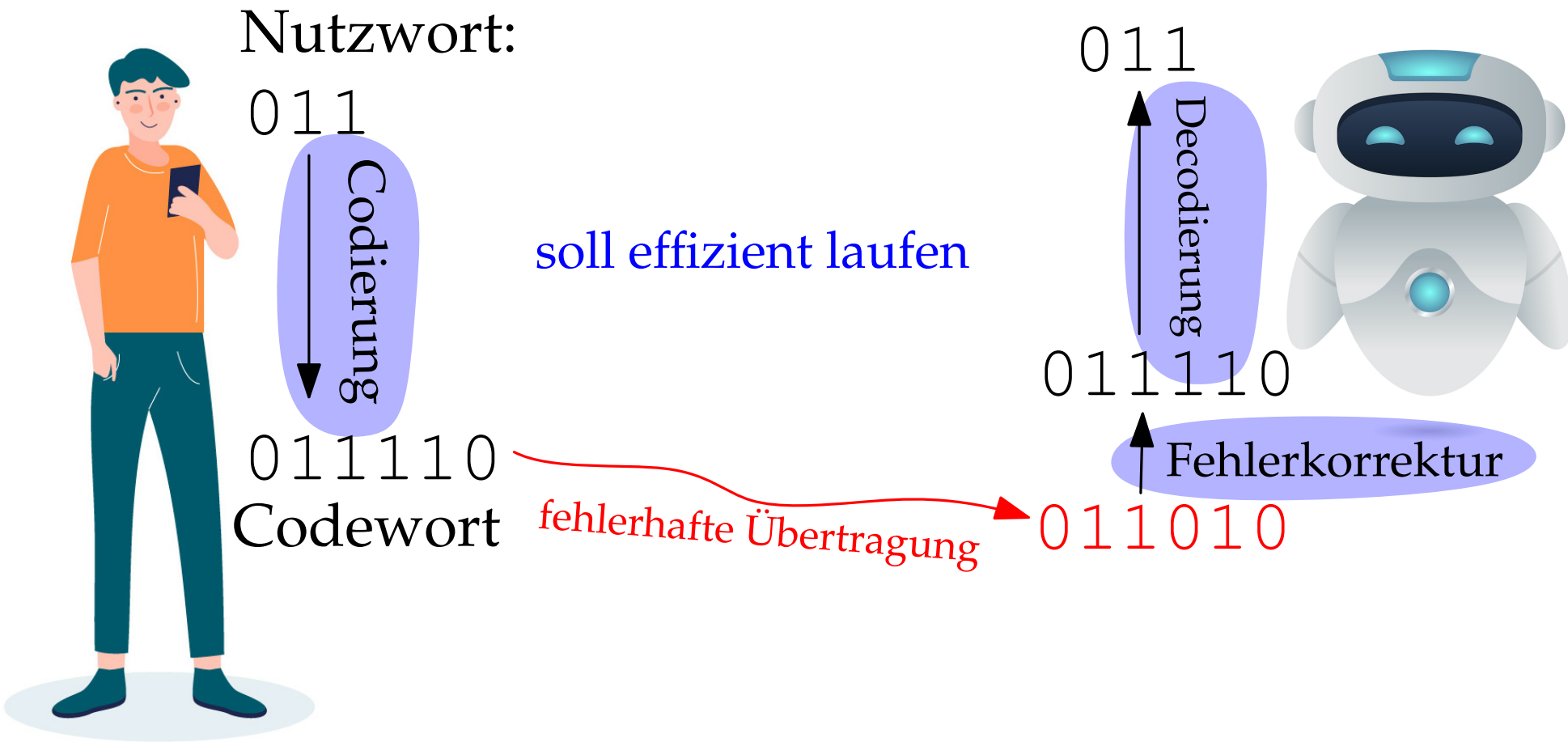
Bilder von `http://www.freepik.com`

Konstruktionen / Algorithmische Fragen



Bilder von `http://www.freepik.com`

Konstruktionen / Algorithmische Fragen



Bilder von `http://www.freepik.com`