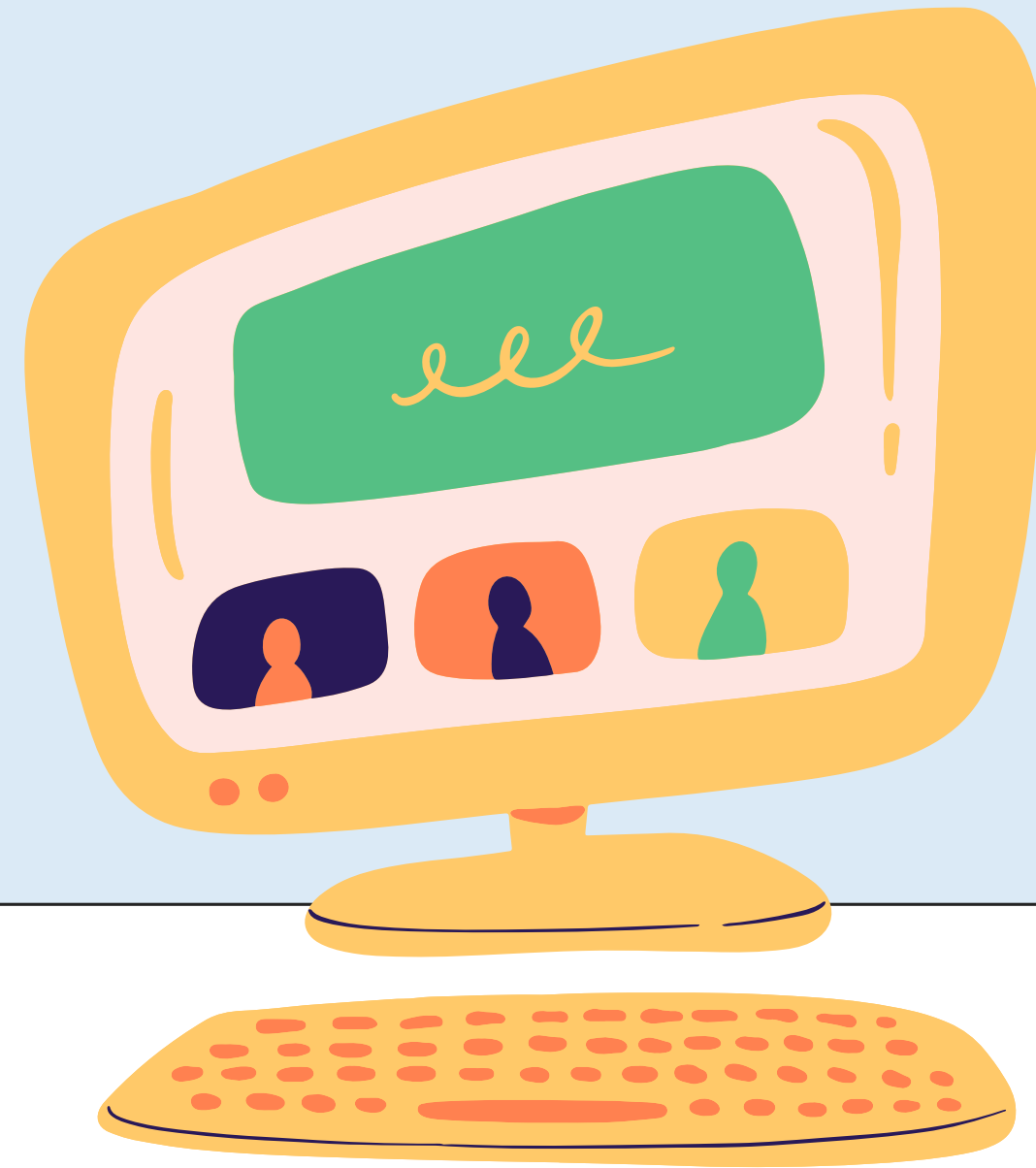


INFORMATIK FÜR KINDER



Sitzung 1

Darmeleon



Unser Team



Bojana



Lina



Elena

ABLAUFPPLAN

Sitzung 1

Wiederholung

Logik

Binär

Computer

Programmiersprache

Sitzung 2

Scratch

Befehle dem Rechner
übergeben

Programmiersprachen

Python

Sitzung 3

Sitzung 4

Themen



Logik

Übungen zum Aufwärmen

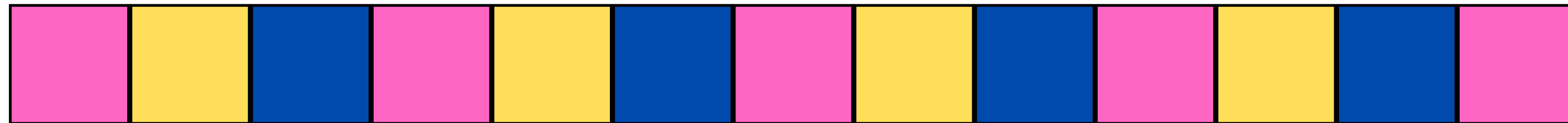
Einführung in Computerwelt

Was ist ein Computer, aus welchen Teilen besteht er, wie funktioniert er? Was ist Hardware, und was ist Software?

Programmiersprache

Einführung in Programme, Programmiersprachen

LOGIK - FARBENFOLGEN



LOGIK - ZAHLENFOLGEN

1	3	5	7	9	11	13	15	17	19	21	23	25
---	---	---	---	---	----	----	----	----	----	----	----	----

0	4	8	12	16	20	24	28	32	36	40	44	48
---	---	---	----	----	----	----	----	----	----	----	----	----

$12+4$

$11 \ 10 \ 13 \ 17$

$65*2$

2	3	5	9	17	33	65	129
---	---	---	---	----	----	----	-----

$2+1 \quad 3+2 \quad 5+4$

$(5*2)-1 \quad (9*2)-1 \quad (17*2)-1 \quad 33*2$

EINFÜHRUNG IN COMPUTERWELT

WIE DENKEN DIE COMPUTER? DAS BINÄRSYSTEM

MENSCHEN-
DENKWEISE



5 SINNE

COMPUTER-
DENKWEISE

1010
0101
1001

0,1 = 2 Ziffer,
2er System

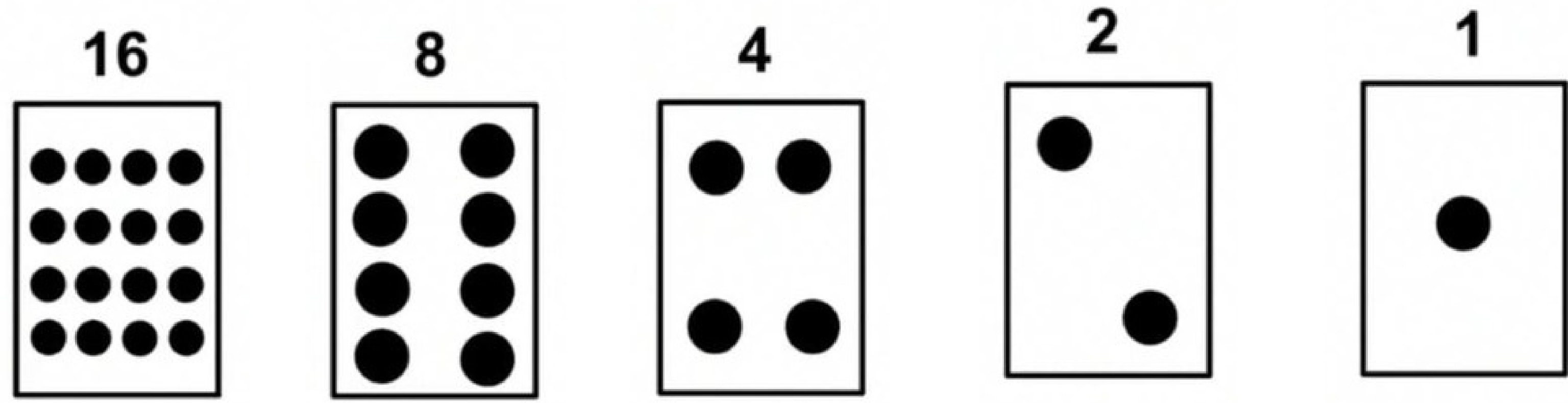
23 23420248329

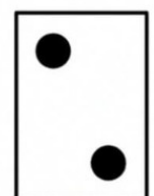
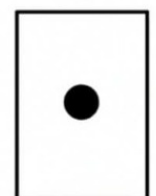
0,1,2,3,4,5,6,7,8,9

10 Ziffer, 10er System

EINFÜHRUNG IN COMPUTERWELT

DAS BINÄRSYSTEM

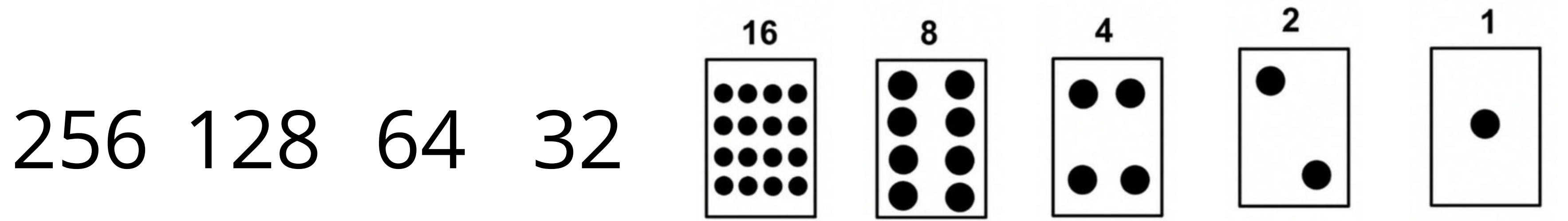


3 =  

EINFÜHRUNG IN COMPUTERWELT

DAS BINÄRSYSTEM

$$22 = 000010110$$



$$22 - 16 = 6 - 4 = 2 - 2 = 0$$

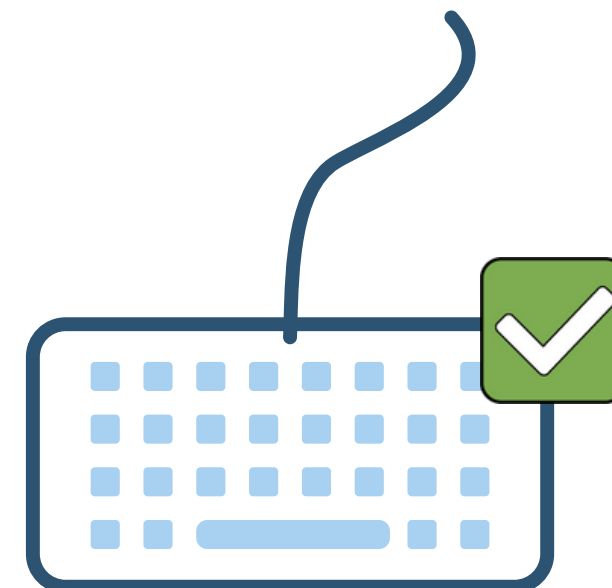
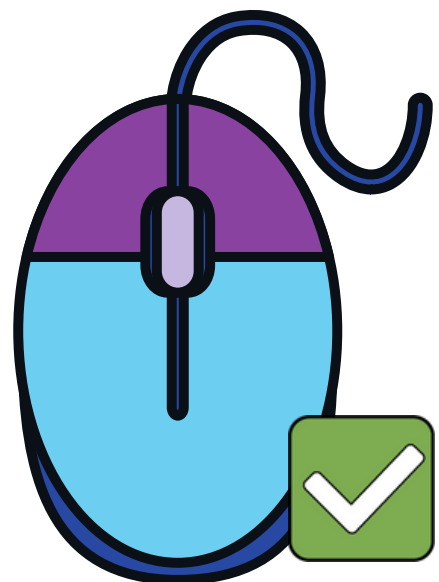
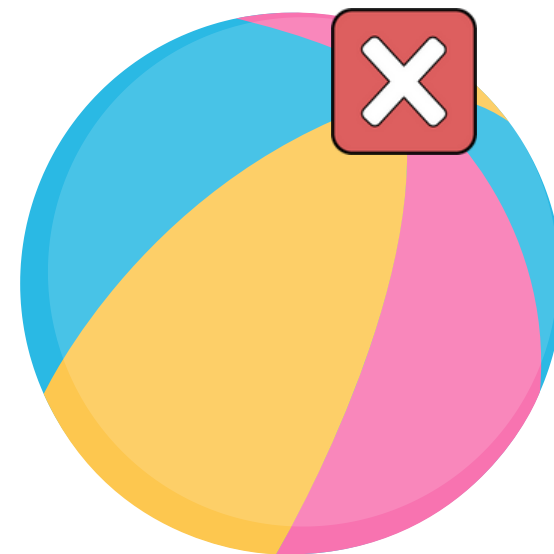
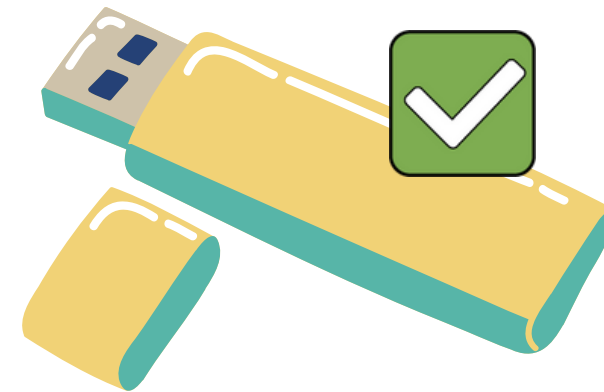
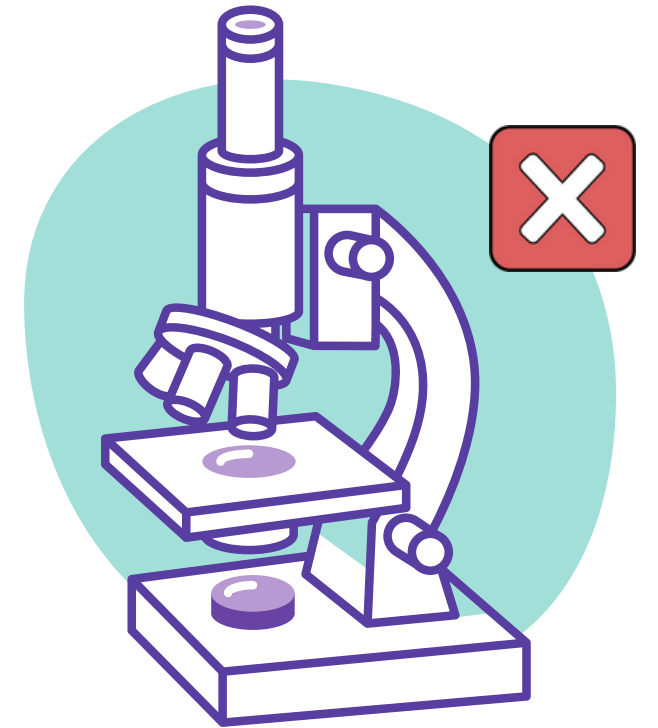
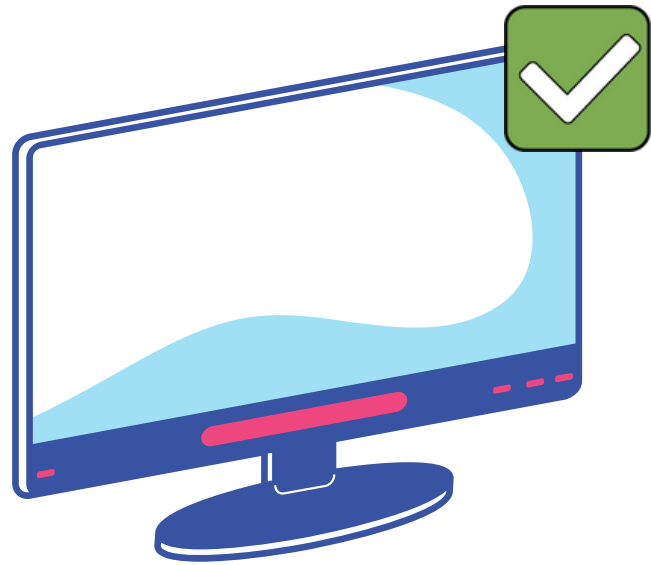
EINFÜHRUNG IN COMPUTERWELT

BINÄRSYSTEM - ÜBUNG - WELCHE ZAHL IST DAS?

		16	8	4	2	1
9	=	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
12	=	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8	=	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
16	=	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

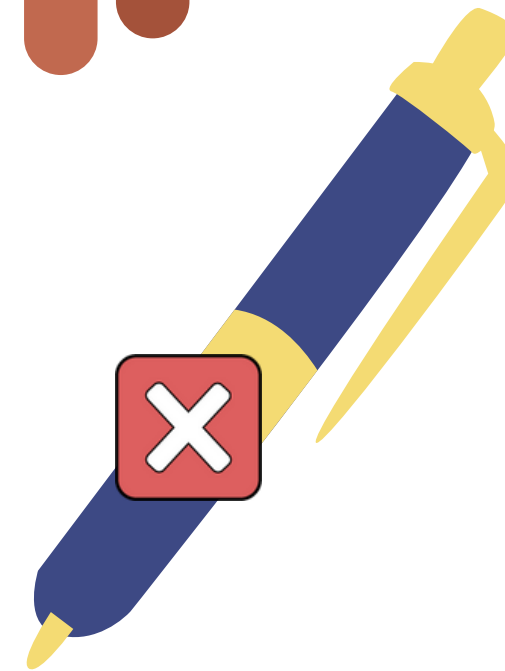
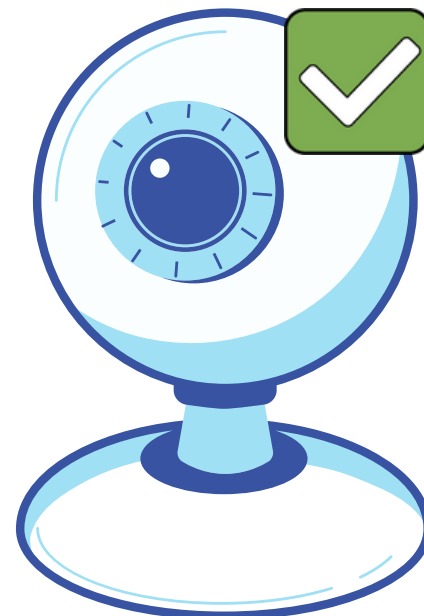
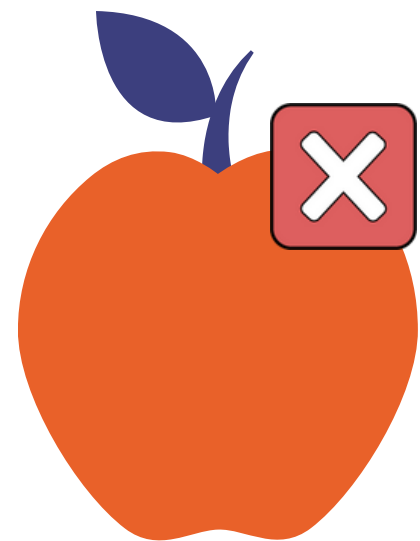
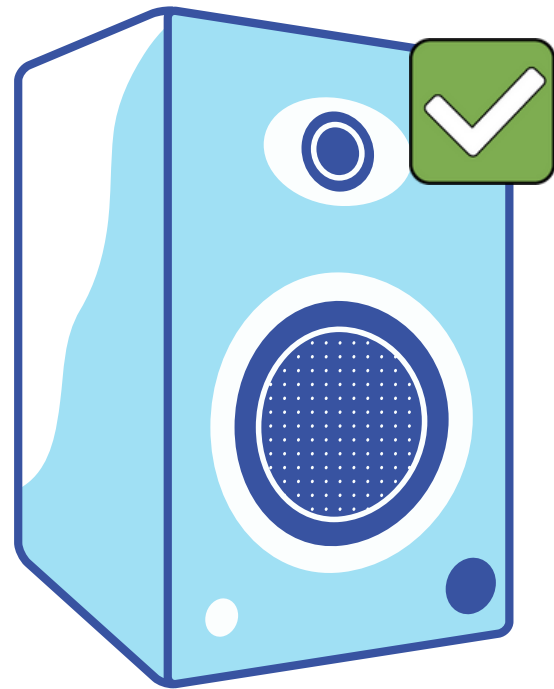
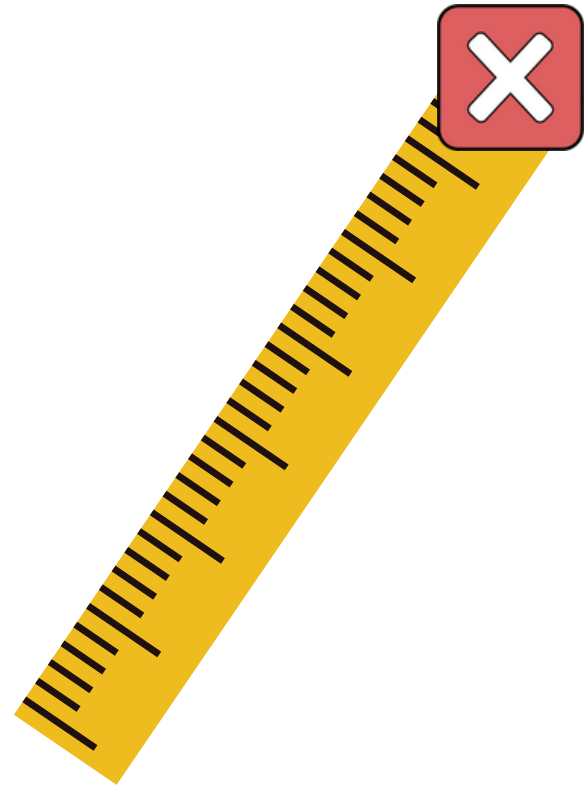
EINFÜHRUNG IN COMPUTERWELT

WAS GEHÖRT ZUM COMPUTER?



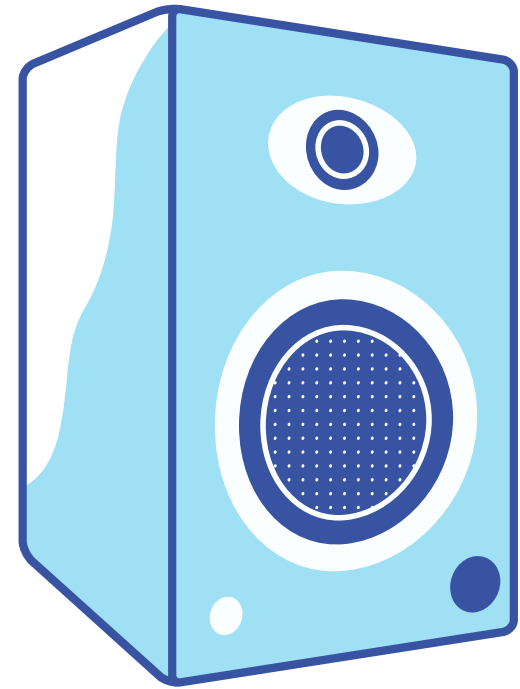
EINFÜHRUNG IN COMPUTERWELT

WAS GEHÖRT ZUM COMPUTER?



EINFÜHRUNG IN COMPUTERWELT

HARDWARE VS. SOFTWARE

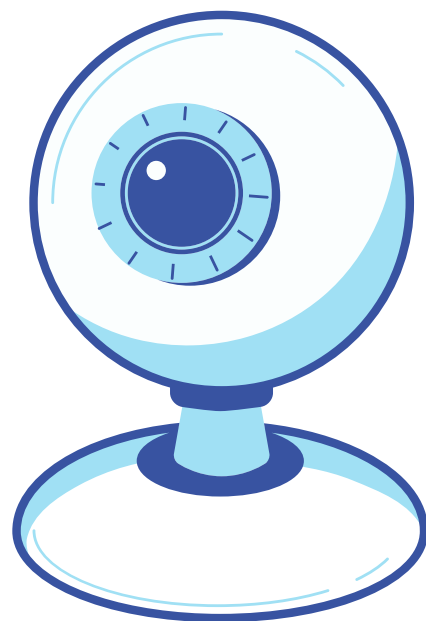
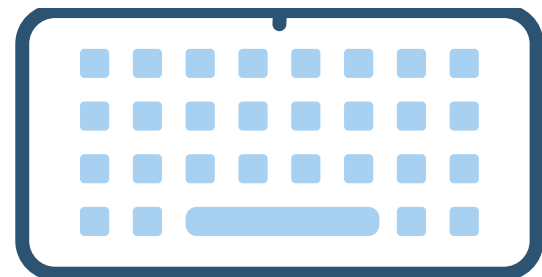


"schwer"
festes
berühren
anfassen

nicht berühren
nicht anfassen



zoom

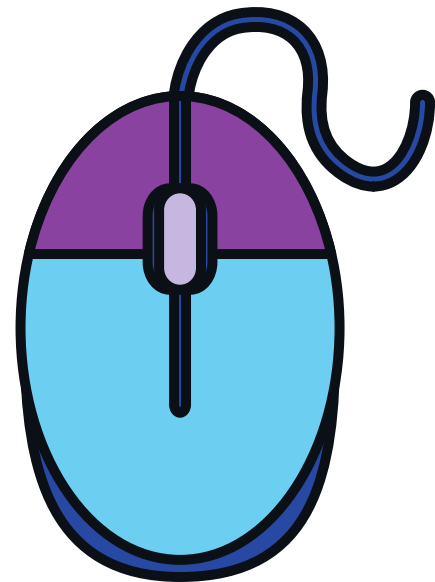
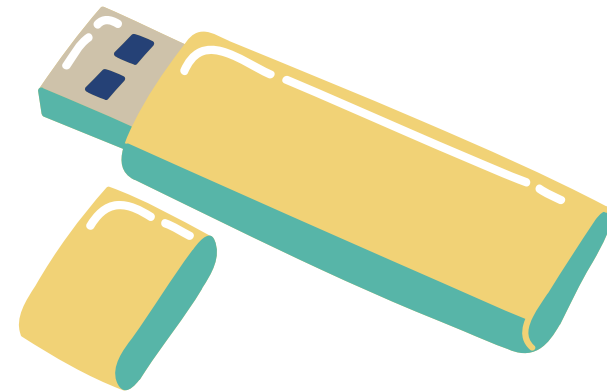


SCRATCH

MINECRAFT

EINFÜHRUNG IN COMPUTERWELT

ERKENNE - WAS IST SOFTWARE, WAS IST HARDWARE?



HARDWARE

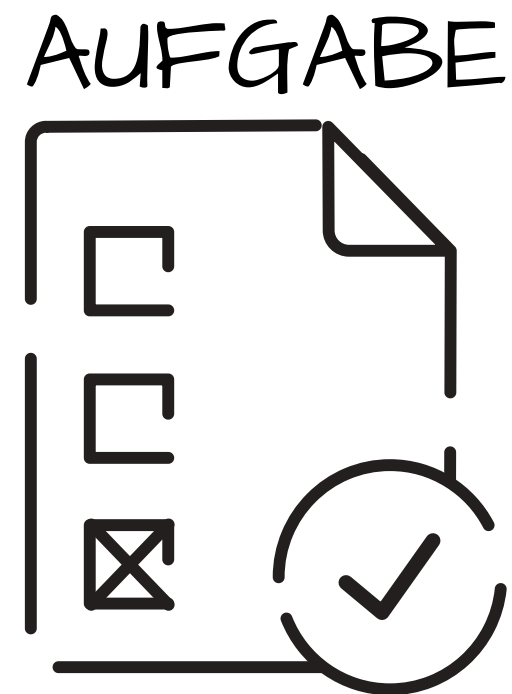
FORTNITE



SOFTWARE




















PROGRAMMIEREN

EINFÜHRUNG
WAS HEISST PROGRAMMIEREN?



PROGRAMMIERSPRACHEN*

EINFÜHRUNG

Front-end web development	Back-end web development	Mobile development
 JavaScript	 JavaScript	 Swift
 Elm	 Scala	 Java
 TypeScript	 Python	 Objective C
	 Go	 JavaScript
	 Ruby	
Game development	Desktop applications	Systems programming
 Unity	 Scala	 Go
 TypeScript	 Go	 Rust
	 Python	

PROGRAMMIERSPRACHEN*

EINFÜHRUNG

High-level program

```
class Triangle {  
    ...  
    float surface()  
        return b*h/2;  
}
```

Low-level program

```
LOAD r1,b  
LOAD r2,h  
MUL r1,r2  
DIV r1,#2  
RET
```

Executable Machine code

```
0001001001000101  
0010010011101100  
10101101001...
```

PROGRAMMIERSPRACHEN

WELCHEN CODE ERKENNT DER COMPUTER ALLEINE?  

100101
101010
100101



```
def sort(list1):  
    if not list1:  
        return []  
    maxIndex = len(list1)-1  
    for i in range(maxIndex):  
        for j in range(maxIndex,i,-1):  
            if list1[j-1] > list1[j]:  
                holder=list1[j-1]  
                list1[j-1] = list1[j]  
                list1[j] = holder  
    return list1  
  
def assertEquals(a1,a2):  
    if a1 != a2:  
        raise Exception()
```



00a010
001h10
00P011



00000
00000
00000



```
#include <string>  
#include <iostream>  
#include <GAT++.hpp>  
  
GAT::Result  
RemoteFile::GetFile (GAT::Context context,  
                    std::string source_url,  
                    std::string target_url)  
{  
    try  
    {  
        GAT::File file (context, source_url);  
        file.Copy (target_url);  
    }  
    catch (GAT::Exception const &e)  
    {  
        std::cerr << "Some_error:_" << e.what() << std::endl;  
        return e.Result();  
    }  
    return GAT_SUCCESS;  
}
```



11111
11111
11111



```
01 .MODEL SMALL  
02 .STACK 100H  
03 .CODE  
04  
05 MOV AX, 0x3C  
06 MOV BX, 00000000000001010B  
07 ADD AX, BX  
08 MOV BX, 14  
09 SUB AX, BX  
10  
11 MOV AH, 04CH  
12 INT 21H
```



001010
002010
002011



HAUSAUFGABE

SCRATCH

bitte herunterladen

<https://scratch.mit.edu/download>

FRAGENRUNDE

