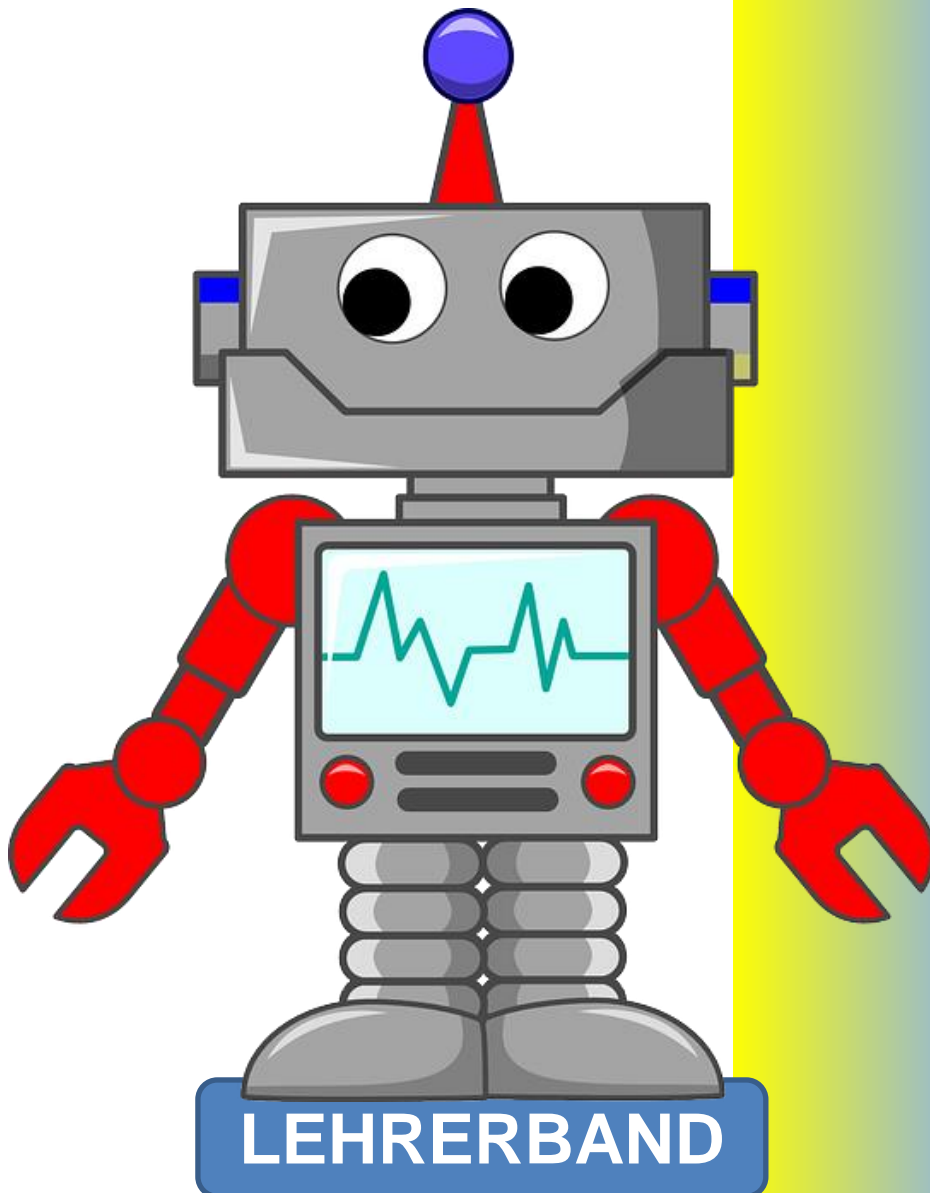


1

ROBOTER

theoretische Grundlagen



Hermann Milchram

29.11.2021

Bildnachweis:

sofern nicht anders angegeben befinden sich die Grafiken im Eigentum von NÖ Media oder sind eigene Aufnahmen und Zeichnungen oder Screenshots von Programmen.

Roboter auf der Titelseite robot-312566_640 Quelle Pixabay CC0

LEDs led-306562_640 Quelle Pixabay CC0

zielscheibe-ziel-bogenschießen-2304567 Quelle Pixabay CC0

smiley-1914523_640 Quelle Pixabay CC0

robot-1470108_640 Quelle Pixabay CC0

Herzlichen Dank für das **Korrekturlesen** an **Ilse Doppler** und **Angela Kampichler**!

INHALTSVERZEICHNIS

INHALTSVERZEICHNIS	1
PROGRAMMIERBARE ROBOTER	2
Was bringt die Beschäftigung mit programmierbaren Robotern?	3
Was sind Roboter?	4
BAUTEILE EINES ROBOTERS	5
AB3_VERGLEICH MENSCH – ROBOTER LÖSUNG	6
AB4_VON SENSOREN UND AKTOREN LÖSUNG	7
LEDs	8
WIE ARBEITEN COMPUTER	9
Dualsystem (Binärsystem).....	9
Umrechnung von Dezimalzahlen in Binärzahlen.....	11
Andere Zahlensysteme in der EDV	13
ANALOG – DIGITAL	14
Umwandlung analoger Signale.....	15
Fuzzy Logic	15
PROGRAMMIERUNG	16
Verschiedene Arten von Programmiersprachen	16
Grundelemente einer modernen Programmiersprache.....	17
Arten der Programmierung	17
Typische Programmierfehler	18
Phasen der Programmentwicklung.....	18
Visuelle Programmiersprachen (VPL)	19

Programmierbare Roboter

vom Kindergarten bis zum Ende der Schulpflicht

Computertechnik be-greifen! IT-Unterricht ab Kindergarten und Vorschulstufe?

Unsere Kinder sind heute von mehr Technik umgeben als irgendeine Generation davor. Videospiele, PCs, Tablets und Smartphones sind im Leben der meisten SchülerInnen, aber auch vieler Kindergartenkinder bereits allgegenwärtig. Das alles sind Formen von Kommunikation, die ihr Leben Tag für Tag beeinflussen. Das Verständnis für die Technik dahinter aber ist kaum vorhanden. Was im anglo-amerikanischen Raum und einigen ost- und nordeuropäischen Ländern bereits selbstverständlich ist, hält nur in sehr zögerlichem Ausmaß, meist durch Einzelinitiativen engagierter Pädagoginnen und Pädagogen, Einzug in den Alltag unserer Sprösslinge.

„Wer aktuelle Technologien verstehen, produktiv nutzen und vielleicht sogar mitgestalten will, benötigt ein grundsätzliches Verständnis für die Art und Weise, wie die Welt der Computer funktioniert. Nur wer versteht, wie Computer „ticken“, kann auch morgen noch kompetent handeln.“ (c't 2014, Heft 14)

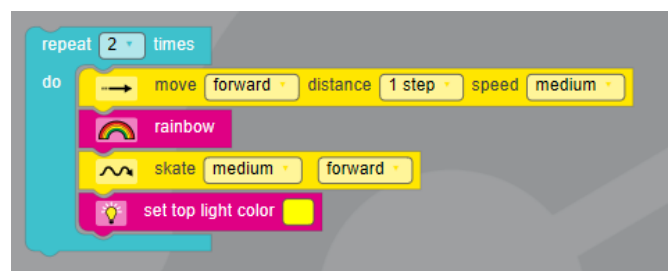
„Programm or be programmed“
(Medientheoretiker Douglas Rushkoff)

Informatikkenntnisse

- helfen beim Lösen von Problemen in allen Lebensbereichen
- vermitteln eine Reihe von Denkwerkzeugen, die sich auch auf komplexe Probleme im Alltag anwenden lassen.

Die amerikanische Computerwissenschaftlerin *Jeannette M. Wing* prägte dafür den Begriff **„Computational Thinking“**, welchen sie als vierte Grundfertigkeit dem Lesen, Schreiben und Rechnen hinzufügen möchte.

Programmieren bedeutet Information in eine Form zu bringen, die von einem Computer verstanden werden kann. Programmierung spielt in vielen Bereichen unseres Alltags eine wesentliche Rolle, ohne dass wir uns dessen bewusst sind: die Programmierung der Mikrowelle, der Waschmaschine, des Backofens oder eines SAT-Empfängers sind nur einige wenige Beispiele dafür. Das heutige Programmieren hat vielfach nur mehr wenig mit dem Schreiben von Computerroutinen in Programmiersprachen wie BASIC, PASCAL oder gar ASSEMBLER zu tun. Moderne, für den Unterricht entwickelte Programmiersprachen, wie **Scratch** <https://scratch.mit.edu/> oder **Blockly** <https://blockly-games.appspot.com/> sind Generatoren mit einer grafischen Oberfläche, in der Programmabläufe per **DRAG&DROP** zusammengefügt werden. Es ist fesselnd, führt rasch zu Ergebnissen und macht Spaß. Trotzdem werden dabei die grundlegenden Konzepte der Programmierung vermittelt und die Fähigkeit Probleme zu lösen gestärkt. Eine zeitgemäße, auf das Berufsleben vorbereitende Ausbildung kommt am Thema Informatik nicht vorbei. Aber ab wann kann und soll informatische Bildung beginnen? Viele Gründe sprechen dafür, Informatik möglichst früh auf den Stundenplan zu



Roboter im Unterricht

setzen. Eines der wesentlichsten Argumente ist, dass jenseits des Unterrichts Smartphones, Tablets, Computer und auch Roboter längst zum Alltag unserer Kinder gehören.

Erstmals gibt es ab dem SJ 2018/19 für die NMS und AHS Unterstufe ein verbindliches Fach „**Digitale Grundbildung**“, in dem unter anderem auch der Begriff des „Computational Thinking“ verankert ist.

Computational Thinking			
Mit Algorithmen arbeiten	<ul style="list-style-type: none">– nennen und beschreiben Abläufe aus dem Alltag, verwenden, erstellen und reflektieren Codierungen (z.B. Geheimschrift, QR-Code),– vollziehen eindeutige Handlungsanleitungen (Algorithmen) nach und führen diese aus,– formulieren eindeutige Handlungsanleitungen (Algorithmen) verbal und schriftlich.	<ul style="list-style-type: none">– entdecken Gemeinsamkeiten und Regeln (Muster) in Handlungsanleitungen,– erkennen die Bedeutung von Algorithmen in automatisierten digitalen Prozessen (z.B. automatisiertes Vorschlagen von potenziell interessanten Informationen).	<ul style="list-style-type: none">– können intuitiv nutzbare Benutzeroberflächen und dahinterstehende technische Abläufe einschätzen.
Kreative Nutzung von Programmiersprachen	<ul style="list-style-type: none">– erstellen einfache Programme oder Webanwendungen mit geeigneten Tools, um ein bestimmtes Problem zu lösen oder eine bestimmte Aufgabe zu erfüllen,– kennen unterschiedliche Programmiersprachen und Produktionsabläufe.	<ul style="list-style-type: none">– beherrschen grundlegende Programmierstrukturen (Verzweigung, Schleifen, Prozeduren).	<ul style="list-style-type: none">– reflektieren die Grenzen und Möglichkeiten von Simulationen.

Das Vermitteln von grundlegenden Konzepten der Informatik in Kindergarten, Grundschule und Unterstufe stellt Pädagogen jedoch vor große Herausforderungen. Da das abstrakte Denkvermögen erst langsam entwickelt wird, muss der didaktische Ansatz dahingehen, die abstrakten Konzepte der Computertechnologie im wahrsten Sinne des Wortes begreifbar zu machen. Hier können die auf den folgenden Seiten vorgestellten programmierbaren Roboter eine wichtige Unterstützung bieten.

Was bringt die Beschäftigung mit programmierbaren Robotern?



- fördert die Entwicklung von Problemlösungsstrategien
- stärkt die Fähigkeit Fehler zu finden und zu korrigieren
- fördert das analytische und logische Denken
- fördert das kollaborative Arbeiten
- stärkt die Kommunikations- und Diskussionsfähigkeit
- hilft bei der Entwicklung des räumlichen Denkens

Noch vor einigen Jahrzehnten waren Roboter bloß Science-Fiction, heute sind wir auf Schritt und Tritt von diesen modernen Helferleins umgeben. Sie bauen Autos, unterstützen Ärzte bei Operationen, entschärfen Bomben, tauchen in die Tiefen der Ozeane und erkunden für uns den Weltraum, mähen für uns den Rasen (**Mähroboter**) und übernehmen Reinigungsarbeiten im Haus (**Staubsaugroboter**). Auch die Raumfahrt ist bei ihren Missionen auf die Unterstützung von Robotern angewiesen. Auf unseren Straßen sind bereits die ersten autonom fahrenden Fahrzeuge unterwegs.



NÖ Media Streaming → **Roboter: Helfer des Menschen**
<https://bit.ly/2QOGXZ0>



<http://www.zukunftstechnologien.info/technik-und-wirtschaft/robotik/>

Roboterfreund für einsame Kinder <https://bit.ly/2vQUxCg>

Planet Wissen „Roboter“
<https://bit.ly/2vNcTEn>



AB1_Ein Blick in den Garten Lösung

Beschreibe, was du auf dem Bild siehst.
 Wozu sind die verschiedenen Teile da?

1. Welche Arbeit führt der Roboter auf dem Bild aus?
2. Wer hat dies früher gemacht?
3. Welches Werkzeug wurde dazu benötigt?
4. Wer steuert den Roboter auf dem Bild?
5. Woher weiß er, was er tun soll?
6. Kennst du noch andere Roboter?
7. Welche Arbeiten erledigen diese?
8. Erkläre mit eigenen Worten, was alle Roboter gemeinsam haben.

Was sind Roboter?



Ein Roboter ist eine Maschine, die imstande ist selbstständig Aufgaben zu erfüllen. Dazu verfügt der Roboter über definierte Schnittstellen, um seine Umwelt zu erfassen, mit ihr zu kommunizieren und sie gegebenenfalls auch zu verändern.



NÖ Media Streaming → **Was ist was: Computer und Roboter**
<https://bit.ly/2DmtiG6>

Filmausschnitt:
 WALL-E's "Day At Work"
<https://bit.ly/1AuQI91>



Beschreibung eines Roboters:

1. Er arbeitet nach programmierten Anweisungen.
2. Er nimmt die Umgebung mit Sensoren wahr.
3. Er kann verschiedene Tätigkeiten selbständig ausführen.
4. Er wird durch einen Prozessor (Teil vom Computer) gesteuert.



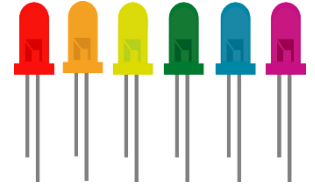
AB2_Was sind Roboter Lösung

Bauteile eines Roboters

Ein Roboter braucht:



- ✓ **Sensoren** → Lichtsensor, Temperaturfühler, Drucksensor, Lagesensor, Ultraschallsensor, Infrarotsensor etc. nehmen die Umwelt wahr und wandeln diese Informationen in elektrische Impulse verschiedener Stärke und Dauer um.
- ✓ **Aktoren** → elektrische Impulse werden in mechanische Bewegungen umgewandelt, um Motoren, Greifarme usw. zu bewegen
- ✓ **LEDs (Light Emitting Diodes)** → können sowohl als **Aktoren** (Ausgabe von Lichtsignalen in unterschiedlichen Farben) als auch als **Sensoren** zur Erkennung des Umgebungslichts verwendet werden.
- ✓ **Schnittstellen für die Kommunikation**



USB-Interface: dient der Verbindung zum Computer und wird meist auch zum Aufladen verwendet



WLAN: Erlaubt ähnlich wie Bluetooth den drahtlosen Datenaustausch zwischen verschiedenen Endgeräten und deinem Roboter.

Vorteil: Ist schneller und geht auch über größere Distanzen.



Bluetooth: Über Bluetooth können Signale drahtlos mit PCs, Tablets und Smartphones ausgetauscht werden. Bevor du Bluetooth zum Datenaustausch verwenden kannst, musst du deinen Roboter mit dem gewünschten Endgerät koppeln!

- ✓ **Mikroprozessor** → steuert die Aktoren und Sensoren
- ✓ **künstliche Intelligenz** (Software = Programme).



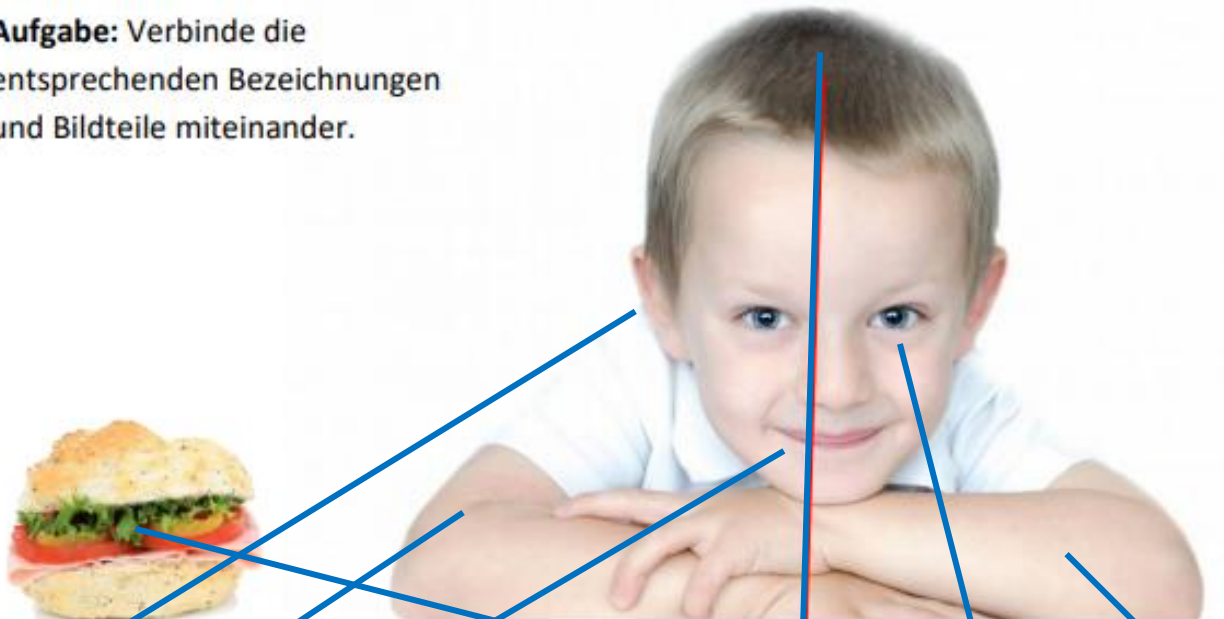
Roboter arbeiten nach dem **EVA**-Prinzip



AB3_Vergleich Mensch – Roboter
AB4_Von Sensoren und Aktoren

AB3_Vergleich Mensch – Roboter Lösung

Aufgabe: Verbinde die entsprechenden Bezeichnungen und Bildteile miteinander.



Ohr Haut Mund, Handzeichen Nahrung Gehirn Auge Muskeln

Daten verarbeiten

hören

sehen

spüren

sprechen, Zeichen geben

bewegen

Energie nutzen

Prozessor

Mikrofon

Lichtsensoren, Kamera

Temperatursensor, Drucksensoren

Lautsprecher, Lichtzeichen

Motoren

Akku / Batterie



[Von spannenden Problemen zu kreativen Lösungen](#) PH Luzern

AB4_Von Sensoren und Aktoren Lösung

Ein Sensor ist eine passive Komponente, er erfasst Dinge.

→ Sensoren sind in etwa vergleichbar mit den fünf Sinnen beim Menschen.

Ein Aktor ist eine aktive Komponente, er bewegt Dinge oder führt etwas aus.

→ Aktoren sind zum Beispiel vergleichbar mit dem Gehen, dem Sprechen oder der Mimik beim Menschen.

Aufgabe: Kreuze an, welche der folgenden Bauteile Sensoren und welche Aktoren sind.

Tipp: Es gibt auch Bauteile, die nichts von beidem sind.



Sensor
 Aktor



Sensor
 Aktor



Sensor
 Aktor



Sensor
 Aktor



Sensor
 Aktor



Sensor
 Aktor



Sensor
 Aktor



Sensor
 Aktor



Sensor
 Aktor



Sensor
 Aktor

Ein Roboter besteht aus verschiedenen Sensoren und Aktoren.

[Von spannenden Problemen zu kreativen Lösungen](#) PH Luzern

LEDs

Am PC oder bei RGB-LEDs kann eine Farbe durch ihre Anteile an den drei Primärfarben **R**ot, **G**rün und **B**lau definiert werden.

Eigenschaften der Grundfarben (Primärvalenzen)

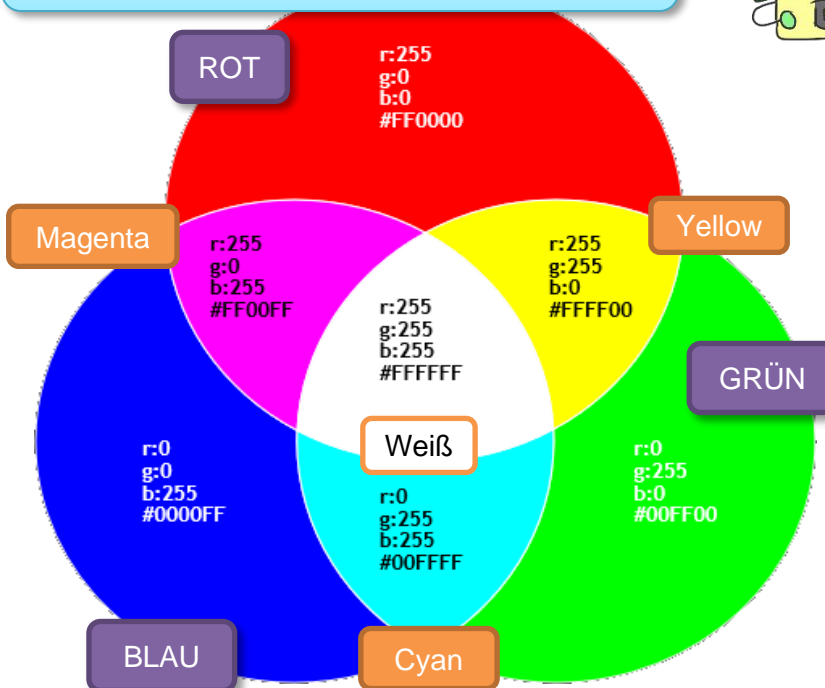
- ✓ Grundfarben können nicht aus anderen Farben gemischt werden
- ✓ Durch Mischen der drei Grundfarben können alle anderen Lichtfarben erzeugt werden

Additiver Farbmischer:

http://www.spectrumcolors.de/cor_rgb_demo.php



AB5_LEDs
Farbmischung Lösung



Dieses als **RGB-System** bezeichnete Verfahren baut auf der **additiven Farbmischung** auf. Jede Farbe wird durch ihren Anteil an den drei Primärfarben (**Rotwert** 0-255, **Grünwert** 0-255, **Blauviolett** 0-255) definiert.

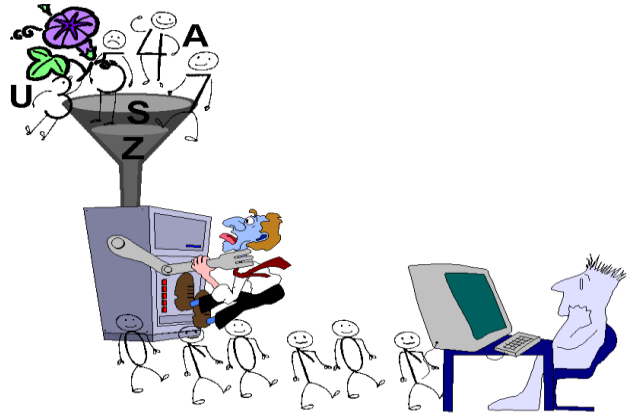
Die drei Farbwerte werden jeweils durch ein **Byte** = 8 Bits dargestellt und können somit die Werte von 0 bis 255 annehmen. Wird eine Grundfarbe für die Darstellung des Farbtons nicht benötigt, beträgt der entsprechende Zahlenwert 0. Da jede der drei Grundfarben in 256 Stufen (0 - 255) dargestellt werden kann, sind insgesamt $256^3 = 256 \times 256 \times 256 = 16\,777\,216$ unterschiedliche Farben darstellbar.

Die meisten Farbdrucker verwenden die Farben **CYAN, MAGENTA, YELLOW**
→ Subtraktive Farbmischung http://www.spectrumcolors.de/cor_cmyk_demo.php



Wie arbeiten Computer

Die Vorstellung, dass **Computer** oder **Roboter** Informationen (über Tastatur eingegeben oder über Sensoren wahrgenommen) verstehen, ist eigentlich falsch. Die Verarbeitung der Daten beruht auf Zahlzeichen, **Zahlensystemen**, Rechengesetzen, die mit rasender Geschwindigkeit im **Mikroprozessor** verarbeitet werden. Die in unserem Alltag verwendeten Schrift- und Zahlensysteme sind aber für die Arbeit mit dem Computer nicht geeignet. Letztendlich müssen alle Informationen für den Computer oder einen Roboter in eine Folge von nur zwei verschiedenen Zahlen (0 und 1) übersetzt werden. Schon **Gottfried Wilhelm Leibnitz** fand im Jahr 1679 erstmals die besondere Eignung des **Zweiersystems** (Binärsystem, Dualsystem) für die maschinelle Verarbeitung von Zahlen. George Boole (1815 - 1864) baute, ausgehend von den logischen Möglichkeiten „**wahr** (true)-**falsch** (false)“, die Grundlagen für die heutige Informatik auf. (**Boolesche Algebra**)



Dualsystem (Binärsystem)



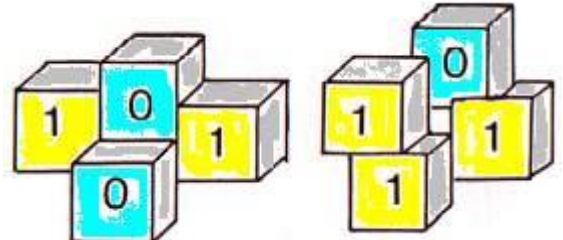
Bit = **binary digit**, kleinste Informationseinheit (bestehend aus 0 oder 1)

Im Computer entspricht ein Bit einem Schaltvorgang: **Strom fließt** oder **Stromkreis unterbrochen**.

oder



Ketten von Bits werden zur Darstellung von Buchstaben und größeren Zahlen benutzt. Mit **8 Bits** = 1 **Byte** können dadurch $2^8 = 256$ Zahlen (0 - 255) dargestellt werden.



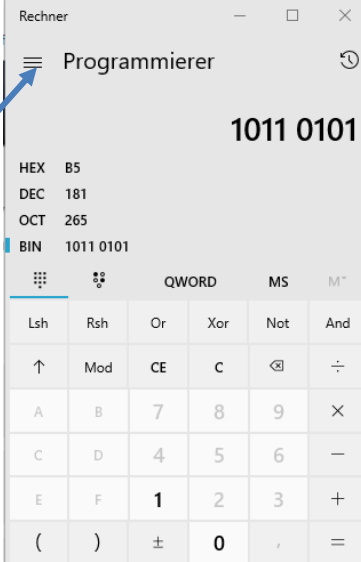
BYTE										
2. Halbbyte				1. Halbbyte						
1	1	1	1	1	1	1	1			
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0			
128	+	64	+	32	+	16				
240				+	15				=	255

Bei der Umrechnung von Binärzahlen in Dezimalzahlen werden die **Stellenwerte** aller mit **1** belegten Stellen addiert!

z.B. **1011 0101** = $2^7+2^5+2^4+2^2+2^0$ = 128 + 32+ 16 +4 + 1 = **181**

Tipp: Schneller geht die Umrechnung zwischen den verschiedenen Zahlensystemen mit dem **Windows-Rechner!**

Hier kannst du zwischen den verschiedenen **Modi des Taschenrechners wechseln!** Für die Umwandlung zwischen den verschiedenen Zahlensystemen muss der Modus „**Programmierer**“ aktiviert werden!





AB6_ Eine binäre Geschichte

AB_5 Eine binäre Geschichte Lösung

- ✓ I was born in year **1770** in Bonn.
- ✓ I became a professional musician at the age of **11**.
- ✓ I lived in Vienna since **1794**.
- ✓ I wrote my first symphony in **1800**.
- ✓ The third symphony „EROICA“ was written in **1804**.
- ✓ When I died in **1827** I was years **56** old.

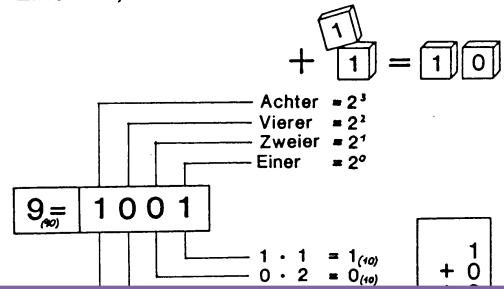
My name is **Ludwig van Beethoven**.

Umrechnung von Dezimalzahlen in Binärzahlen



Zur Umwandlung einer Dezimalzahl in eine **Binärzahl** wird die Dezimalzahl durch die Basis (beim Dualsystem 2) dividiert. Das Ergebnis der Division wird ohne Berücksichtigung des Restes erneut dividiert. Dieser Vorgang wird so lange wiederholt, bis das Ergebnis Null ist. Die **Divisionsreste** (Überlauf) bilden **von rechts nach links angeschrieben** die gesuchte Binärzahl.

Ziffern: 0, 1



Dieser Vorgang, bei dem die Reste der Division ermittelt werden, wird als **Restwertdivision** (**Modulo = mod**) bezeichnet!

$$9 \text{ mod } 2 = 1;$$

$$12 \text{ mod } 7 = 5$$

$$16 \text{ mod } 8 = 0$$

$$72 \text{ mod } 2 = 0$$

Binäre Zahlen werden wie dezimale Zahlen erzeugt. Anstelle von 10 Ziffern (0,1,2...9) gibt es aber nur 2 Ziffern (0,1) Bei jeder Erweiterung um eine Stelle verdoppelt sich der Stellenwert.

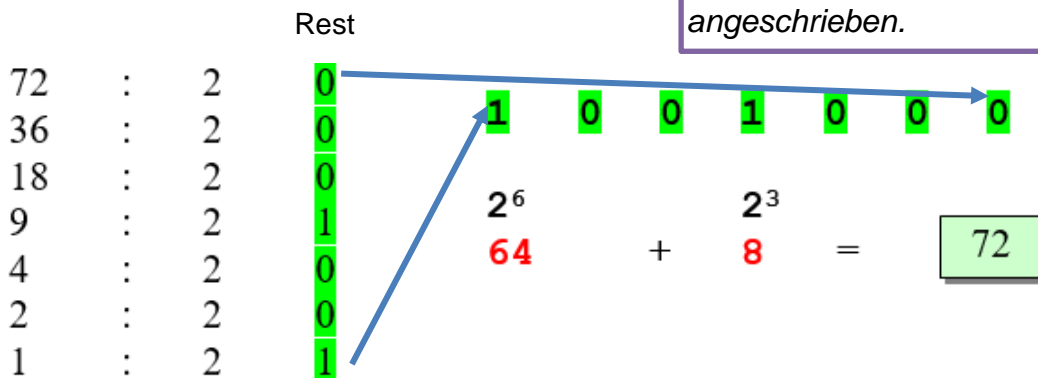


AB7_Dezimalzahl in Binärzahl umwandeln

In den meisten **Programmiersprachen** wird für die **Modulo-Funktion** das Zeichen „%“ verwendet.

Umwandlung der Dezimalzahl 72 in eine Binärzahl

Die einzeln ermittelten Restwerte werden anschließend, beginnend mit dem zuletzt ermittelten Restwert, in einer Reihe angeschrieben.



AB_6 Dezimalzahl in Binärzahl umrechnen Lösung

$$134 \% 2 = 0 \quad 134 = \text{BIN } 10000110 = 2^1 + 2^2 + 2^7 = 2 + 4 + 128 = 134$$

$$67 \% 2 = 1$$

$$33 \% 2 = 1$$

$$16 \% 2 = 0$$

$$8 \% 2 = 0$$

$$4 \% 2 = 0$$

$$2 \% 2 = 0$$

$$1 \% 2 = 1$$

$$229 \% 2 = 1 \quad \text{BIN } 11100101 = 2^0 + 2^2 + 2^5 + 2^6 + 2^7 = 1 + 4 + 32 + 64 + 128 = 229$$

$$114 \% 2 = 0$$

$$57 \% 2 = 1$$

$$28 \% 2 = 0$$

$$14 \% 2 = 0$$

$$7 \% 2 = 1$$

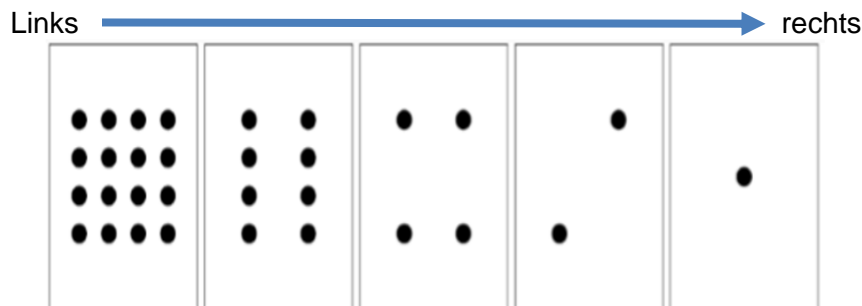
$$3 \% 2 = 1$$

$$1 \% 2 = 1$$

Bevor das Arbeitsblatt „AB7_Binäres zählen“ verteilt wird, ist es sinnvoll sein das Prinzip den SchülerInnen zunächst einmal zu demonstrieren. Dazu werden fünf Karten, welche auf der einen Seite mit Punkten ($2^0=1$, $2^1=2$, $2^2=4$, $2^3=8$, $2^4=16$) bedruckt sind, während sie auf der anderen Seite leer sind. Fünf Kinder werden nach vorne gebeten. Die Karten werden an die Kinder in der folgenden Reihenfolge verteilt.



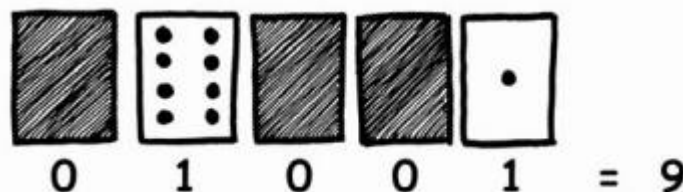
AB8_Binäres zählen



Fragen an die SchülerInnen:

- ✓ Was fällt euch an der Anzahl der Punkte auf den Karten auf?
Jede Karte enthält jeweils doppelt so viele Punkte wie die vorherige Karte
- ✓ Wie viele Punkte müssten also auf den nächsten Karte auf der linken Seite sein, wenn wir noch weitere Karten hinzufügen würden? $2^5=32$, $2^6=64$, $2^7=128$, →8 Karten entsprechen einem 8Bit-System, damit können die Zahlen 0 -255 (256 Zahlen = 2^8) dargestellt werden.

Wir können diese Karten verwenden um Zahlen darzustellen, indem wir gewisse Karten drehen, sodass deren Punkte sichtbar sind (dies schreiben wir als 1), oder wir lassen sie, sodass keine Punkte sichtbar sind (wir schreiben eine 0). Wir zählen die Anzahl der sichtbaren Punkte. Fragen Sie die Kinder, wie man die Zahl 6 darstellen kann (4 Punkte und 2 Punkte), danach 15 (8, 4, 2 und 1 Punkt), danach 21 (16, 4 und 1).



- ✓ Versuchen Sie nun die Klasse von null hochzählen zu lassen. Der Rest der Klasse soll sorgfältig zuschauen und versuchen das Muster zu erkennen.
Jede Karte dreht halb so oft wie die Karte rechts davon.

AB_7 Binäres zählen Lösung

- ✓ Wie stellt man **3 = 00011** **12 = 01100** und **19 = 10011** dar?
- ✓ Gibt es mehrere Möglichkeiten eine Zahl zu bilden? **Nein**
- ✓ Gib die größte **31** bzw. kleinste Zahl **0** an, die du auf diese Weise mit den 5 Karten bilden kannst?
- ✓ Gibt es Zahlen zwischen der größten und der kleinsten Zahl, die man nicht erzeugen kann? **Nein**

Quelle: https://classic.csunplugged.org/wp-content/uploads/2015/11/binary_CSunplugged-german-staub.pdf

Andere Zahlensysteme in der EDV



Beim Binärsystem enthält jedes **Halb-Byte** 16 Kombinationsmöglichkeiten. Die meisten modernen Datenverarbeitungsanlagen verwenden daher heute das **Hexadezimalsystem** (Sedezimalsystem). Dieses System arbeitet mit der Zahl 16 als Basis. Mit diesem System ist es daher möglich, ein Halb-Byte durch ein einziges Zeichen darzustellen.

Dezimal Basis 10	Binär Basis 2	Hexadezimal Basis 16
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10		A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

239

2. Halbbyte				1. Halbbyte			
1	1	1	0	1	1	1	0
14				15			
E				F			

Keine Buchstaben
sondern
**SEDEZIMAL -
Zeichen!**

Die dezimale Zahl **239** heißt im Dualsystem **1110 1111**. Im Hexadezimalsystem reichen für deren Darstellung die beiden Sedezimalzeichen **EF** aus.

Welche dezimale Zahl wird durch **9D** dargestellt?

Hexadezimal	Binär	Dezimal
9D	1001 1101	$2^7+0+0+2^4+2^3+2^2+0+2^0=128+16+8+4+1=157$

AB_9 Hexadezimalsystem Lösung

Hexadezimal	Binär	Dezimal
89	1001 1101	$2^7+0+0+2^4+2^3+2^2+0+2^0=128+16+8+4+1=157$
2A	0010 1010	$0 + 0 + 2^5 + 0 + 2^3 + 0 + 2^1 + 0 = 32+8+2 = 42$
CD	1100 1101	$2^7 + 2^6 + 0 + 0 + 2^3 + 2^2 + 0 + 2^0 = 128 + 64 + 8 + 4 + 1 = 205$

ANALOG – DIGITAL



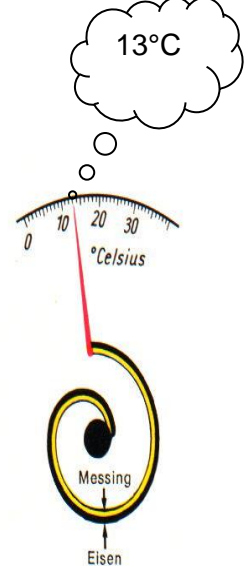
Der Mensch hat seit jeher versucht alles genau zu messen und zu qualifizieren. Nicht alles aber ist so leicht in Zahlen festzuhalten wie z.B. eine Herde Schafe auf einer Weide. Viele Vorgänge in der Natur lassen sich nur unvollkommen in zählbaren Vorgängen ausdrücken. Denken wir z.B. nur an ein Gefäß mit Wasser, das erhitzt wird. Der Temperaturanstieg ist ein kontinuierlich verlaufender Vorgang der durch Zahlen

(Tabelle, Thermometer mit Digitalanzeige) nur ausschnittsweise wiedergegeben werden kann. Verwendet man zur Darstellung ein Flüssigkeitsthermometer oder ein



Bimetall Thermometer, so kann man am Anstieg der Flüssigkeitssäule bzw. an der Krümmung des Bimetall Streifens den kontinuierlichen Anstieg, analog zum tatsächlichen Anstieg der Temperatur, genau mitverfolgen. Versucht man jedoch den Wert auf der Skala abzulesen

(13°C), so erhält man nur einen, für einen Moment gültigen digitalisierten Näherungswert. Die Genauigkeit des abgelesenen Wertes hängt von der Genauigkeit der Skala ab. Betrachtet man die Skala mit einer Lupe, könnte man feststellen, dass der Wert etwa bei 13,2° C liegt. Je feiner unterteilt die Skala ist, desto genauer wäre der abzulesende Wert. Trotzdem stellt die bestimmte Größe immer nur einen Näherungswert dar.

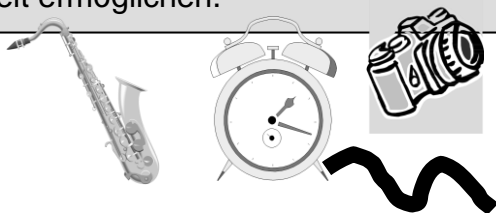


Mit zunehmender Erwärmung biegt sich der **Bimetallstreifen** immer weiter. Der Vorgang verläuft kontinuierlich, **analog** zum tatsächlichen Temperaturanstieg.

Analog



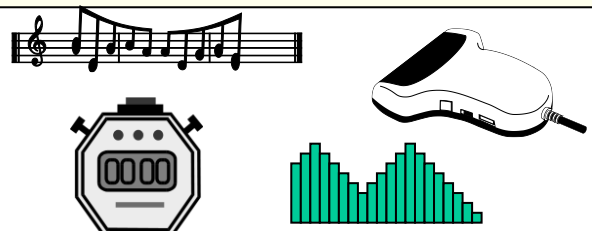
Unendliche Menge von nicht immer eindeutig unterscheidbaren Zeichen, die eine **wirklichkeitsgetreue Darstellung** der Umwelt ermöglichen.



Digital



Endliche Menge eindeutig unterscheidbarer Zeichen, die uns eine **Momentaufnahme** eines Zustandes wiedergeben.



Wir Menschen treffen oft sehr unklare Aussagen: „ein bisschen, etwas, ein wenig ...“ Herkömmliche elektronische Schaltungen können mit solchen analogen Entscheidungen nichts anfangen.

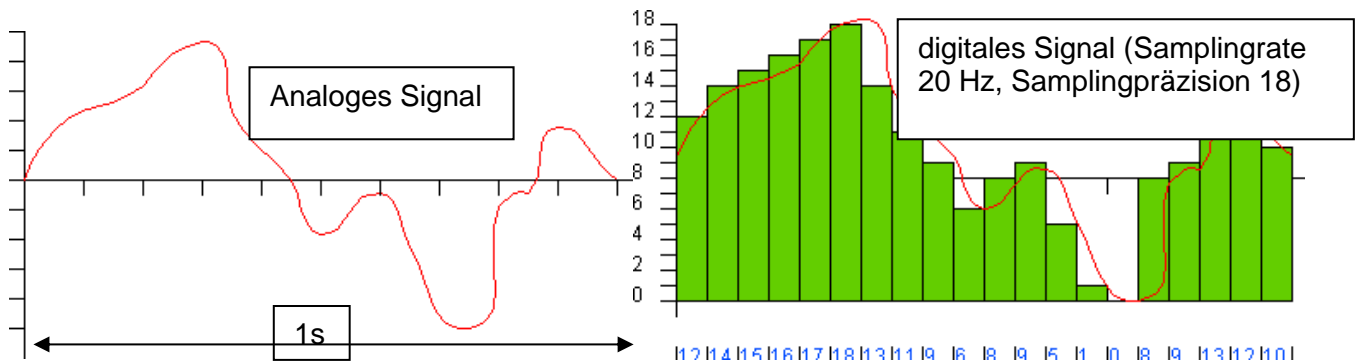
Die meisten Wahrnehmungen, die der Mensch mit seinen Sinnen (Auge, Ohr, Nase ...) aufnimmt, sind solche analogen Informationen. Digitale Informationen dagegen bedienen sich eines endlichen, abzählbaren Zeichenvorrates (Ziffern, Buchstaben, ASCII-Tabelle ...)

Zur Umwandlung von digitalen Signalen in analoge Signale und umgekehrt werden **Digital-Analog Umsetzer** (DA-Wandler) verwendet.

Umwandlung analoger Signale



Analoge Signale müssen zur Weiterverarbeitung im Computer digitalisiert werden! Der **Analog-Digital-Wandler** (auch AD-Wandler) tastet nun nach und nach die Welle ab und merkt sich für jeden Moment eine Zahl, die angibt, wie hoch der Wellenberg ist. Die verschiedenen AD-Wandler unterscheiden sich dabei durch ihre sogenannte **Samplingrate**, also wie oft pro Sekunde sie abtasten, und die **Samplingpräzision**, also wie genau eine Zahl ermittelt wird.



Beispiel für DA- und AD-Wandlung



- ✓ Ein **Scanner** verwandelt die analogen Bildinformationen einer Fotografie in ein digitales Muster für den PC.
- ✓ Ein **Modem** verwandelt die Binärdaten eines PCs in analoge Signale zur Übertragung im herkömmlichen Telefonnetz
- ✓ Mit einer **Soundkarte** können analoge Aufnahmen von Kassetten und Schallplatten digitalisiert und auf CDs gebrannt werden.
- ✓ Die **Sensoren** in einem Roboter wandeln die empfangenen Signale in elektrische Impulse unterschiedlicher Stärke um.

Fuzzy Logic



Es gibt heute aber bereits schon Computersysteme, die mit Fuzzy Logic arbeiten. Durch die Kombination der Fuzzy Logic und neuronaler Netze sind beispielsweise lernfähige Systeme möglich, die mit „unpräzisen“ Informationen arbeiten könnten. Die Möglichkeit, dass eine Variable einen bestimmten Wert annimmt, wird mit einem Wahrheitsgrad (Zugehörigkeitswert) zwischen 1 (wahr) und 0 (falsch) angegeben. In der **Fuzzy Logic** wird das Ergebnis einer Operation mit einem gewissen Wahrscheinlichkeitswert errechnet. Neben den herkömmlichen Werten „wahr“ oder „falsch“ kann ein Ergebnis beispielsweise noch die Werte „wahrscheinlich wahr“, „möglicherweise wahr“, „möglicherweise falsch“ und „wahrscheinlich falsch“ annehmen.

Fuzzy bedeutet so viel wie „undeutliche Logik“, eine Form der Logik, die in einigen Expertensystemen und anderen Anwendungen der Künstlichen Intelligenz (KI) eingesetzt wird.

Programmierung

Unter Programmieren versteht man die Festlegung einer Arbeitsvorschrift (**Programm**), die eine automatische Verarbeitung durch den Computer ermöglicht.

Ein **Programm** ist eine **zeitliche und logische Abfolge von Anweisungen**. Wir begegnen solchen Abläufen fast überall in unserem Leben: zB Fernsehprogramm, Waschprogramm, Bedienungsanleitungen, Bastelanleitungen ...



AB10_ Was ist ein Programm

In der Computerwissenschaft ist eine **Programmiersprache** eine künstliche Sprache zur Aufstellung von Befehlsfolgen, die schließlich von einem Mikroprozessor abgearbeitet werden können.

Verschiedene Arten von Programmiersprachen

Eine Programmiersprache kann entweder der Arbeitsweise eines Prozessors angepasst sein → **maschinenorientierte Expertensprache** (zB ASSEMBLER)



Ein **Assembler** ist ein Programm, das in **Assemblersprache** geschriebene Anweisungen in **Maschinsprache** übersetzt!

Code in Assemblersprache:

342B:0100	B84000	MOV	AX,0040
342B:0103	8ED8	MOV	DS,AX
342B:0105	B83412	MOV	AX,1234
342B:0108	A37200	MOV	[00721],AX
342B:010B	EA0000FFFF	JMP	FFFF:0000

Dasselbe Programm in Maschinsprache (Binärschreibweise):

```
10111000 01000000 00000000 10001110 11011000 10111000 00110100 00010010
10100011 01110010 00000000 11101010 00000000 00000000 11111111 11111111
```

oder sich an dem zu lösenden Problem orientieren

→ **anwenderorientierte Problemsprache** (zB Python, C#, C, C++, Visual Basic, JavaScript, PHP, ..)

```
// ConsoleOutput.cs
class ConsoleOutput
{
    static void Main()
    {
        System.Console.WriteLine();
        System.Console.WriteLine("Hello world!");
    }
}
```

Zusätzlich gibt es noch eine Gruppe von hochentwickelten Spezialsprachen, die einem besonderen Anwendungsbereich z.B. Datenbankabfragen (**SQL**) dienen.

```
SELECT COUNT(CustomerID), Country FROM Customers GROUP BY Country;
```

Grundelemente einer modernen Programmiersprache

Die wesentlichen Elemente jeder modernen Programmiersprache sind:

- ✓ **Funktionen/Methoden**
- ✓ **Variablen/Objekte**
- ✓ **Befehle/Anweisungen**
- ✓ **Operatoren** (+, -, /, *, <, > ...)
- ✓ **Entscheidungen** und **Schleifen** (ITERATIONEN)
- ✓ **Kommentare** (dienen lediglich der Beschreibung, werden nicht vom Prozessor ausgeführt!)

Arten der Programmierung

- ✓ **Strukturierte Programmierung**
Zerlegungen von Programmen in Unterprogramme (Teilprogramme, Funktionen, Methoden). Die einzelnen Anweisungen im Hauptprogramm und in den einzelnen Teilprogrammen werden der Reihe nach (sequentiell) abgearbeitet. Durch Entscheidungen (IF ... THEN ... ELSE) und Schleifen (ITERATIONEN), kann die Programmausführung gestoppt oder gewisse Abläufe wiederholt werden.
- ✓ **Objektorientierte Programmierung (OOP)**
Moderne Betriebssysteme bzw. graphische Benutzeroberflächen wie Windows sind wahre Programmmonster und bestehen meist aus vielen Millionen Zeilen kompliziertester Quellcodes. Windows unterstützt zum Beispiel die unterschiedlichsten Hardwaresysteme, Drucker, Sprachen und Anwendungsgebiete. Normalerweise müsste ein Entwickler all diese verschiedenen Möglichkeiten bei der Programmierung berücksichtigen. Der Aufwand wäre so groß, dass niemand Programme entwickeln würde. Das Zauberwort, um die Programmierung unter Windows zu vereinfachen heißt **OOP** (Objektorientierte Programmierung). Kernpunkt dieses Konzepts ist das **Black-Box-Prinzip**. Das bedeutet, dass der Programmierer Funktionen von Windows nutzen kann, ohne zu wissen, wie sie eigentlich funktionieren.

Die Aufgaben eines Programmierers unter Windows beschränken sich dabei auf drei Grundgebiete:

Eigenschaften	vergeben und ablesen
Ereignisse	abfangen und darauf reagieren
Methoden	auf Objekte anwenden

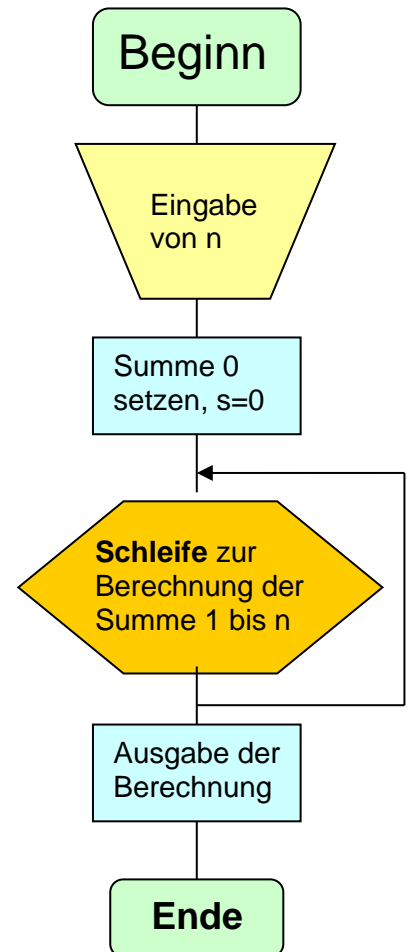
Objekte:

Objekte sind sichtbare oder unsichtbare Elemente eines Programmes (Programmfenster, Schaltflächen, Textfelder, Bildfelder ...), die bestimmte Eigenschaften aufweisen.

Eigenschaften

Eigenschaften sind einstellbare Werte, die das Erscheinungsbild oder die Funktionsweise von Objekten festlegen. Je nach Art und Aufgabe haben verschiedene Objekte auch unterschiedliche Eigenschaften (Schriftart, Schriftgröße, Hintergrundfarbe, Text, Ausrichtung...). Die Eigenschaften der Objekte werden vom Programmierer entweder beim Programmwurf oder während der Laufzeit des Programms, als Folge eines Ereignisses, verändert.

Programmablaufplan (PAP)



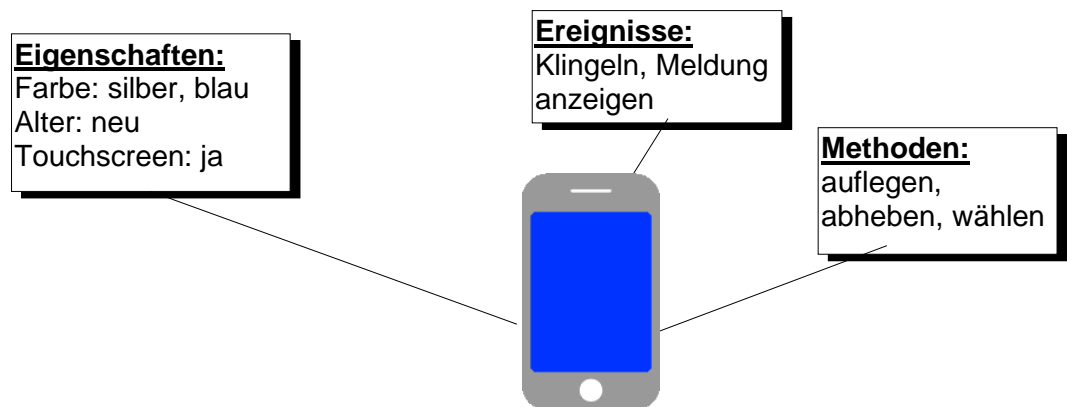
Ereignisse

Ein Ereignis ist eine Aktion, die von einem Objekt erkannt wird (zB Klicken auf eine Schaltfläche, Eingabe eines Textes in ein Textfeld,...). Aufgabe des Programmiers ist es nun festzulegen, wie auf ein bestimmtes Ereignis reagiert werden soll. Bei der prozessorientierten Programmierung (POP) gibt es keinen durchgehenden Programmcode mit definierten Start- und Endpunkten.

Methoden:

Methoden sind Aktionen, die sich auf ein Objekt beziehen (Löschen eines Textes oder einer Grafik, Drucken, Aktualisieren eines Bildschirmes, Verschieben eines Objektes).

Ein Handy oder Smartphone stellt für die meisten Menschen eine solche Blackbox dar. Jeder von uns benutzt es, die wenigsten aber wissen, wie es wirklich funktioniert. Eine solche Blackbox, deren Innenleben man nicht kennt, nennen wir **Objekt**. Jedes Objekt besitzt bestimmte **Eigenschaften** (Properties) und löst Ereignisse aus, auf die man durch geeignete Methoden reagieren kann.



- ✓ **Prozessorientierte Programmierung (Ereignisorientierte Programmierung) POP**
Bei der ereignisorientierten Programmierung wird nicht mehr eine Liste von Befehlen nach einem festen Algorithmus abgearbeitet. Stattdessen wird eine Sammlung von Prozeduren und Funktionen verwendet. Der Aufruf einzelner Funktionen wird durch bestimmte Ereignisse (Tastatureingaben, Mausklicks, Berührungen am Touchscreen, Rückmeldung von Sensoren) ausgelöst! Solche Programme verfügen also nicht über ein geplantes, festes Ablaufschema, sondern der Ablauf wird mit der Benutzung durch die Anwender oder durch Sensoren, die auf die Umgebung reagieren, neu bestimmt.

Typische Programmierfehler

- ✓ **Syntaxfehler:** Eingegebene Befehle werden vom Computer nicht erkannt.
- ✓ **Bedienungsfehler:** zB, Drucker nicht eingeschaltet, Sensoren falsch angeschlossen, ..
- ✓ **Logische Fehler:** Unlogischer Programmaufbau führt oft zu einem völlig unsinnigen Programmablauf.

Phasen der Programmentwicklung

- ✓ Genaue Beschreibung der **Aufgabenstellung**
- ✓ **Problemanalyse**
- ✓ **Programmstrukturierung** (Welche Daten sind erforderlich, erforderliche Variablen und Konstante, Algorithmen aufstellen, ...)
- ✓ **Codierung** (Quelltext erstellen)
- ✓ **Programmtest** (Prüfung auf Syntaxfehler, logische Fehler, Testläufe um Fehler zu erkennen)
- ✓ **Programmdokumentation** (Programmbeschreibung, Kommentare zu Variablen und Konstanten ...)

Visuelle Programmiersprachen (VPL)

Da die textorientierte Programmierung professioneller Programmiersprachen für die meisten SchülerInnen abschreckend wirkt verwenden wir für die Programmierung der Roboter visuelle Programmiersprachen (VPL), die uns das Erlernen von der grundlegenden Funktionen moderner Programmiersprachen erleichtern.

Visuelle Programmiersprachen benutzen vorgefertigte Programmelemente, meist in Form von Blöcken, die per Drag&Drop wie Puzzleteile zu einem Programm zusammengefügt werden können.

Die Blöcke sind so gestaltet, dass sich nur auch vom logischen Ablauf passende teile aneinanderfügen lassen. Meist erleichtert eine farbliche Kennzeichnung der einzelnen Blöcke die Orientierung bei der Auswahl der richtigen „Puzzleteile“

Zwei der am häufigsten verwendeten blockorientierten, visuellen Lernsprachen sind **Scratch** und **Blockly**. Vom Prinzip her sind VPLs Programmgeneratoren die aus den ausgewählten und zusammengefügt Programmblöcken einen Quellcode in einer professionellen Programmiersprache (Java Skript, Python, ...) generieren.

Diese per **Drag&Drop** in **Scratch** zusammengesetzten Programmblöcke generieren im Hintergrund den nebenstehend Quellcode in **Java-Skript**.



```
//When play button pressed
setAllLedsColor("green");
pen("DOWN");
var cmf = 4;
for(var mvz = 0; mvz < cmf; mvz++)
{
  forwardForDistance("02", "10");
  spinLeftByDegrees("02", "90");
}
pen("UP");
setLedColor("00", "red");
playSound("3");
```



AB11_Visuelle Programmierung (VPL)

Scratch ist eine visuelle Programmiersprache, deren Ziel es ist, bereits Kinder mit den Prinzipien der Programmierung vertraut zu machen.



(Bildquelle: Flickr CC BY 2.0, Wesley Fryer)

„**Blockly** ist eine Bibliothek, die Webanwendungen, Android- und iOS-Apps einen visuellen Code-Editor hinzufügt. Der Blockly-Editor verwendet verzahnte, grafische Blöcke zur Darstellung von Programmierkonzepten wie Variablen, logischen Ausdrücken, Schleifen und mehr. Es ermöglicht den Benutzern, Programmierprinzipien anzuwenden, ohne sich Gedanken über die Syntax machen zu müssen. Die abgebildeten Algorithmen können u. a. als **JavaScript**- oder **Python-Code** exportiert werden“.

<https://de.wikipedia.org/wiki/Blockly>



AB12_Prüfe dein Wissen AB13_Das Magisches Robotik Quadrat



- (1) Welche Bauteile eines Roboters nehmen Informationen aus der Umwelt war?
- Bluetooth-Schnittstelle
 - Motoren
 - Sensoren
 - LEDs
- (2) Welcher Bauteil eines Roboters steuert die Motoren und Sensoren?
- USB-Interface
 - Mikroprozessor
 - Aktoren
 - Bluetooth Schnittstelle
- (3) Welche Bauteile in einem Roboter wandeln empfangene Signale in elektrische Impulse um?
- Aktoren
 - Sensoren
 - Motoren
 - Rezeptoren
- (4) Welches der folgenden Zahlensysteme wird nicht für die Verarbeitung der Daten im Computer verwendet?
- Dualsystem
 - Hexadezimalsystem
 - Dezimalsystem
 - Binärsystem
- (5) Das Byte:
- 1 Byte = 8 Bit.
 - 1 Byte = die kleinste Informationseinheit.
 - 1 Byte entspricht der Größe eines Gebissabdrucks in einem frischen Hamburger.
 - Es gibt 256 von ihnen.
- (6) Welche der folgenden Aussagen stimmen? (mehrere Antworten sind richtig)
- Ein Bit sind 8 Byte
 - Ein Byte ist eine Einheit zur Angabe der Größe eines Datenträgers
 - Ein Bit ist die kleinste Informationseinheit
 - Ein Bit ist der Speicherplatz für eines der beiden Zeichen: 0 oder 1
- (7) Welcher der folgenden Begriffe ist keine Programmiersprache?
- Blockly
 - Scratch
 - Android
 - Python
- (8) Gottfried Wilhelm Leibnitz erkannte bereits 1679 die besondere Eignung des Dualsystems für die maschinelle Verarbeitung! true false
- (9) Die Boolesche Algebra bildet die Grundlage der heutigen Informatik! true false
- (10) Bluetooth und WLAN sind Schnittstellen zur drahtlosen Datenübermittlung! true false
- (11) Ein Bit besteht aus zwei Halbbytes! true false
- (12) Mit einem Byte können 256 Zahlen dargestellt werden! true false
- (13) Roboter arbeiten nach dem EVA-Prinzip! true false
- (14) Wie heißt in der Programmierung eine Kontrollstruktur, die das mehrfache Notieren von Befehlen erspart! **Schleife (ITERATION)**

AB_12 Das magische Robotik Quadrat

Trage jeweils die Zahl der richtigen Antwort in das Kästchen bei der Frage ein!

(1) Fuzzy Logic, (2) EVA-Prinzip (3) LEDs (4) Sedezimalzeichen (5) Hexadezimalsystem (6) George Boole (7) Mikroprozessor (8) Künstliche Intelligenz (9) Blue tooth, (10) Sensoren (11) AD-Wandler (12) Restwertdivision (modulo) (13) binary digit (14) Binärsystem (15) Byte (16) USB-Interface

<p>16 Schnittstelle</p> <p>für die Kommunikation mit einem Roboter?</p>	<p>3 Welcher</p> <p>Bauteil eines Roboters kann sowohl Sensor und Aktor sein?</p>	<p>2 Nach</p> <p>welchem Prinzip arbeiten Roboter und Computer?</p>	<p>13 Was</p> <p>bedeutet die Abkürzung „bit“?</p>
<p>5 Welches</p> <p>Zahlensystem findet in modernen Rechenanlagen vor allem Verwendung?</p>	<p>10 Welche</p> <p>Bauteile eines Roboters wandeln empfangene Signale in elektrische Impulse um?</p>	<p>11 Wie heißt</p> <p>ein elektronischer Bauteil, der analoge Signale in digitale Informationen umwandelt?</p>	<p>8 Wofür steht</p> <p>die Abkürzung „KI“</p>
<p>9 Schnittstelle</p> <p>zur drahtlosen Datenübertragung.</p>	<p>6 Wer baute</p> <p>ausgehende von den beiden logischen Möglichkeiten „true“ und „false“ die Grundlagen der heutigen Informatik auf?</p>	<p>7 Bauteil</p> <p>eines Roboters zur Steuerung der Aktoren und Sensoren.</p>	<p>12 Wofür wird in</p> <p>den meisten Programmiersprachen das Zeichen „%“ verwendet?</p>
<p>4 Wie heißen</p> <p>die Buchstaben A-F, die im Hexadezimalsystem zum Darstellen von Zahlen verwendet werden.</p>	<p>15 Wie heißt eine</p> <p>Kette aus 8 Bits, die zur Darstellung von Zahlen und Buchstaben verwendet wird?</p>	<p>14 Andere</p> <p>Bezeichnung für Dualsystem.</p>	<p>1 Wie heißt die</p> <p>Logik, die in der künstlichen Intelligenz und bei Expertensystemen verwendet wird?.</p>

Antworten richtig eingetragen, so ergeben die Summen der Spalten, Zeilen und der Diagonalen jeweils die Zahl 34!