

Herzlich Willkommen bei Inf-Einf-B Thema 1.

Heute viel Input. Wenn es zu schnell geht, in Ruhe Shorts und Lecture Notes anschauen.

Kurs-Website mit Wochenplan, Videos, Notizen, Aufgaben, FAQ, uvm.: [inf.zone](#)

Die Vorlesungen werden aufgezeichnet und im Internet veröffentlicht. Die Videos werden allerdings meist erst 3 Wochen später online sein; Folien und Notizen am Tag nach der Vorlesung. Machen Sie sich Notizen!

Ihre Fragen sind in der Aufzeichnung in der Regel nicht zu hören (ich wiederhole sie).

Seien Sie furchtlos und stellen Sie Fragen!
“Publicly not knowing is valuable!”

Nutzen Sie die Zeit, bis es los geht! Stellen Sie sich mindestens zwei anderen Personen in Ihrer Nähe vor, die Sie noch nicht kennen! (Um das Eis zu brechen, sagen Sie einfach, dass Sie nur diese Anweisung befolgen.)

Wir machen heute eine 30 min Pause.

Es gibt Kekse.

Dies ist Inf-Einf-B.

lernen zu programmieren in Scratch

when  clicked

go to x: y:



set pen size to



erase all

repeat

turn  degrees



pen down

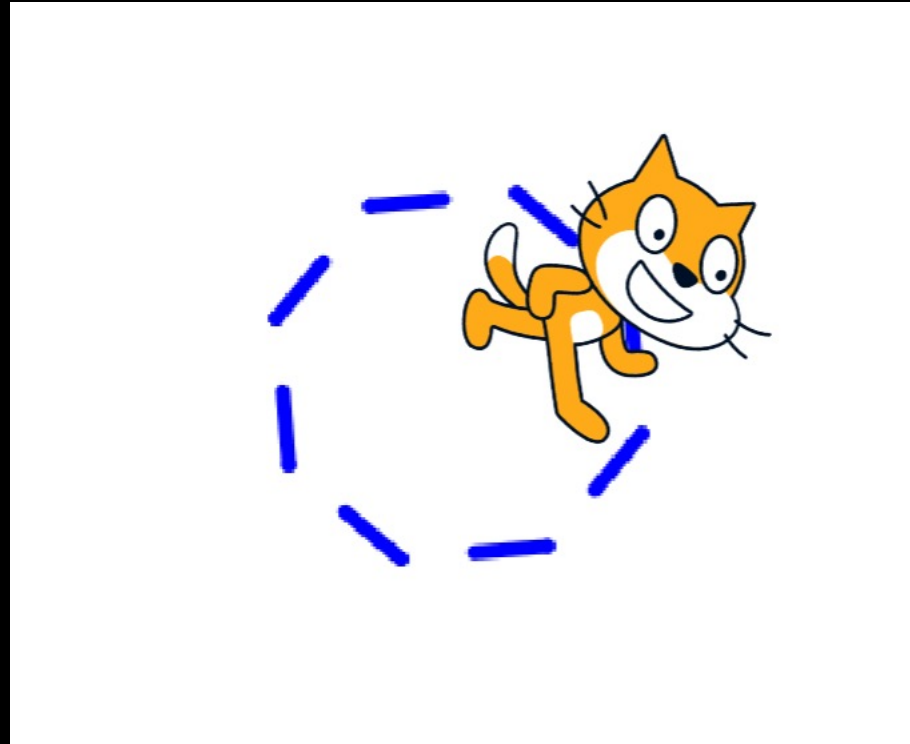
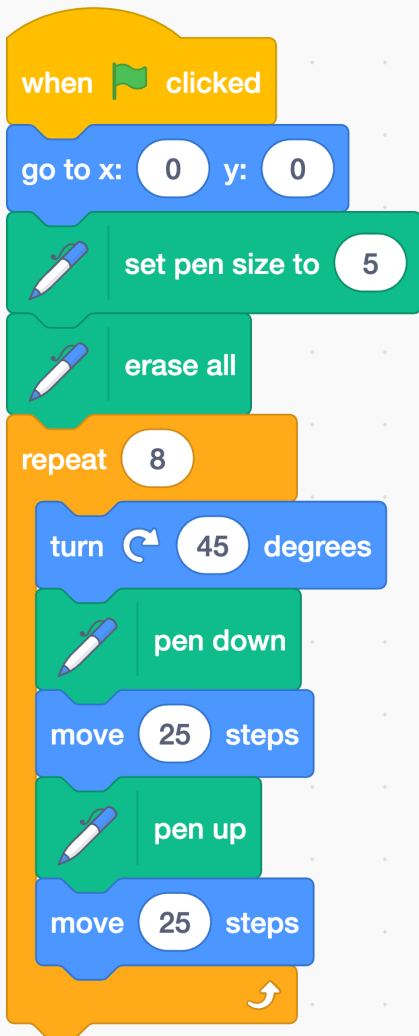
move steps

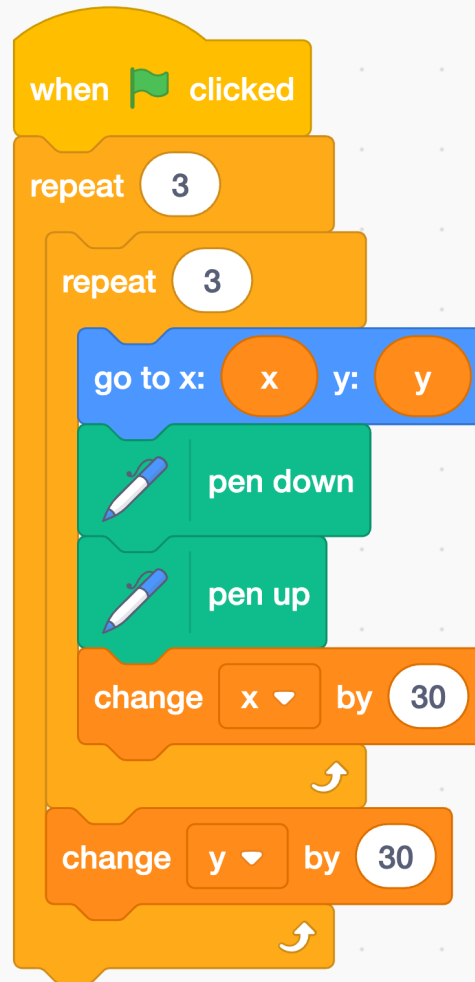
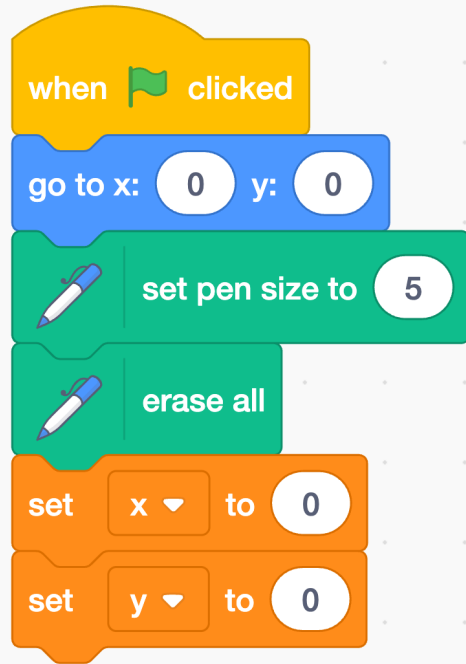


pen up

move steps







Dieses Programm soll ein Raster mit 3x3 Punkten ausgeben.

Tut es das?

Tracing-Problem 1

lernen zu programmieren in C

lernen Probleme zu lösen

Tracing-Problem 2

```
a = 1  
b = 1  
count = 0
```

```
WIEDERHOLE 5 MAL:
```

```
    temp = a + b
```

```
    a = b
```

```
    b = temp
```

```
    WENN b > 10:
```

```
        count = count + 1
```

```
GIB count AUS
```

Was wird
ausgegeben?

lernen Probleme zu lösen mit Funktionen

lernen Probleme zu lösen mit Variablen

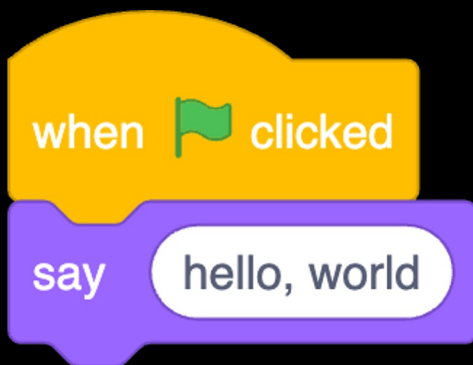
lernen Probleme zu lösen mit Bedingungen

lernen Probleme zu lösen mit Schleifen

lernen Probleme zu lösen (und wie nicht)



Von Scratch zu C



```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    printf("hello, world\n");
```

```
}
```

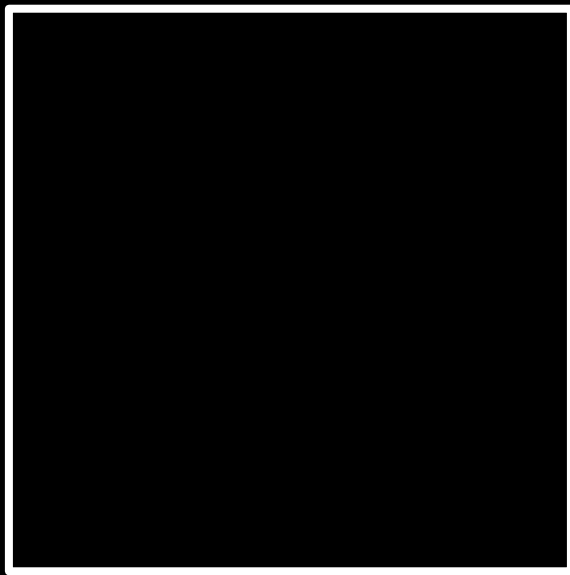
Sourcecode

01111111	01000101	01001100	01000110	00000010	00000001	00000001	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000010	00000000	00111110	00000000	00000001	00000000	00000000	00000000
10110000	00000101	01000000	00000000	00000000	00000000	00000000	00000000
01000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
11010000	00010011	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	01000000	00000000	00111000	00000000
00001001	00000000	01000000	00000000	00100100	00000000	00100001	00000000
00000110	00000000	00000000	00000000	00000101	00000000	00000000	00000000
01000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
01000000	00000000	01000000	00000000	00000000	00000000	00000000	00000000
01000000	00000000	01000000	00000000	00000000	00000000	00000000	00000000
11111000	00000001	00000000	00000000	00000000	00000000	00000000	00000000
11111000	00000001	00000000	00000000	00000000	00000000	00000000	00000000
00001000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000011	00000000	00000000	00000000	00000100	00000000	00000000	00000000
00111000	00000010	00000000	00000000	00000000	00000000	00000000	00000000

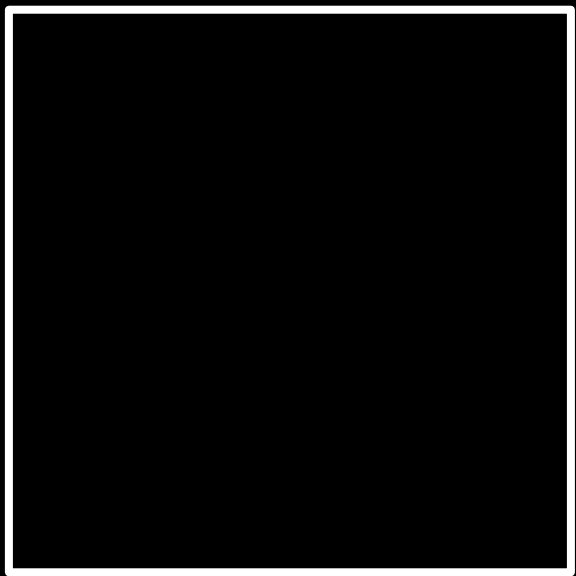
...

Maschinencode

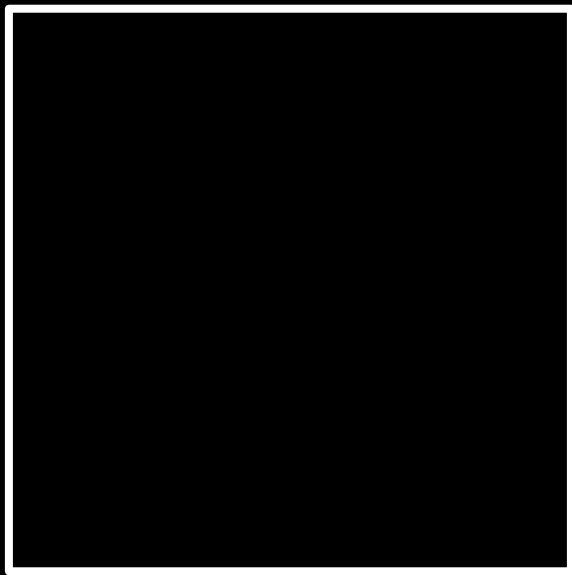
Eingabe →



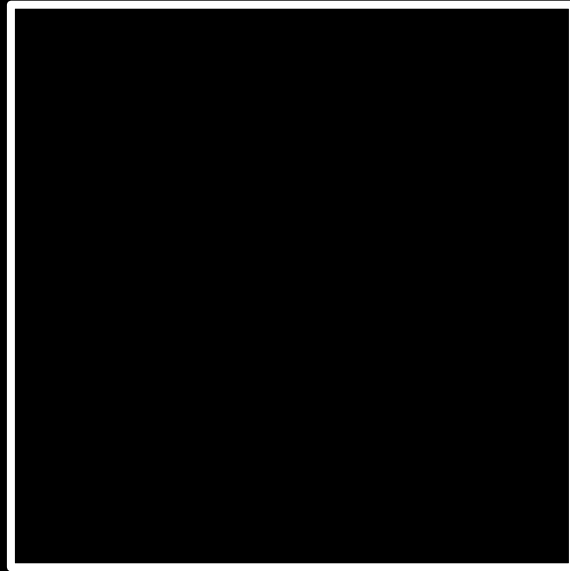
→ Ausgabe



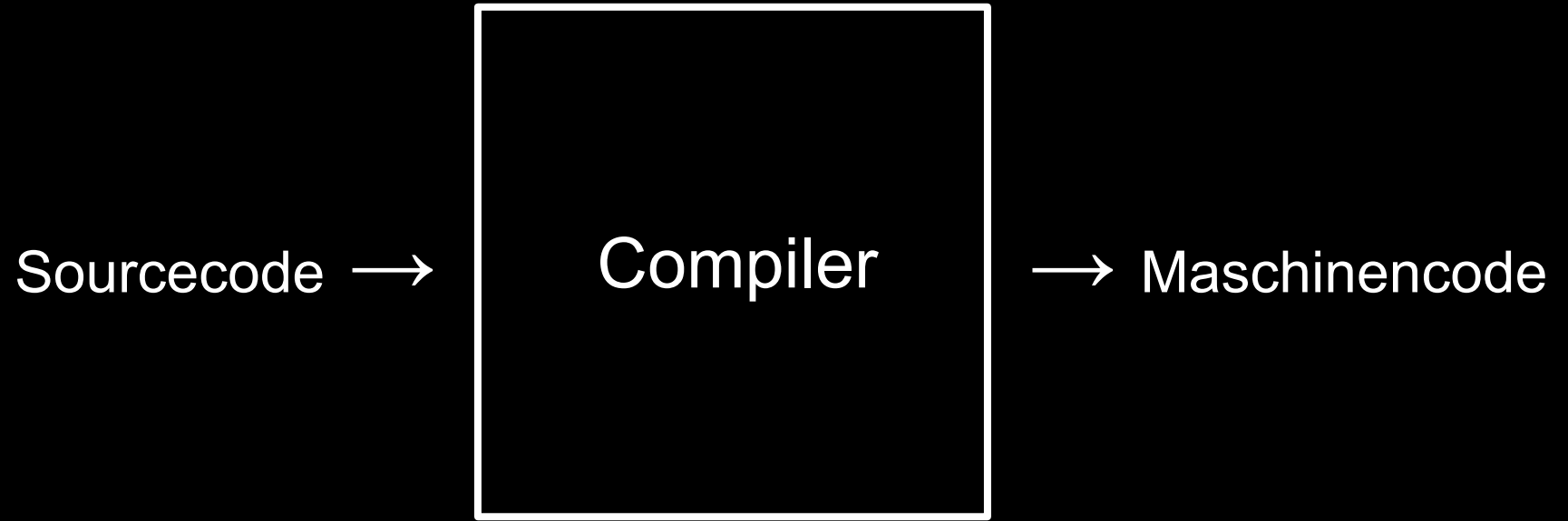
Sourcecode →



Sourcecode →



→ Maschinencode



VS Code for CS50

<https://cs50.dev>



EXPLORER



hello.c



✓ HELLO [CODESPACES]

hello.c



> OUTLINE

> TIMELINE

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     printf("hello, world\n");
6 }
```

TERMINAL



\$ make hello



EXPLORER



✓ HELLO [CODESPACES]

hello.c



> OUTLINE

> TIMELINE



hello.c



```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     printf("hello, world\n");
6 }
```

TERMINAL



\$ make hello



EXPLORER



✓ HELLO [CODESPACES]

hello.c



> OUTLINE

> TIMELINE



hello.c



```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     printf("hello, world\n");
6 }
```

TERMINAL



\$ make hello



EXPLORER



✓ HELLO [CODESPACES]

hello.c



> OUTLINE

> TIMELINE



hello.c



```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     printf("hello, world\n");
6 }
```

TERMINAL



\$ make hello



EXPLORER



hello.c



✓ HELLO [CODESPACES]

hello.c



> OUTLINE

> TIMELINE

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     printf("hello, world\n");
6 }
```

TERMINAL



\$ make hello

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

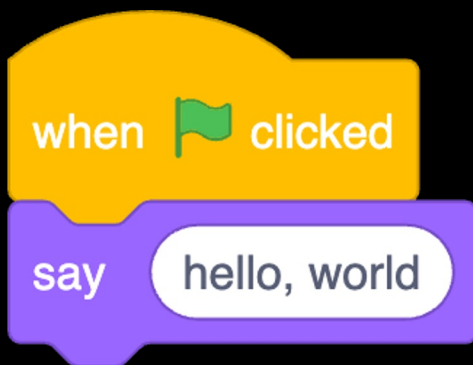
```
    printf("hello, world\n");
```

```
}
```

```
code hello.c
```

```
make hello
```

```
./hello
```



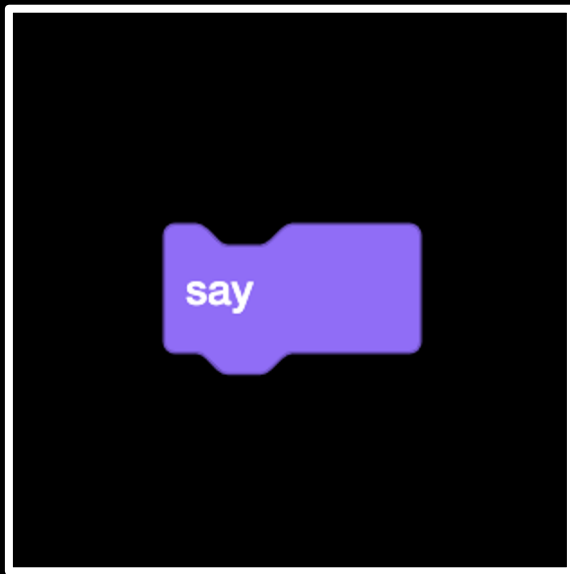
Argumente →

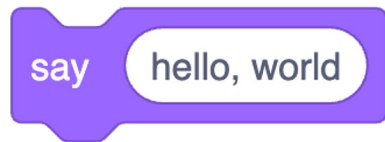


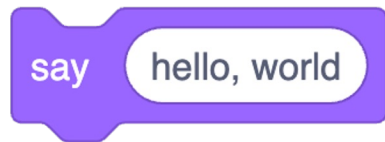
Funktion

→ Seiteneffekte

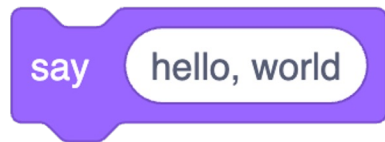
hello, world



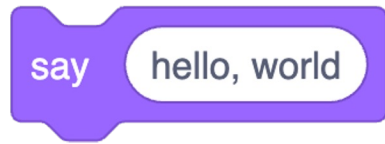




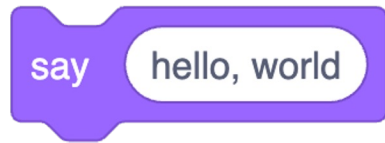
```
print (
```



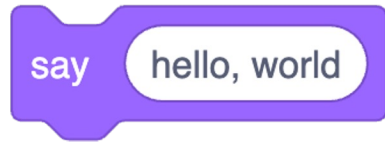
```
printf(      )
```



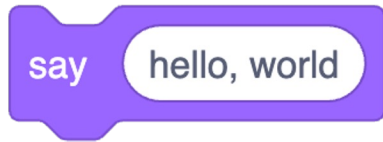
```
printf( hello, world  )
```



```
printf("hello, world ")
```



```
printf("hello, world\n")
```



```
printf("hello, world\n");
```

Escape-Sequenzen

z.B. `\n`

Header-Dateien

`stdio.h`

Bibliotheken

Manual Pages

manual.cs50.io

stdio.h

manual.cs50.io/#stdio.h

manual.cs50.io/3/printf

cs50.h

manual.cs50.io/#cs50.h

get_char

get_double

get_float

get_int

get_long

get_string

...

get_char

get_double

get_float

get_int

get_long

get_string

...

when  clicked

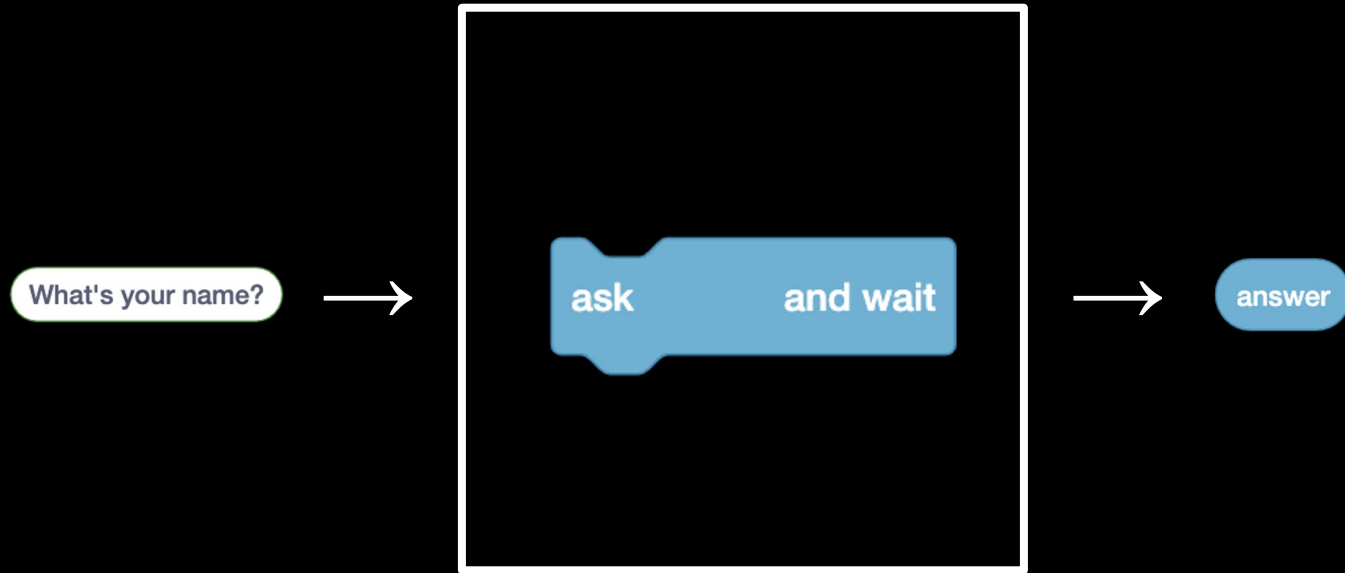
ask and wait

say

Argumente →

Funktion

→ Rückgabewert



ask

What's your name?

and wait

answer

ask What's your name? and wait

answer

```
get_string( )
```

ask What's your name? and wait

answer

```
get_string( What's your name? )
```

ask What's your name? and wait

answer

```
get_string("What's your name? ")
```

ask What's your name? and wait

answer

```
answer = get_string("What's your name? ")
```

ask What's your name? and wait

answer

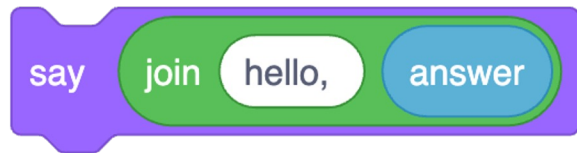
```
string answer = get_string("What's your name? ")
```

ask What's your name? and wait

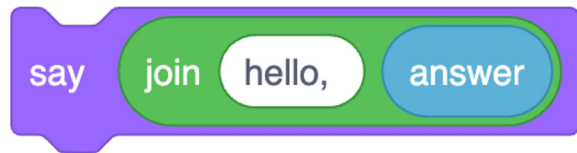
answer

```
string answer = get_string("What's your name? ");
```

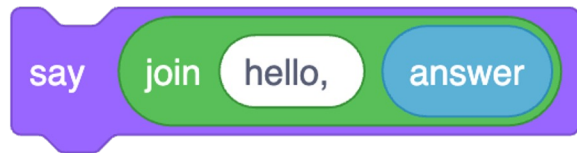




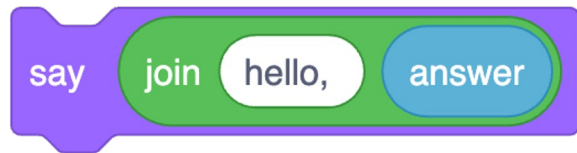
```
printf(          );
```



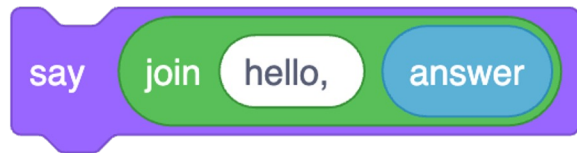
```
printf( hello, %s      );
```



```
printf("hello, %s  " );
```



```
printf("hello, %s\n"      );
```



```
printf("hello, %s\n", answer);
```

Datentypen

bool

char

double

float

int

long

string

...

bool

char

double

float

int

long

string

...

bool

char

double

float

int

long

string

...

get_char

get_double

get_float

get_int

get_long

get_string

...

get_char

get_double

get_float

get_int

get_long

get_string

...

Formatcodes

%c

%f

%i

%li

%s

%c

%f

%i

%li

%s

%c

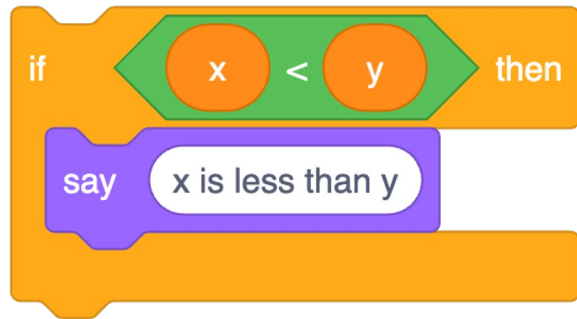
%f

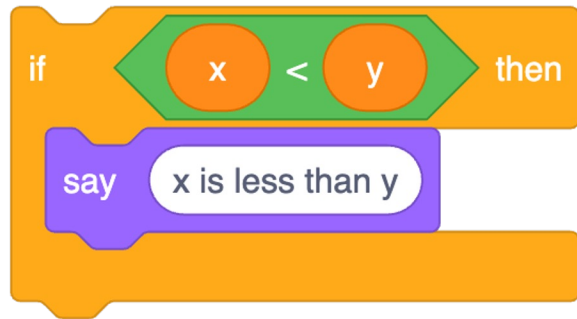
%i

%li

%s

Bedingte Anweisungen

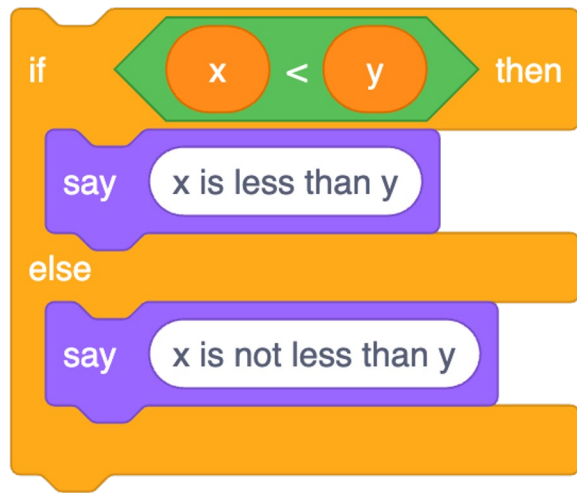


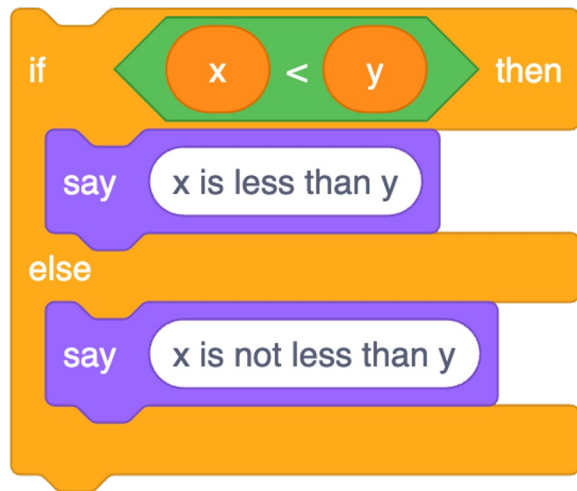


```
if (x < y)
{
}
}
```

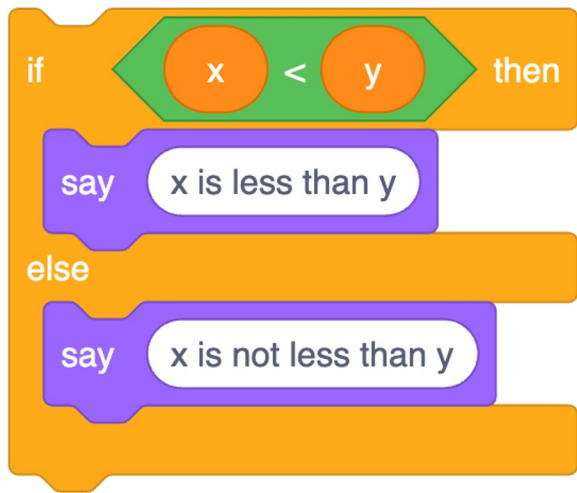


```
if (x < y)
{
    printf("x is less than y\n");
}
```

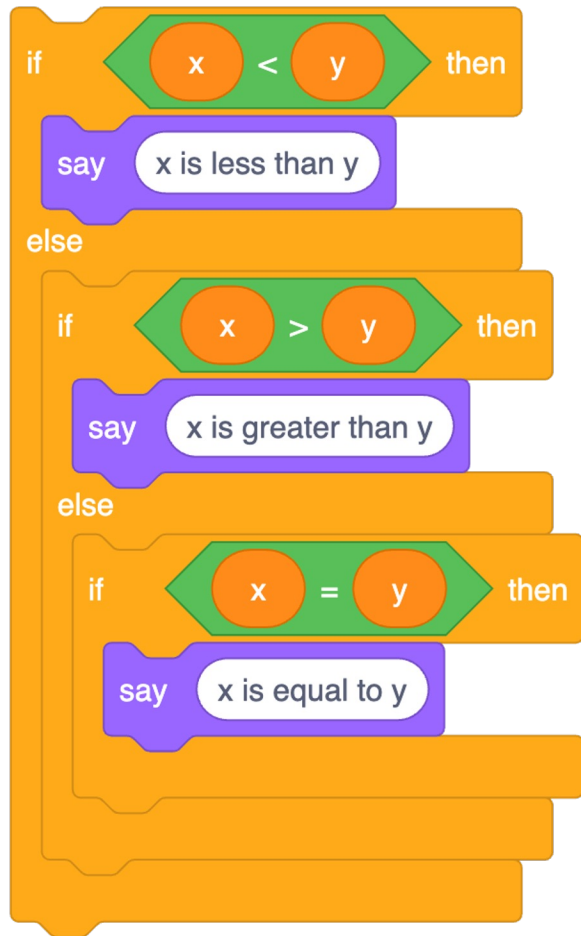


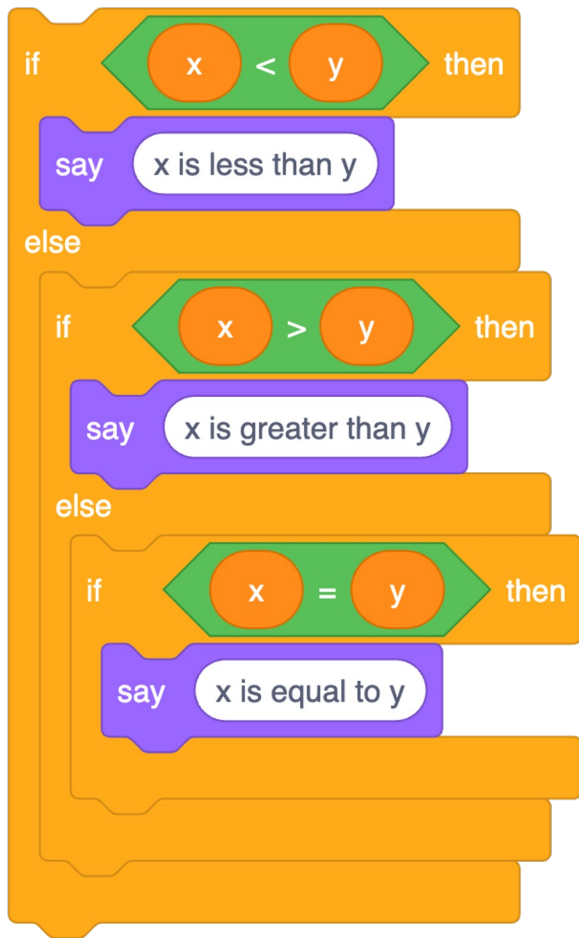


```
if (x < y)
{
}
else
{
}
```

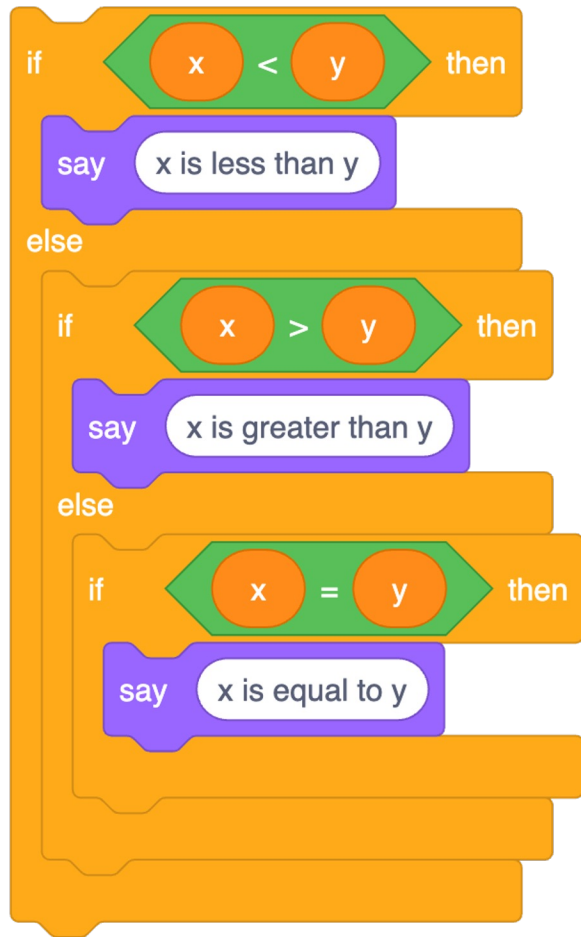


```
if (x < y)
{
    printf("x is less than y\n");
}
else
{
    printf("x is not less than y\n");
}
```

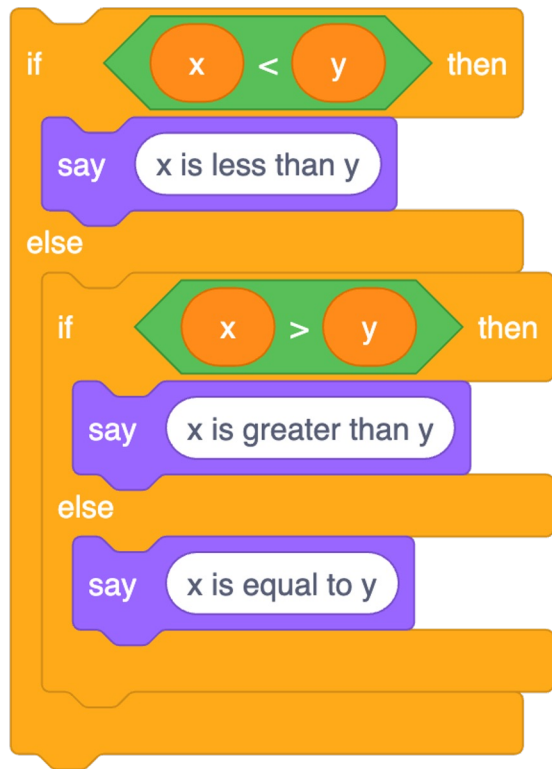




```
if (x < y)
{
}
else if (x > y)
{
}
else if (x == y)
{
}
```

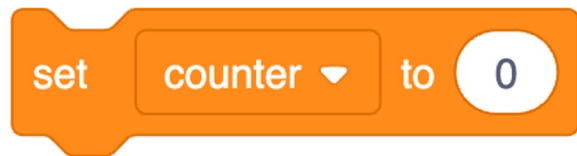


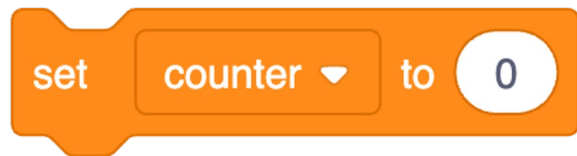
```
if (x < y)
{
    printf("x is less than y\n");
}
else if (x > y)
{
    printf("x is greater than y\n");
}
else if (x == y)
{
    printf("x is equal to y\n");
}
```



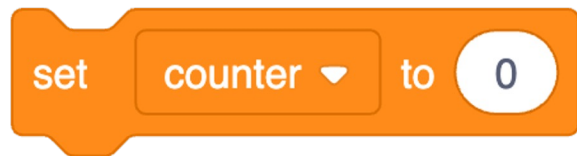
```
if (x < y)
{
    printf("x is less than y\n");
}
else if (x > y)
{
    printf("x is greater than y\n");
}
else
{
    printf("x is equal to y\n");
}
```

Variablen

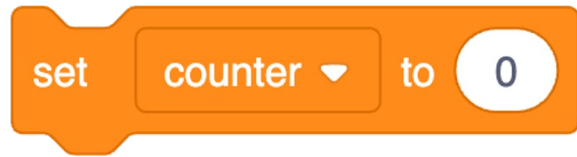




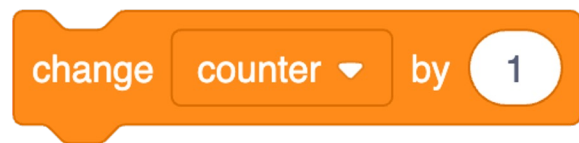
```
counter = 0
```

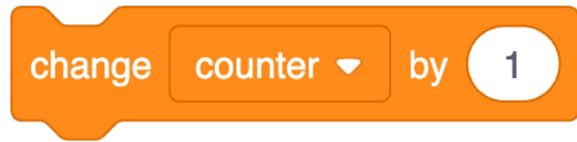


```
int counter = 0
```

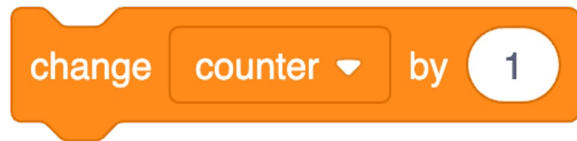


```
int counter = 0;
```

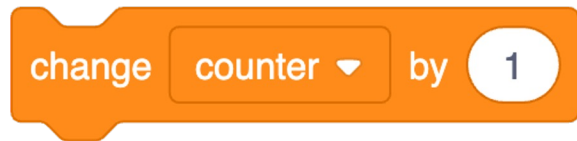




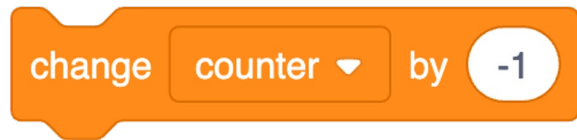
```
counter = counter + 1;
```



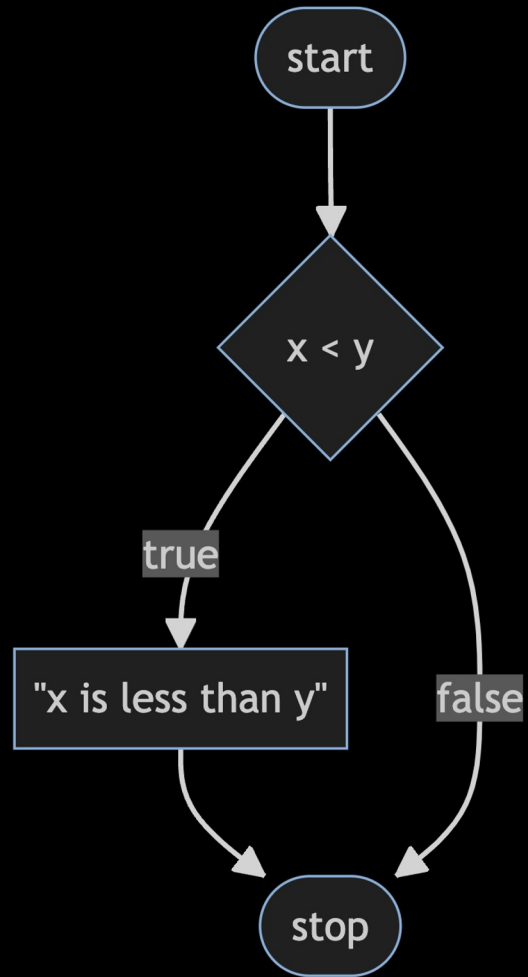
```
counter += 1;
```

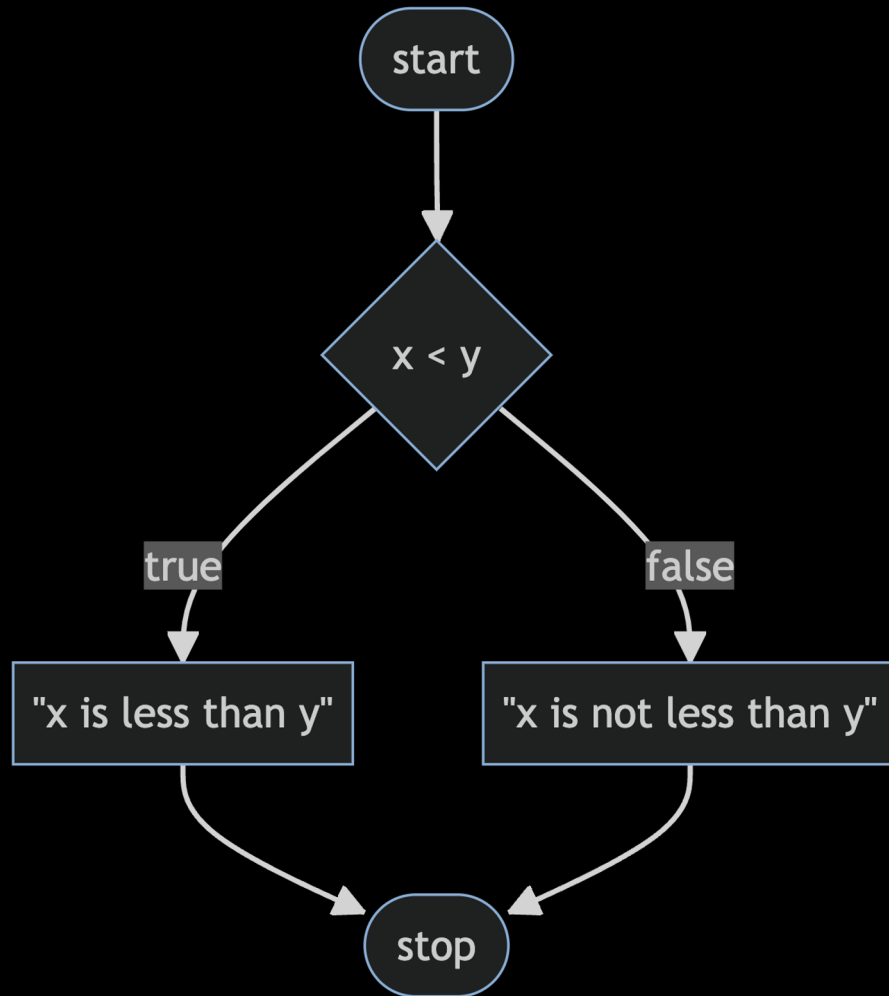


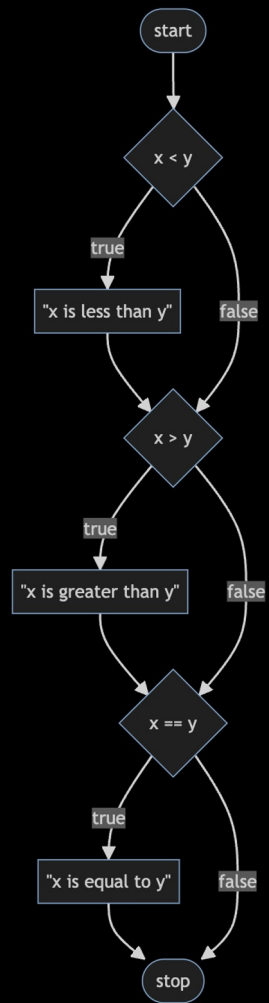
```
counter++;
```

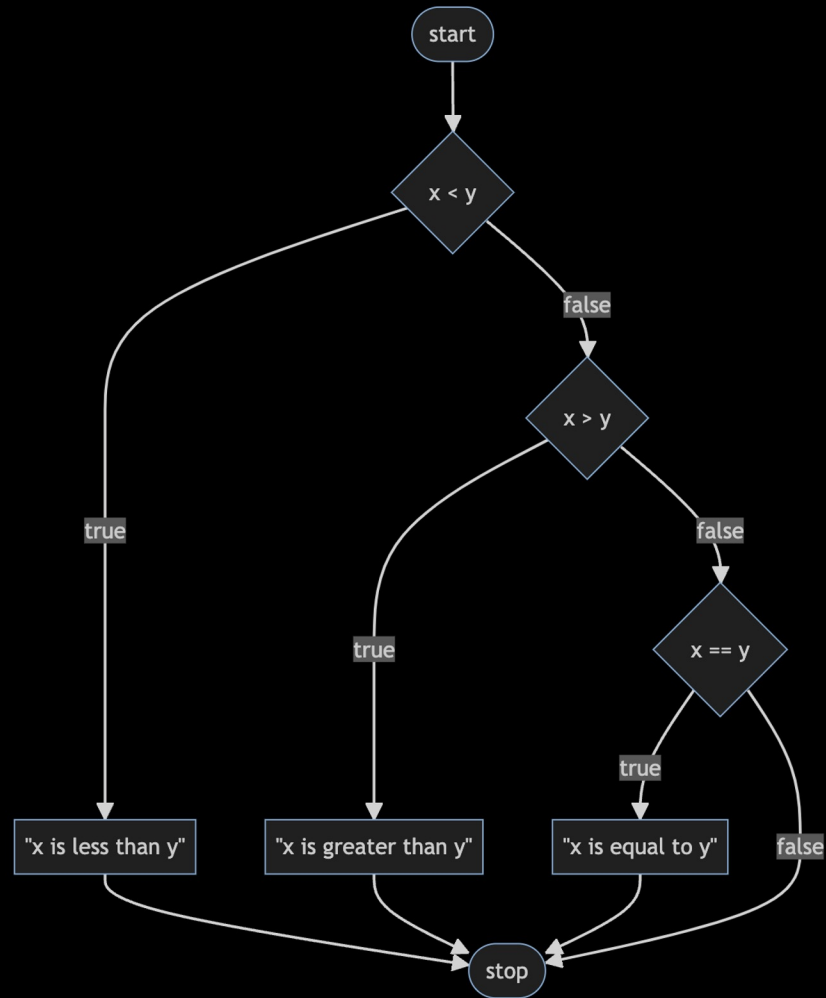


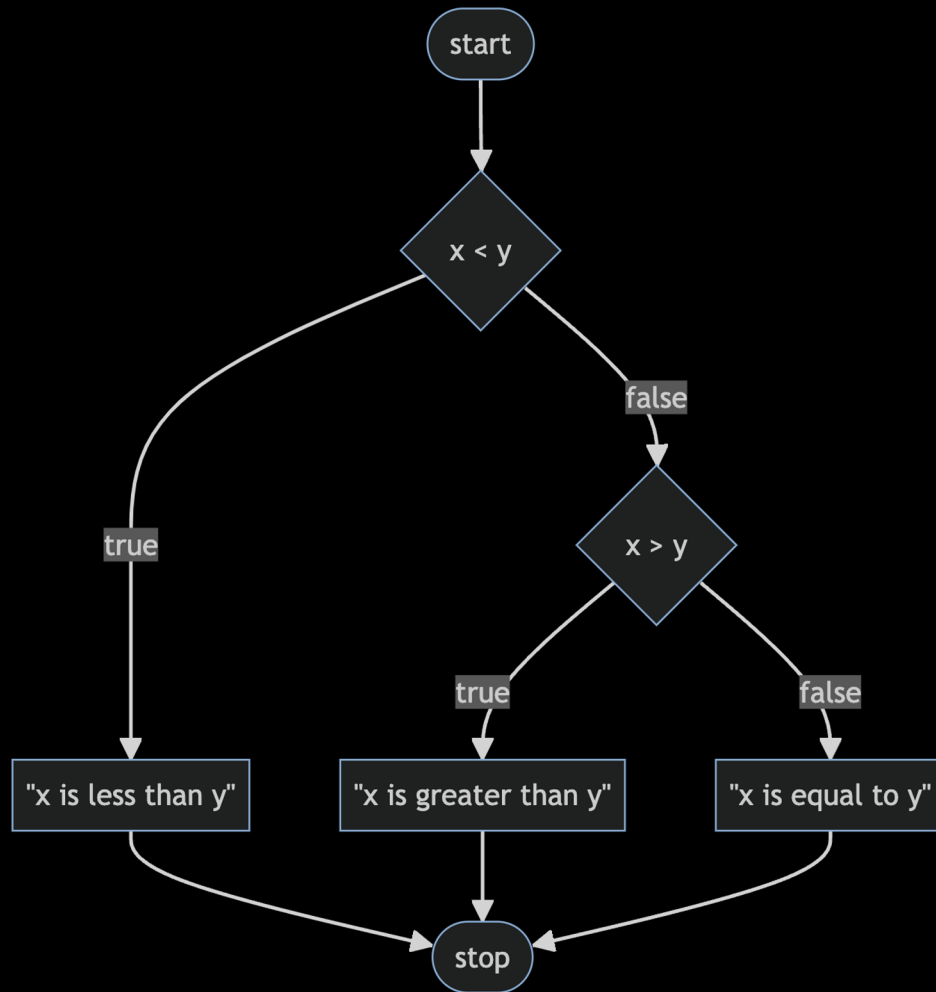
```
counter--;
```











bool

char

double

float

int

long

string

...

bool

char

double

float

int

long

string

...

get_char

get_double

get_float

get_int

get_long

get_string

...

Schleifen

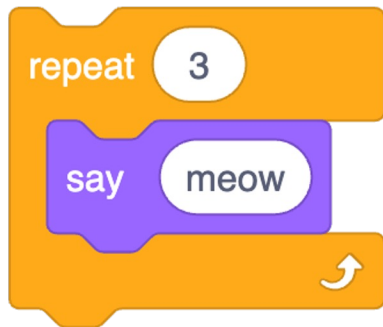




```
int counter = 3;
while (counter > 0)
{
    printf("meow\n");
    counter = counter - 1;
}
```



```
int counter = 3;
while (counter > 0)
{
    printf("meow\n");
    counter = counter - 1;
}
```



```
int counter = 3;  
while (counter > 0)  
{  
    printf("meow\n");  
    counter = counter - 1;  
}
```



```
int counter = 3;  
while (counter > 0)  
{  
    printf("meow\n");  
    counter = counter - 1;  
}
```



```
int counter = 3;
while (counter > 0)
{
    printf("meow\n");
    counter = counter - 1;
}
```



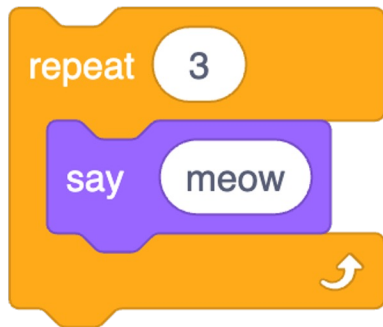
```
int counter = 3;
while (counter > 0)
{
    printf("meow\n");
    counter = counter - 1;
}
```



```
int counter = 3;
while (counter > 0)
{
    printf("meow\n");
    counter = counter - 1;
}
```



```
int counter = 3;  
while (counter > 0)  
{  
    printf("meow\n");  
    counter = counter - 1;  
}
```



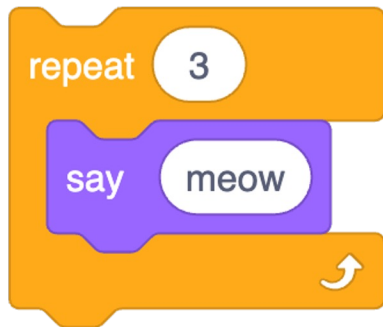
```
int counter = 3;
while (counter > 0)
{
    printf("meow\n");
    counter = counter - 1;
}
```



```
int counter = 3;
while (counter > 0)
{
    printf("meow\n");
    counter = counter - 1;
}
```



```
int counter = 3;
while (counter > 0)
{
    printf("meow\n");
    counter = counter - 1;
}
```



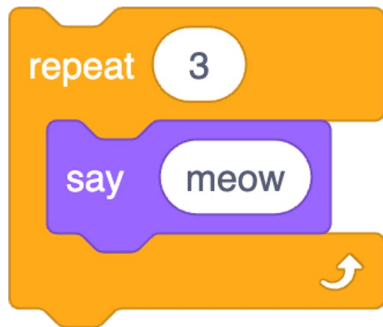
```
int counter = 3;
while (counter > 0)
{
    printf("meow\n");
    counter = counter - 1;
}
```



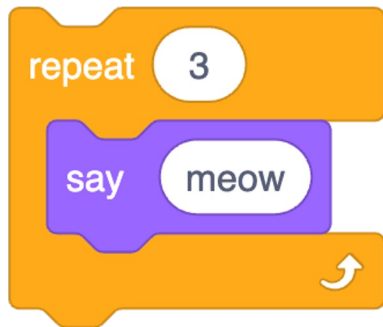
```
int counter = 3;
while (counter > 0)
{
    printf("meow\n");
    counter = counter - 1;
}
```



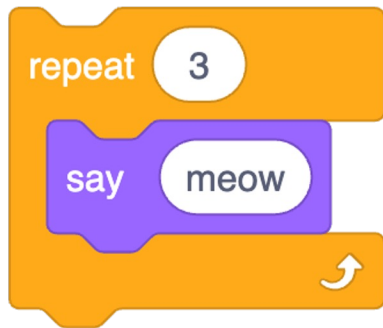
```
int counter = 3;
while (counter > 0)
{
    printf("meow\n");
    counter = counter - 1;
}
```



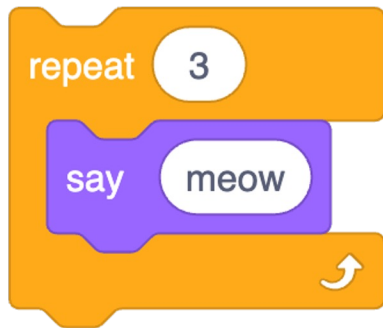
```
int i = 3;
while (i > 0)
{
    printf("meow\n");
    i = i - 1;
}
```



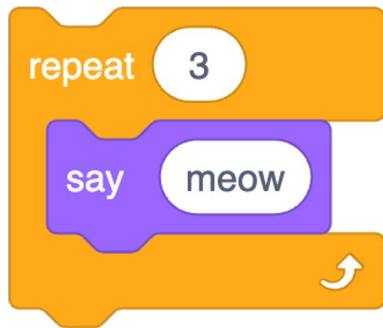
```
int i = 3;
while (i > 0)
{
    printf("meow\n");
    i -= 1;
}
```



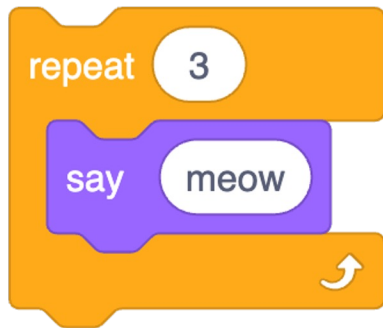
```
int i = 3;
while (i > 0)
{
    printf("meow\n");
    i--;
}
```



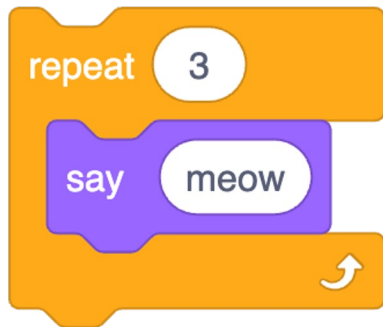
```
int i = 1;
while (i <= 3)
{
    printf("meow\n");
    i++;
}
```



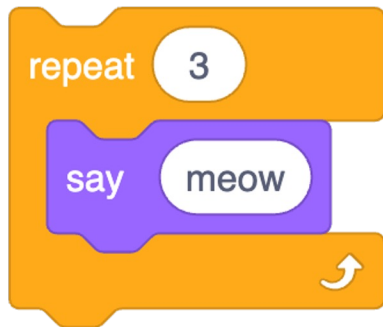
```
int i = 0;  
while (i < 3)  
{  
    printf("meow\n");  
    i++;  
}
```



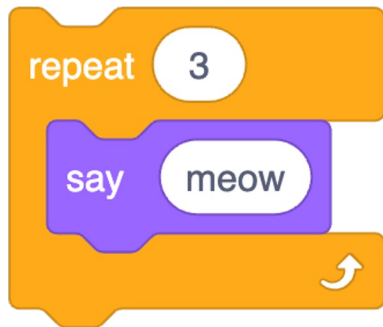
```
for (int i = 0; i < 3; i++)  
{  
    printf("meow\n");  
}
```



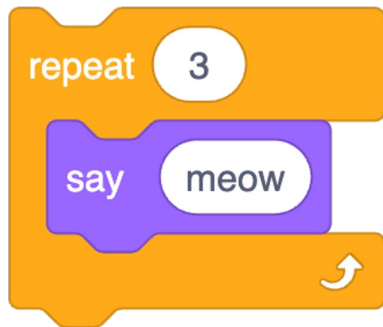
```
for (int i = 0; i < 3; i++)  
{  
    printf("meow\n");  
}
```



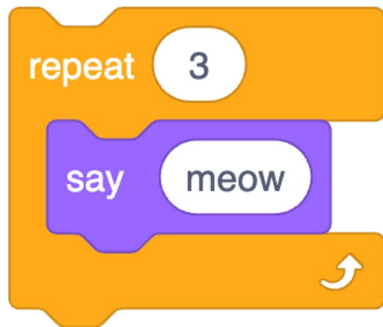
```
for (int i = 0; i < 3; i++)  
{  
    printf("meow\n");  
}
```



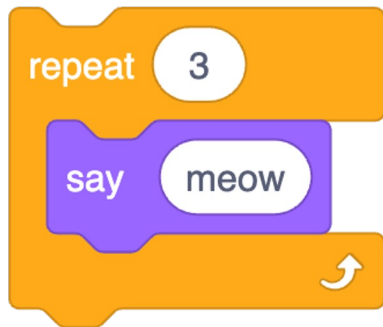
```
for (int i = 0; i < 3; i++)  
{  
    printf("meow\n");  
}
```



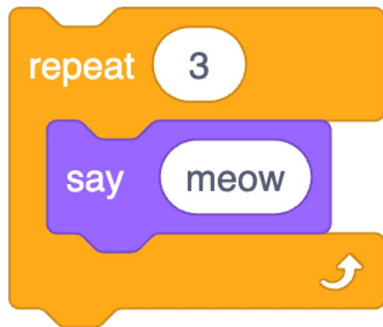
```
for (int i = 0; i < 3; i++)  
{  
    printf("meow\n");  
}
```



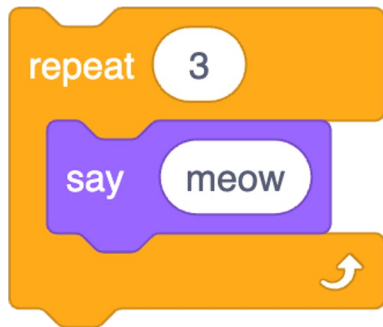
```
for (int i = 0; i < 3; i++)  
{  
    printf("meow\n");  
}
```



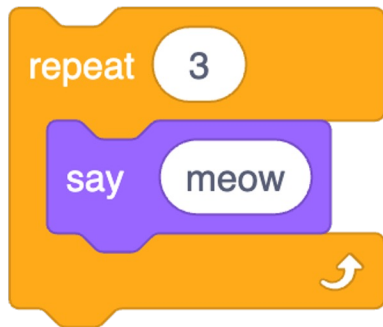
```
for (int i = 0; i < 3; i++)  
{  
    printf("meow\n");  
}
```



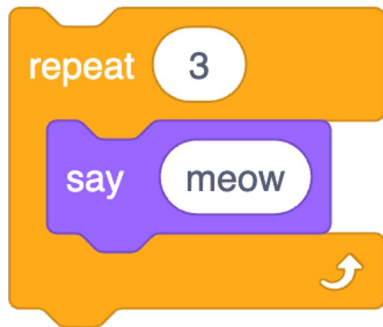
```
for (int i = 0; i < 3; i++)  
{  
    printf("meow\n");  
}
```



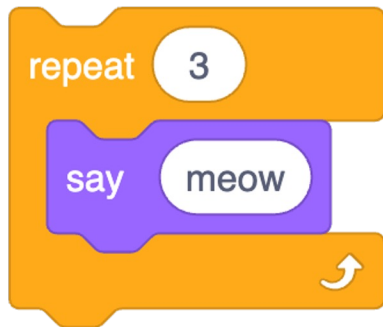
```
for (int i = 0; i < 3; i++)  
{  
    printf("meow\n");  
}
```



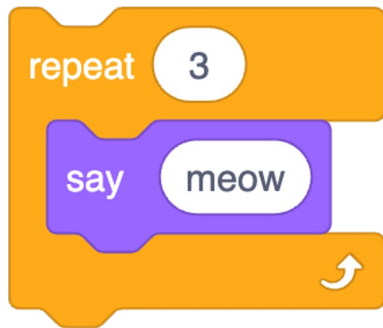
```
for (int i = 0; i < 3; i++)  
{  
    printf("meow\n");  
}
```



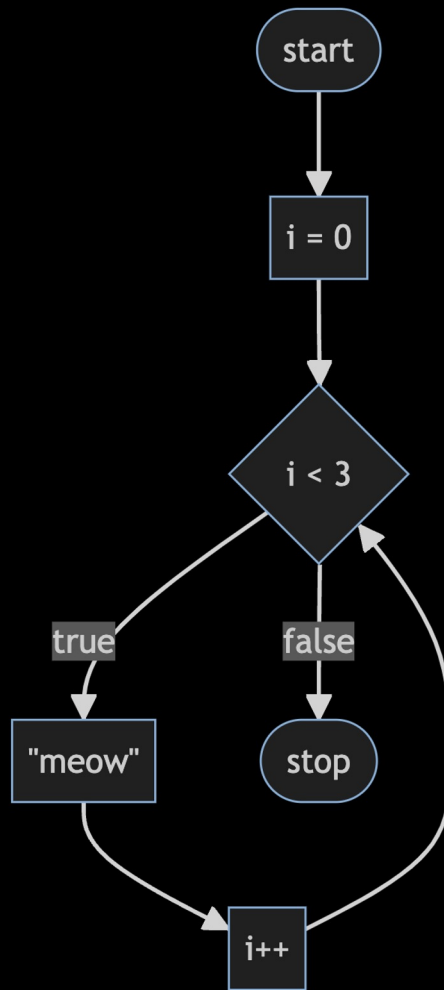
```
for (int i = 0; i < 3; i++)  
{  
    printf("meow\n");  
}
```



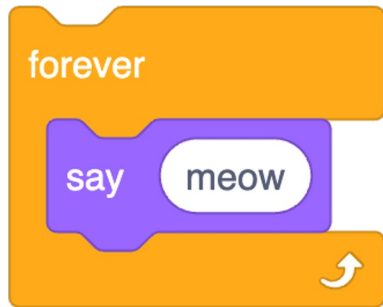
```
for (int i = 0; i < 3; i++)  
{  
    printf("meow\n");  
}
```



```
for (int i = 0; i < 3; i++)  
{  
    printf("meow\n");  
}
```



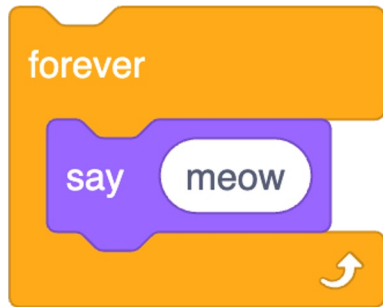




```
while (    )  
{  
  
}
```



```
while (true)
{
}
```



```
while (true)
{
    printf("meow\n");
}
```



Parson-Problem

```
1 printf("warm\n");
2 }
3 else if (temp > 25)
4 if (temp <= 10)
5 #include <cs50.h>
6 {
7 int temp = get_int("Temperatur: ");
8 else
9 int main(void)
10 printf("heiß\n");
11 #include <stdio.h>
12 if (temp < 10)
13 printf("kalt\n");
14 }
15 {
16 else if (temp >= 25)
```

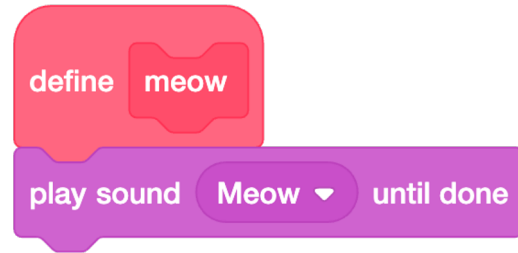
Schreiben Sie ein Programm, das eine Temperatur in °C einliest und ausgibt, ob es “kalt” (unter 10°), “warm” (10-25°) oder “heiß” (über 25°) ist.

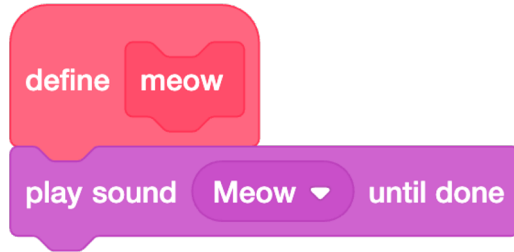
```
#include <cs50.h>
#include <stdio.h>
```

Parson-Problem

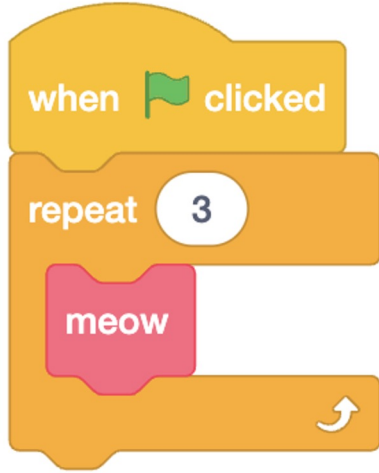
```
int main(void)
{
    int temp = get_int("Temperatur: ");
    if (temp < 10)
    {
        printf("kalt\n");
    }
    else if (temp > 25)
    {
        printf("heiß\n");
    }
    else
    {
        printf("warm\n");
    }
}
```

Schreiben Sie ein Programm, das eine Temperatur in °C einliest und ausgibt, ob es “kalt” (unter 10°), “warm” (10-25°) oder “heiß” (über 25°) ist.



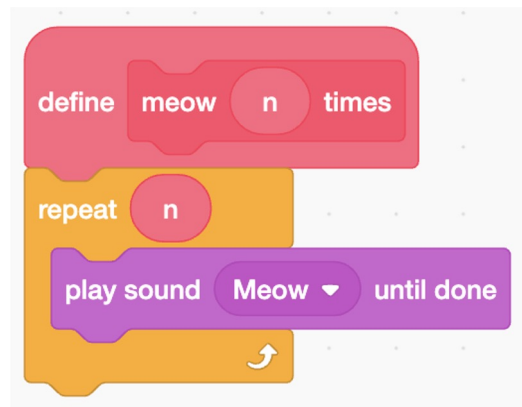


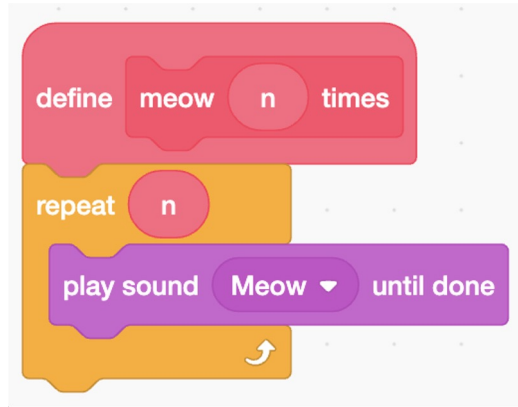
```
void meow(void)
{
    printf("meow\n");
}
```



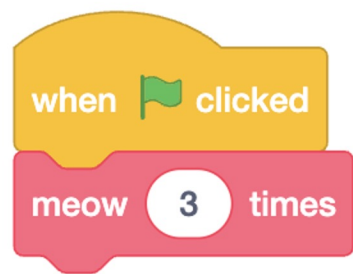


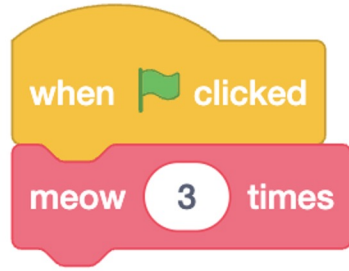
```
int main(void)
{
    for (int i = 0; i < 3; i++)
    {
        meow();
    }
}
```





```
void meow(int n)
{
    for (int i = 0; i < n; i++)
    {
        printf("meow\n");
    }
}
```





```
int main(void)
{
    meow(3);
}
```

Argumente →



Funktion

→ Seiteneffekte

Argumente →



Funktion

→ Rückgabewert

Scope

Linux

graphical user interface

GUI



EXPLORER



hello.c



✓ HELLO [CODESPACES]

hello.c



> OUTLINE

> TIMELINE

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     printf("hello, world\n");
6 }
```

TERMINAL



\$ make hello



EXPLORER



✓ HELLO [CODESPACES]

hello.c



> OUTLINE

> TIMELINE



hello.c



```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     printf("hello, world\n");
6 }
```

TERMINAL



\$ make hello

Kommandozeilen-Interface

CLI

cd

cp

ls

mkdir

mv

rm

rmdir

...

MARIO
000000

● × 00

WORLD
1-1

TIME

SUPER MARIO BROS.

©1985 NINTENDO

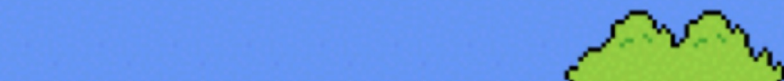


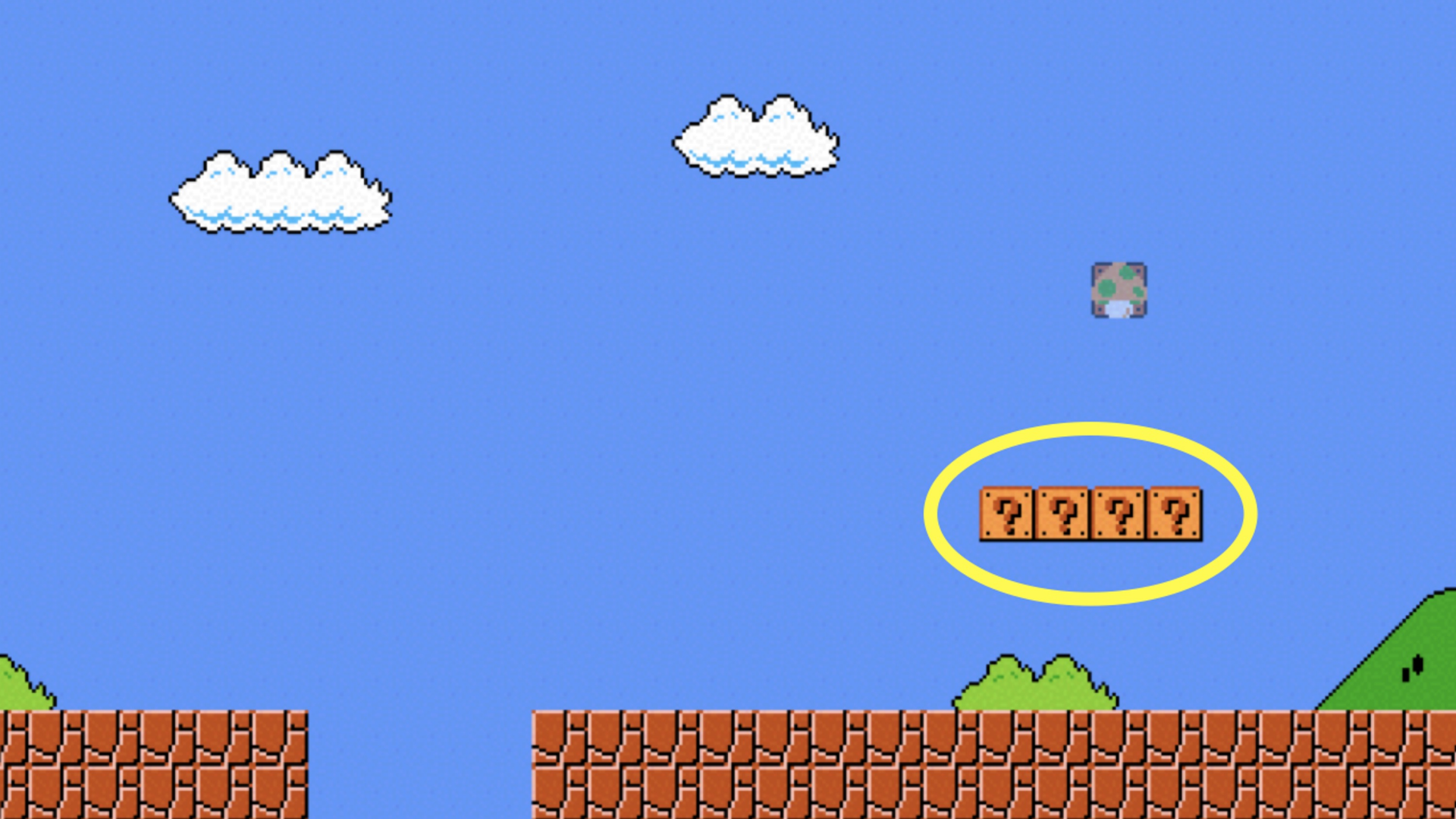
1 PLAYER GAME

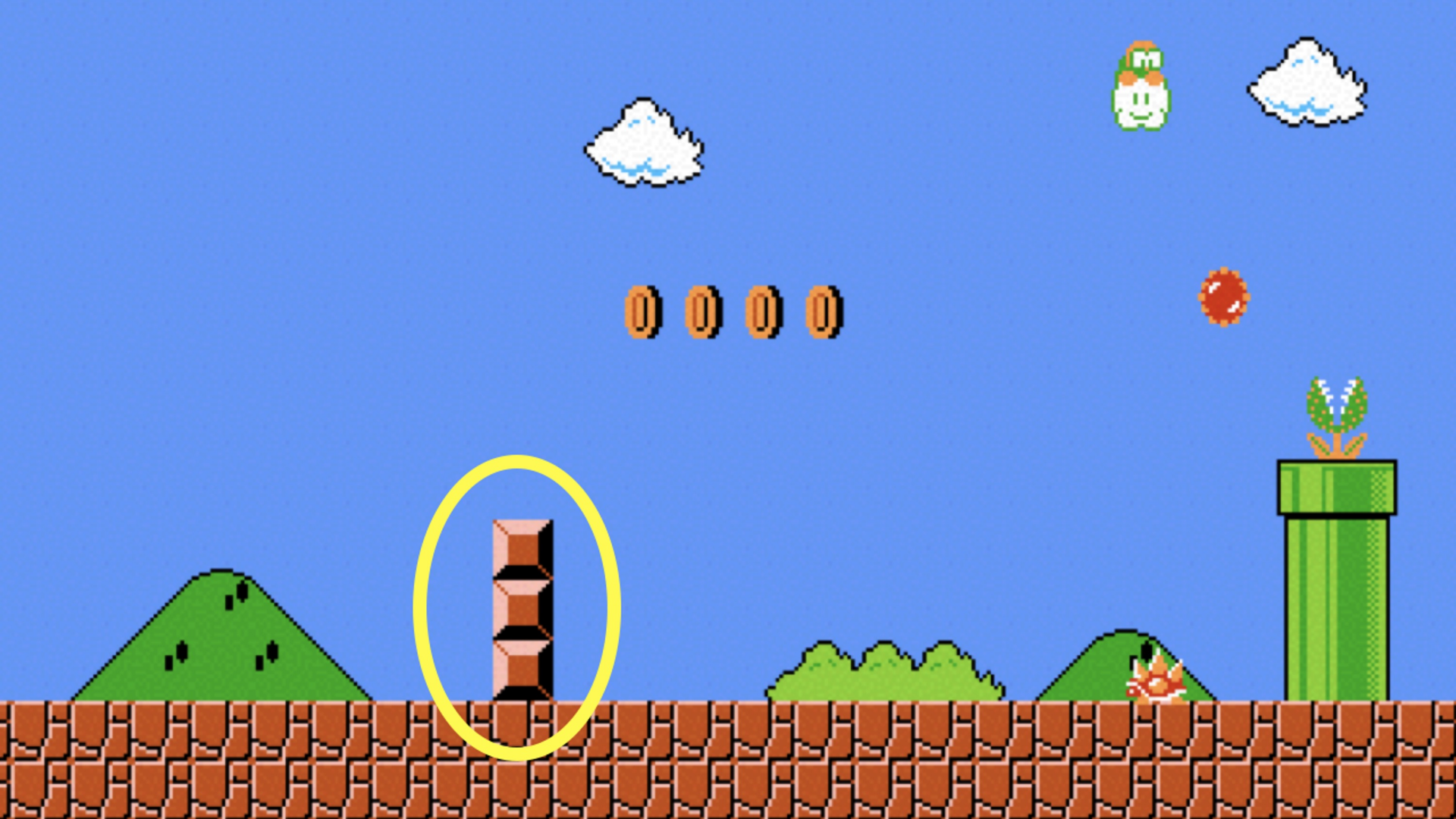
2 PLAYER GAME

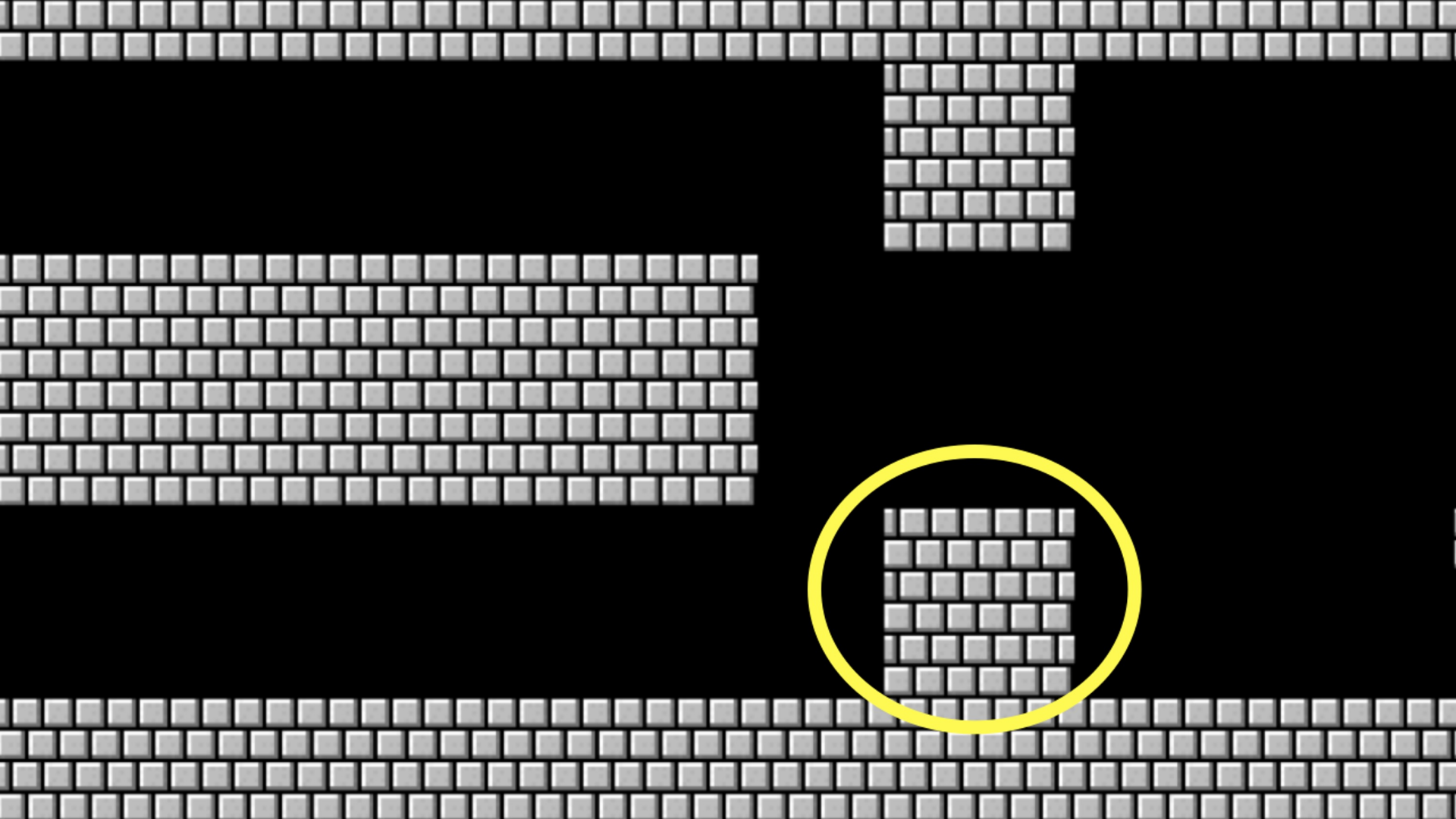
TOP- 000000











Konstanten

Kommentare



0000

0001

0010

0011

0100

0101

0110

~~0~~111

1000

Integer-Überlauf

O O

11111111111111111111111111111111

4294967295

2147483647

-2147483648

bool

char

double

float

int

long

string

...

bool

char

double

float

int

long

string

...

get_char

get_double

get_float

get_int

get_long

get_string

...

get_char

get_double

get_float

get_int

get_long

get_string

...

%c

%f

%i

%li

%s

%c

%f

%i

%li

%s

Truncation

Type-Casting

Fließkomma-Ungenauigkeit

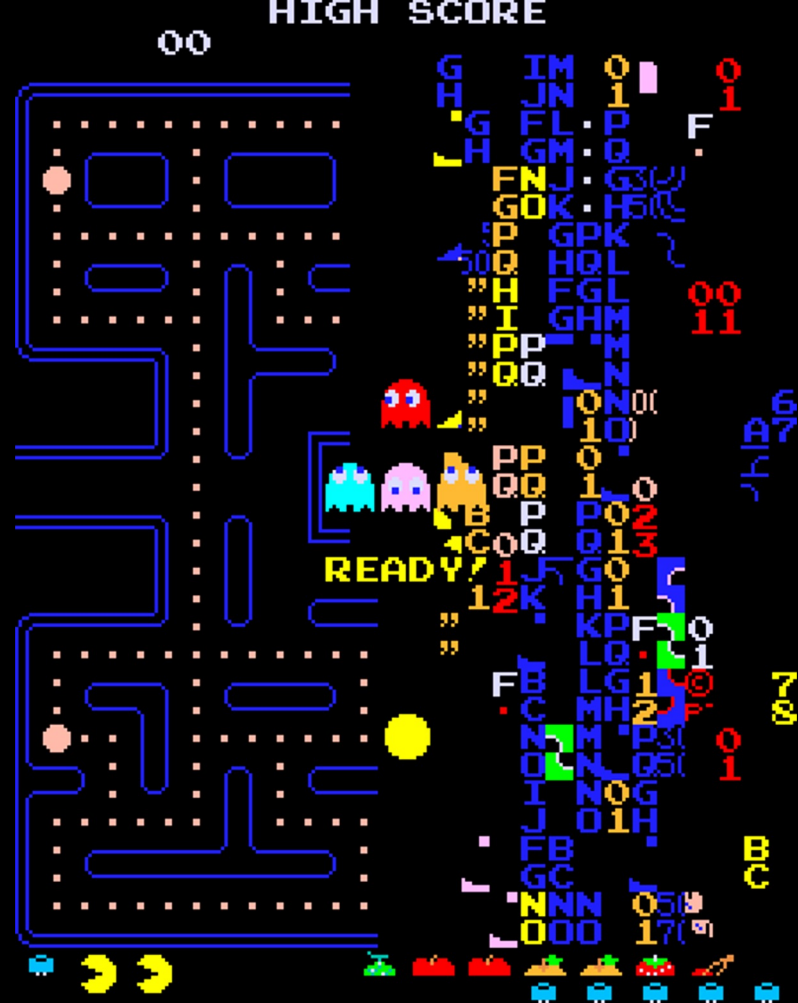
1999

1999

1900

19 January 2038

13 December 1901



Using MAME to warp to level 256, the split screen is shown.

\$74800

HIGH SCORE
\$74800

L-22

BONUS
3700



$$10 \times (\text{level} + 4)$$

$$10 \times (22 + 4)$$

260



Korrektheit, Design, Stil

Dies war Inf-Einf-B.