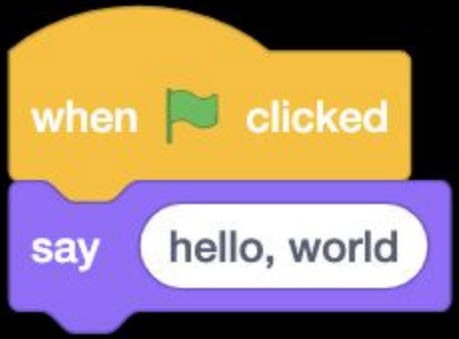
This is CS50


when clicked
say
hello, world

```c
#include <stdio.h>

int main(void)
{
    printf("hello, world\n");
}
```

- functions
  - arguments, return values
- conditionals
- Boolean expressions
- loops
- variables
- ...

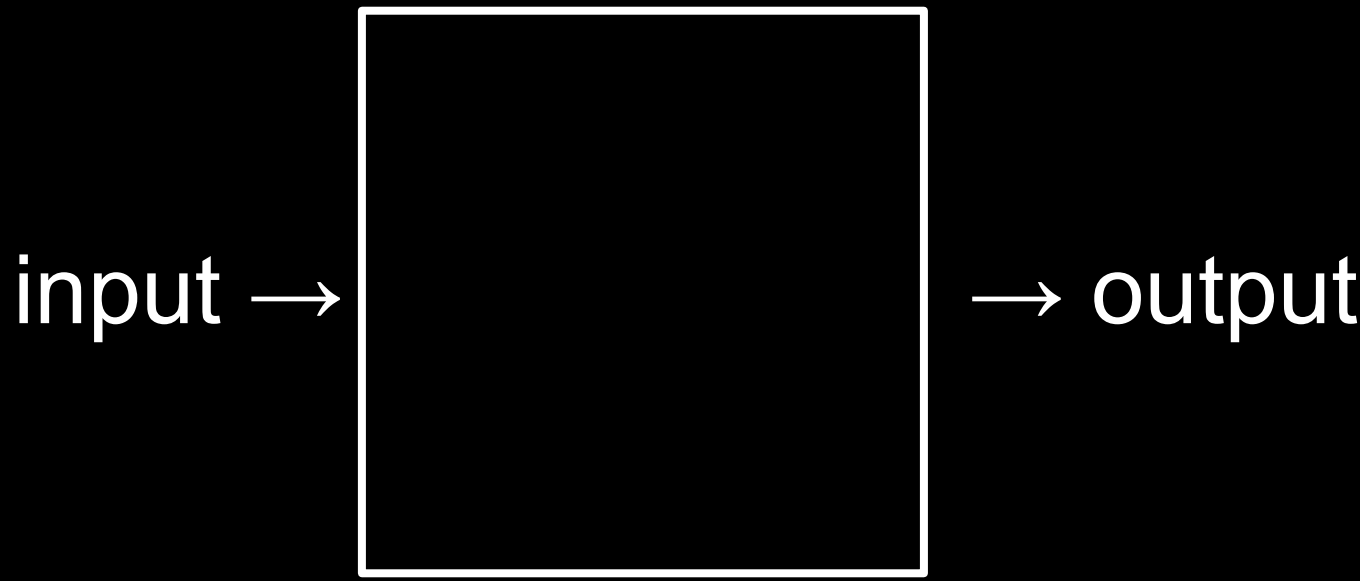
correctness

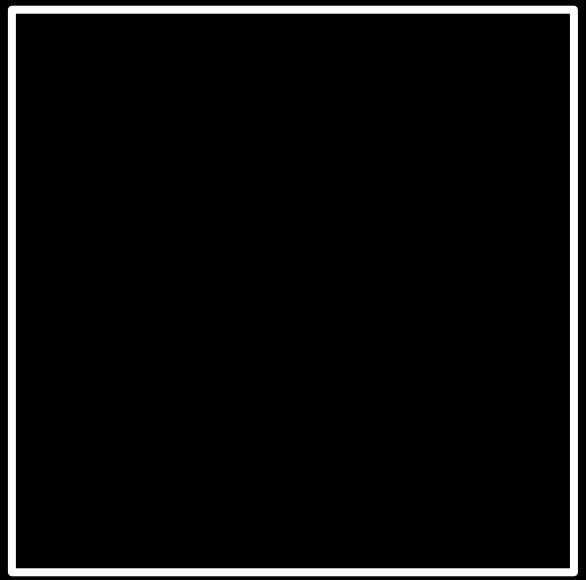
design

style

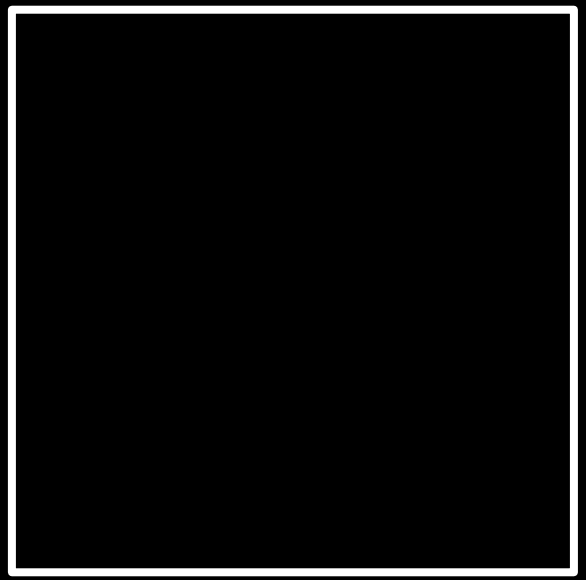
```c
#include <stdio.h>

int main(void)
{
    printf("hello, world\n");
}
```

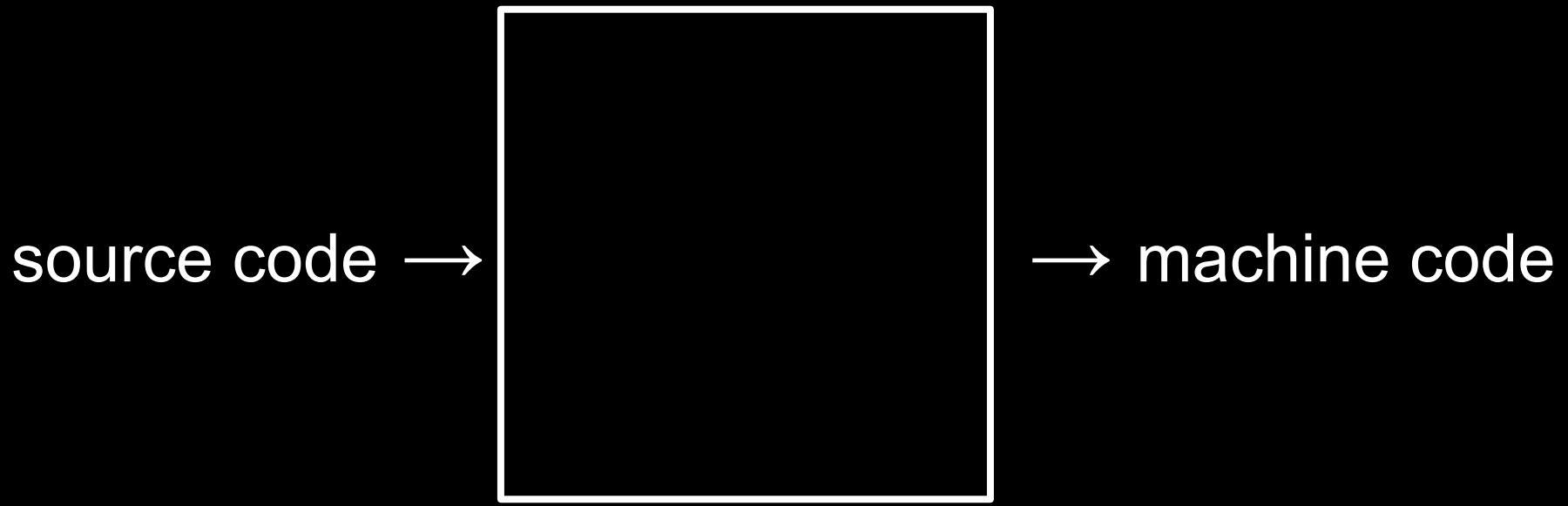
```
01111111 01000101 01001100 01000110 00000010 00000001 00000001 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000010 00000000 00111110 00000000 00000001 00000000 00000000 00000000
10110000 00000101 01000000 00000000 00000000 00000000 00000000 00000000
01000000 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
...
```

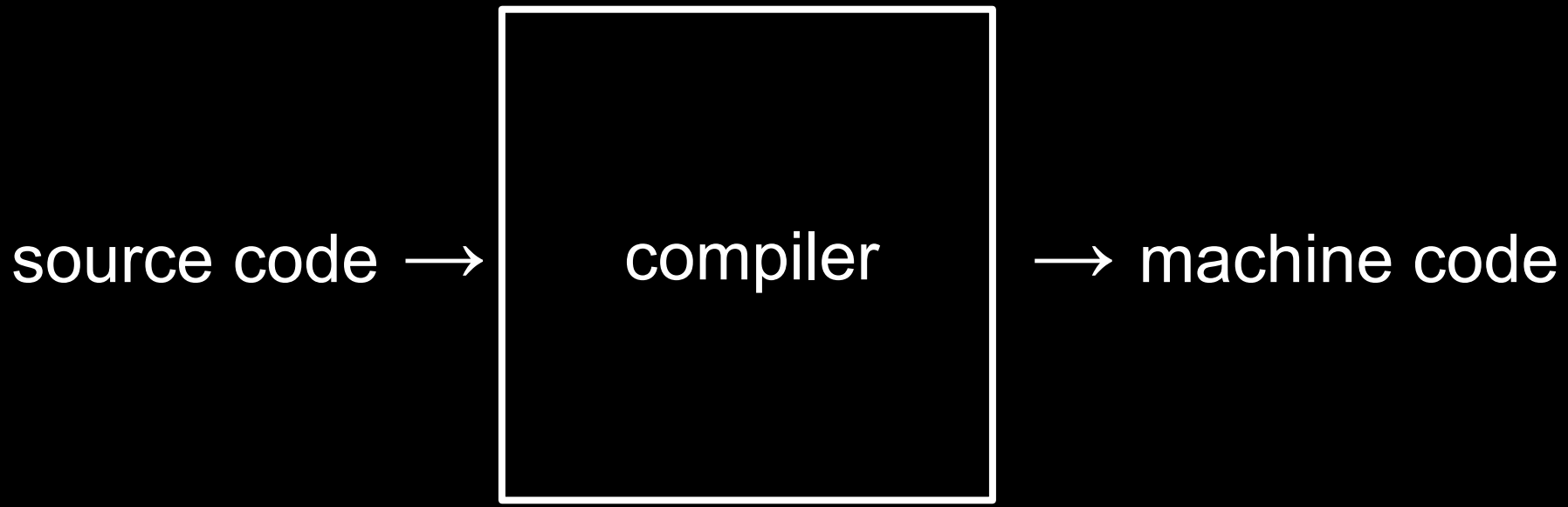

input →
→ output



source code ⟶

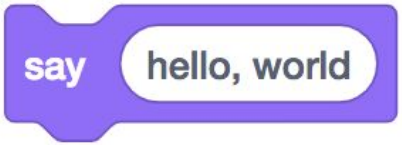


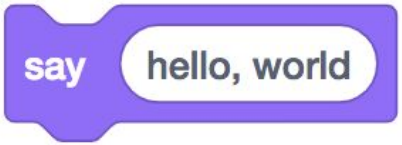

source code ⟶
⟶ machine code


source code
compiler
machine code

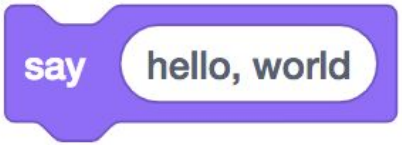
```
make hello

./hello
```

functions, arguments

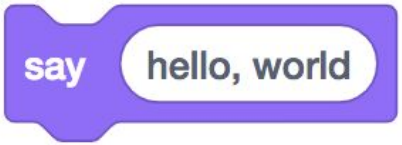

say
hello, world



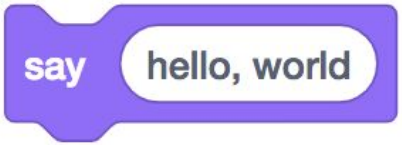

```
print (                )
```



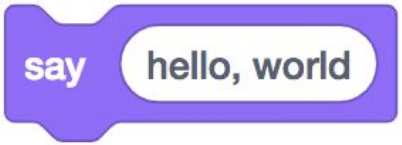

```
printf(                )
```



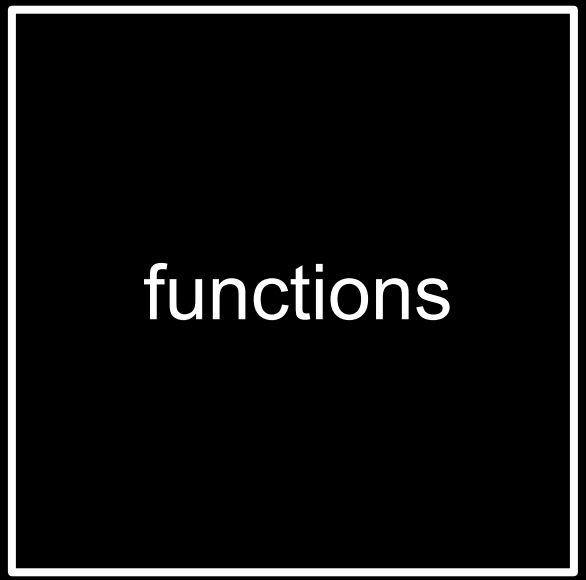

```
printf( hello, world )
```

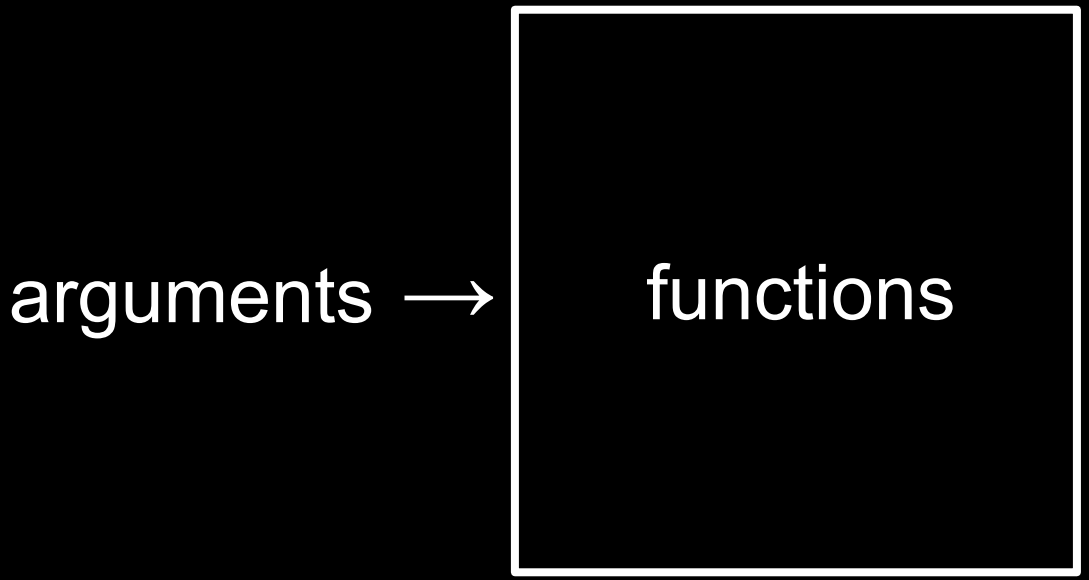


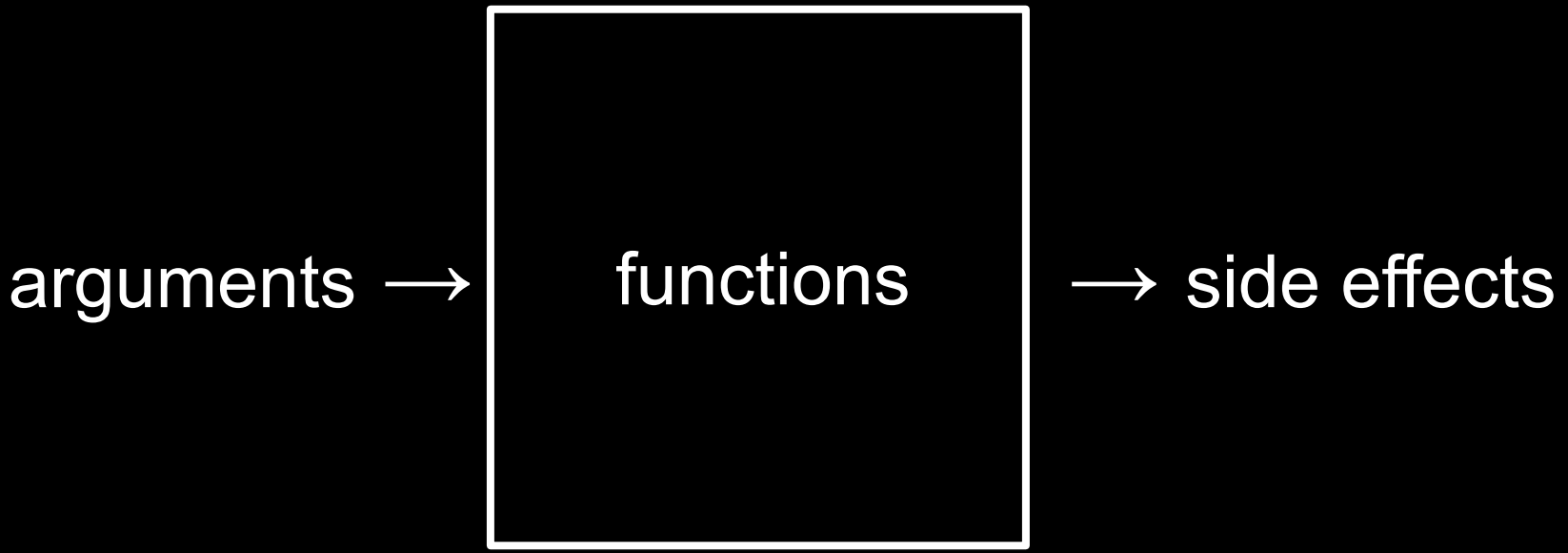

```
printf("hello, world")
```

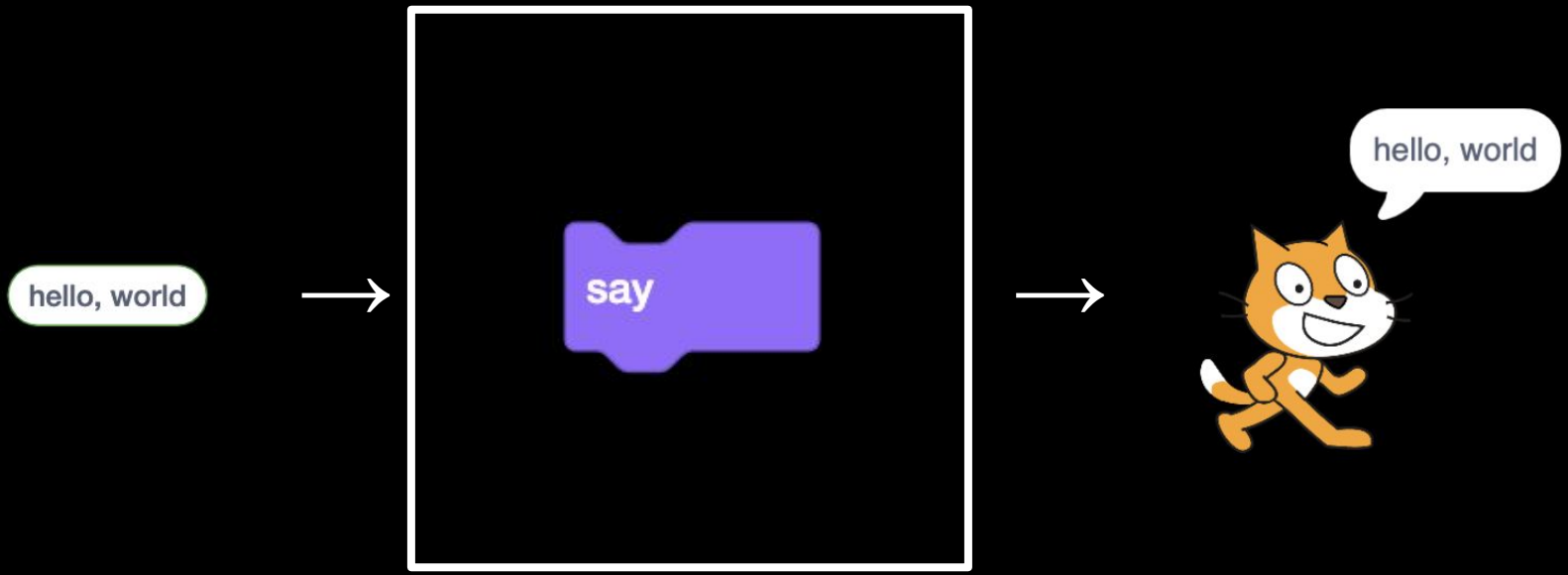



```
printf("hello, world");
```


functions


arguments ⟶
functions


arguments
functions
side effects


hello, world
say
hello, world

return values, variables

ask
What's your name?
and wait
answer

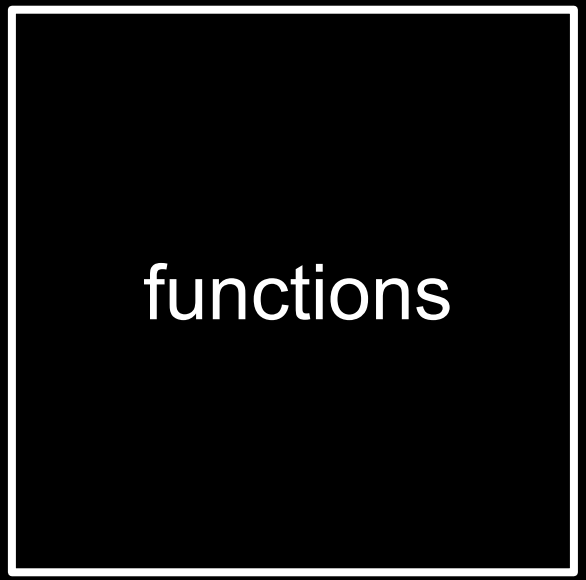
```
get_string(                    )
```

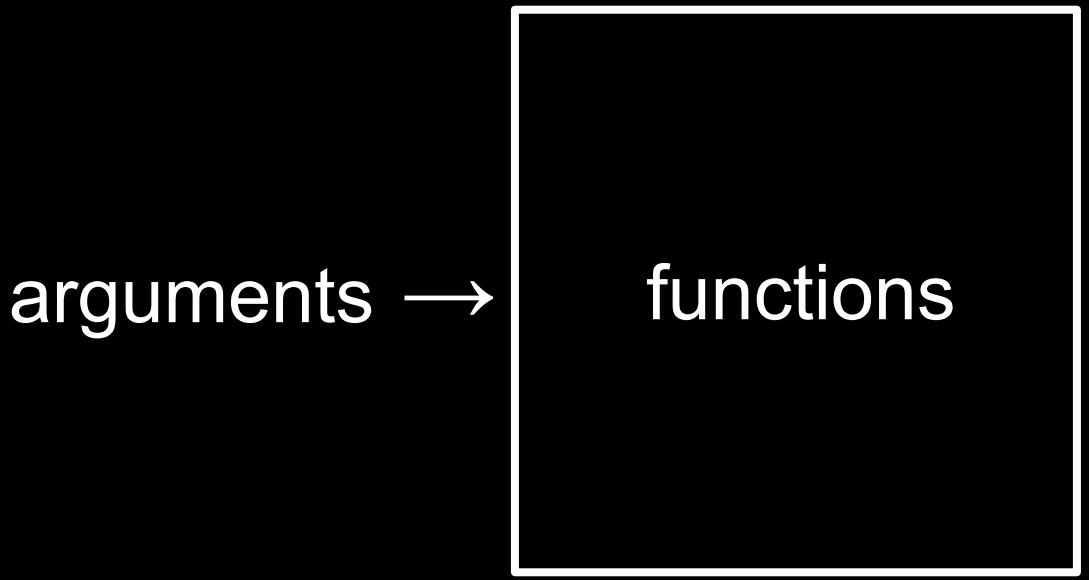
```
get_string("What's your name? ")
```

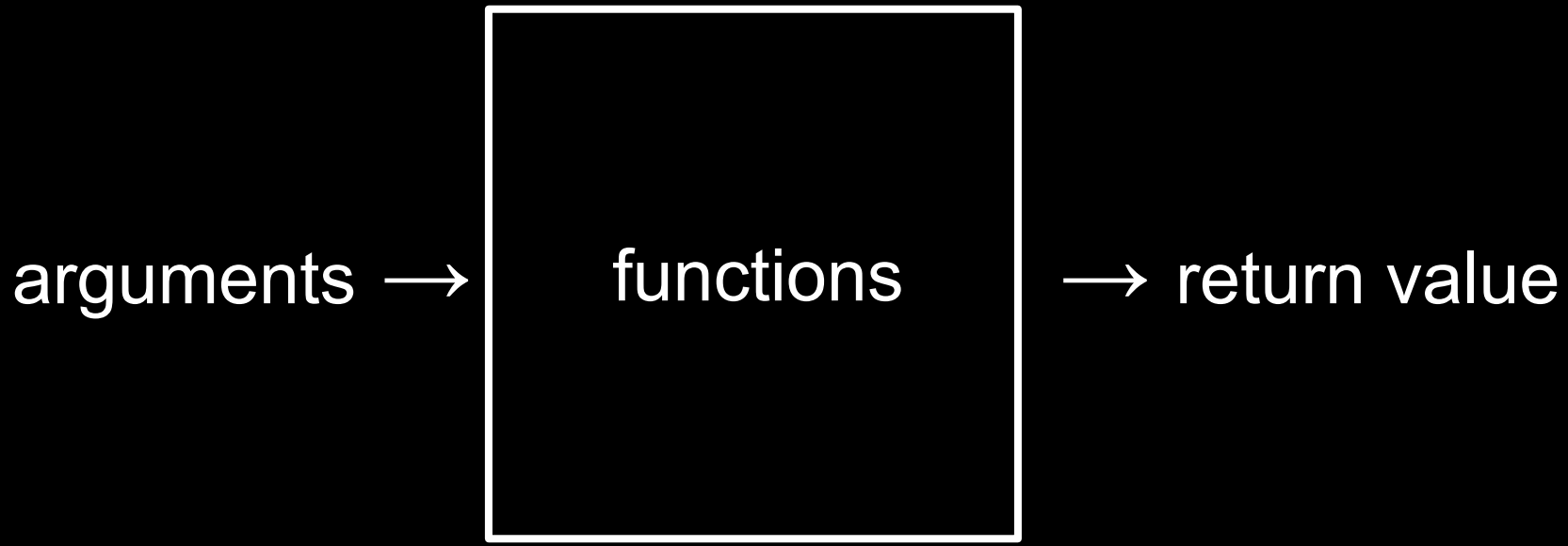
```
answer = get_string("What's your name? ")
```

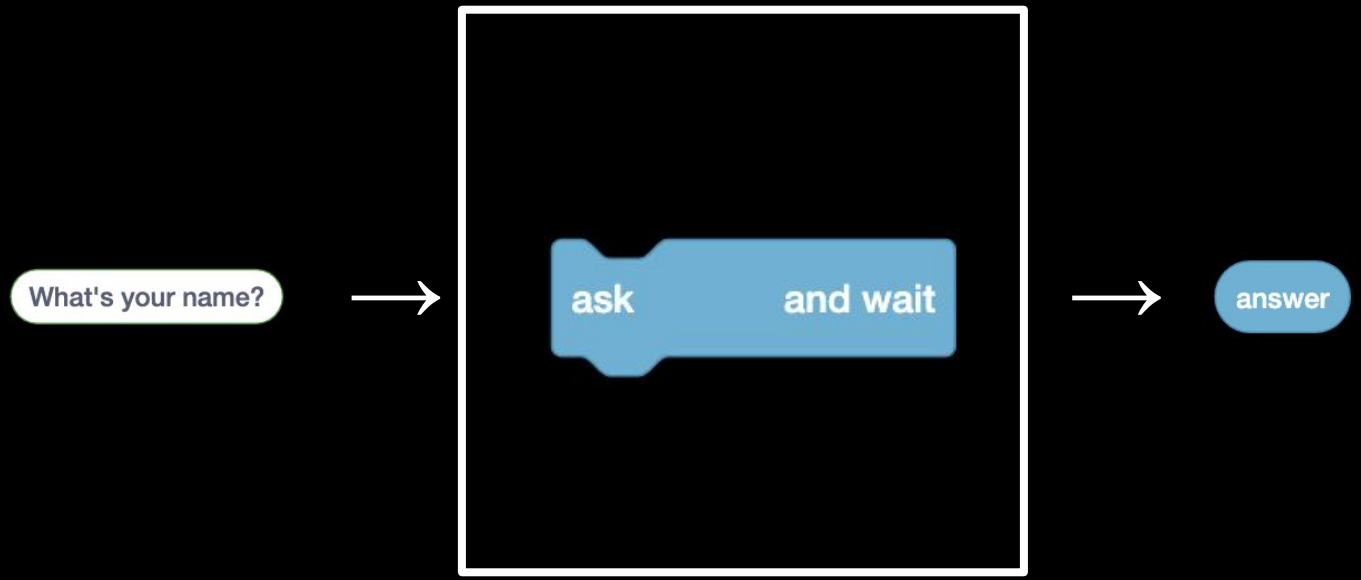
```
string answer = get_string("What's your name? ")
```

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


functions


arguments ⟶
functions


arguments
functions
return value


What's your name?
ask
and wait
answer

say
join
hello,
answer

```
printf(                    );
```

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

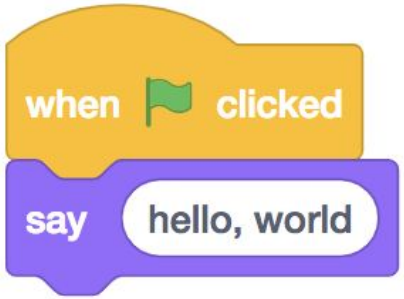
```
printf("hello, %s", answer);
```

main

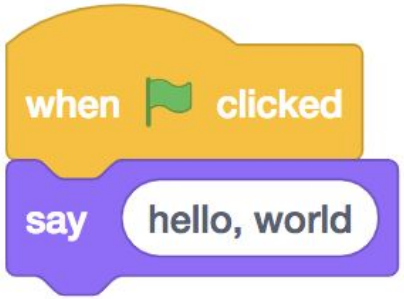
when clicked

```
int main(void)
{

}
```

header files




```
int main(void)
{
    printf("hello, world\n");
}
```




```c
#include <stdio.h>

int main(void)
{
    printf("hello, world\n");
}
```

```
cd
cp
ls
mkdir
mv
rm
rmdir
...
```

types

```
bool

char

double

float

int

long

string

...
```

```
get_char

get_double

get_float

get_int

get_long

get_string

...
```

format codes

%c

%f

%i

%li

%s

```
%c  char

%f  float, double

%i  int

%li long

%s  string
```

operators

+

-

*

/

%

```
+  addition

-  subtraction

*  multiplication

/  division

%  remainder
```

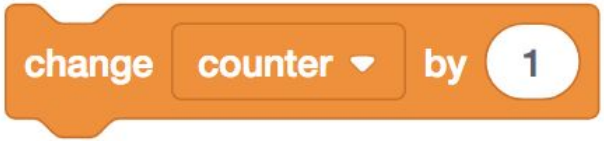
variables, syntactic sugar

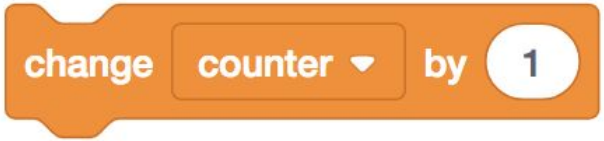
set
counter
to
0

```
counter = 0
```

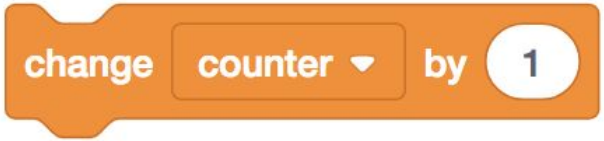
```
int counter = 0
```

```
int counter = 0;
```

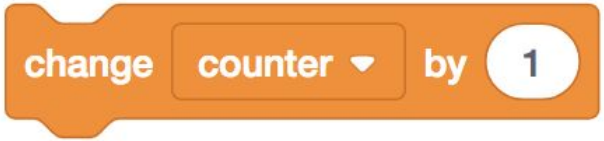

change
counter
by
1



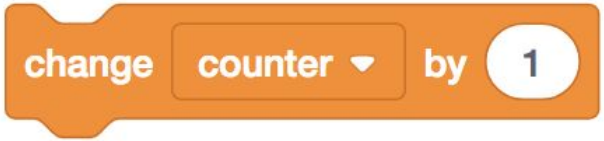

```
counter = counter + 1
```




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




```
counter += 1;
```



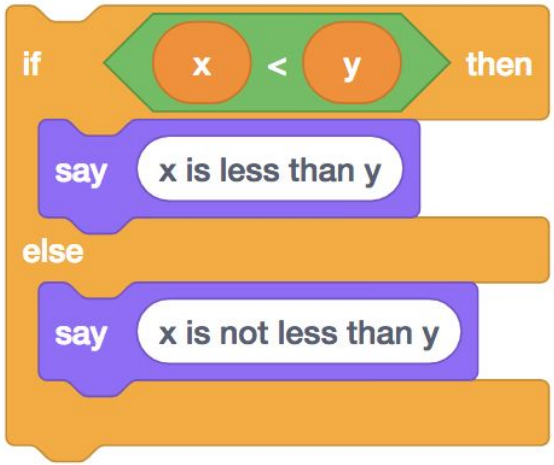

```
counter++;
```

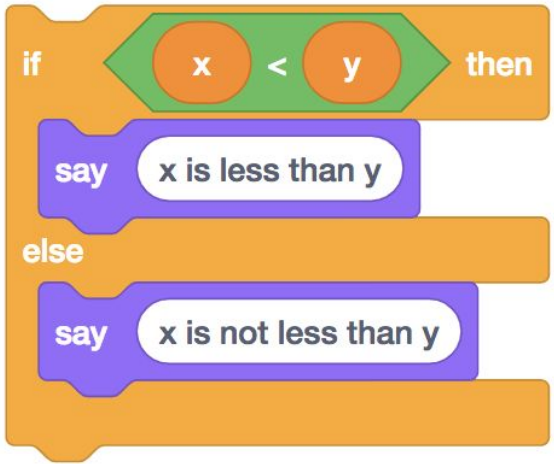
conditions

if
x
<
y
then
say
x is less than y

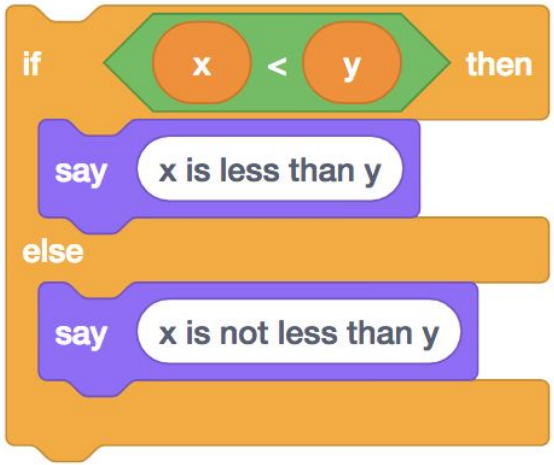
```
if (x < y)
{

}
```

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

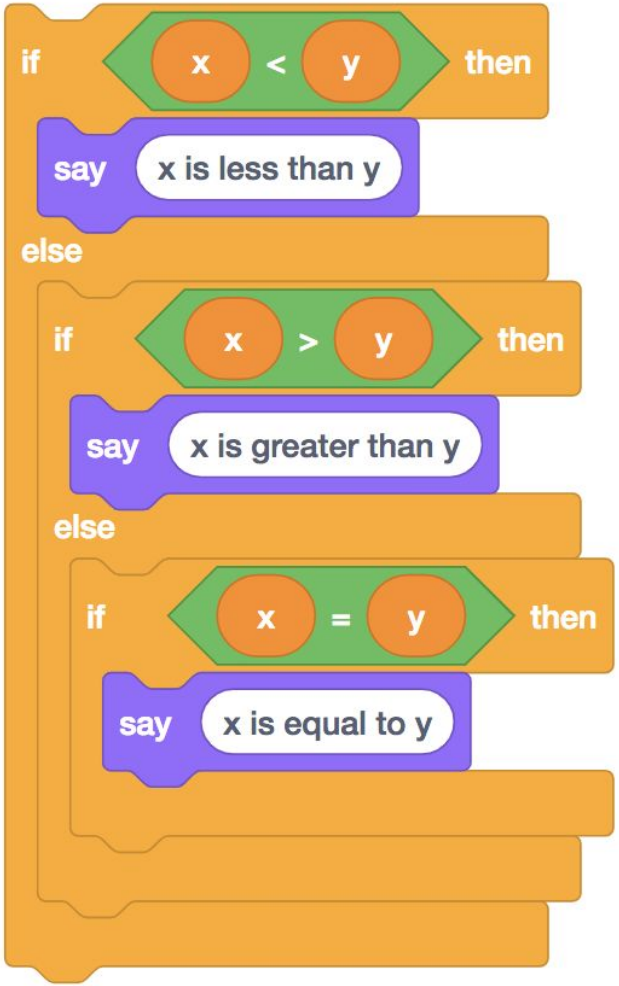

if
x
<
y
then
say
x is less than y
else
say
x is not less than y

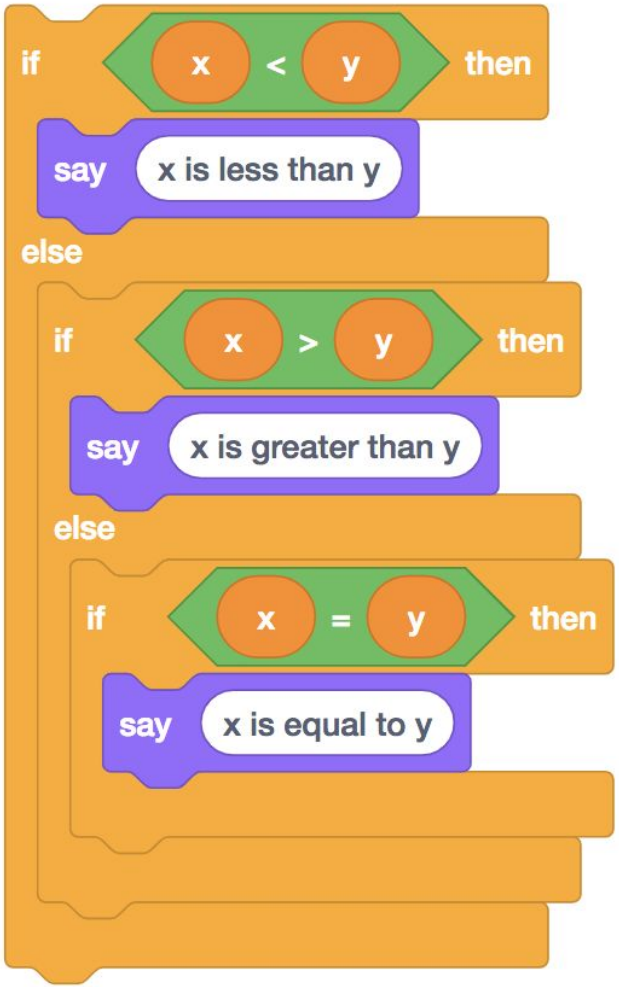



```
if (x < y)
{

}
else
{

}
```

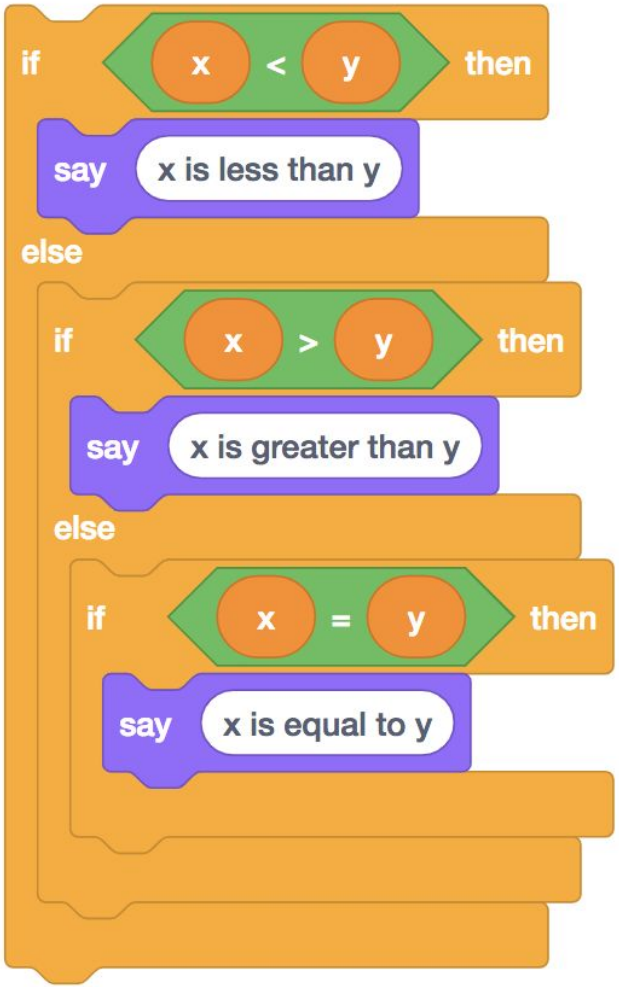



```
if (x < y)
{
    printf("x is less than y\n");
}
else
{
    printf("x is not less than y\n");
}
```

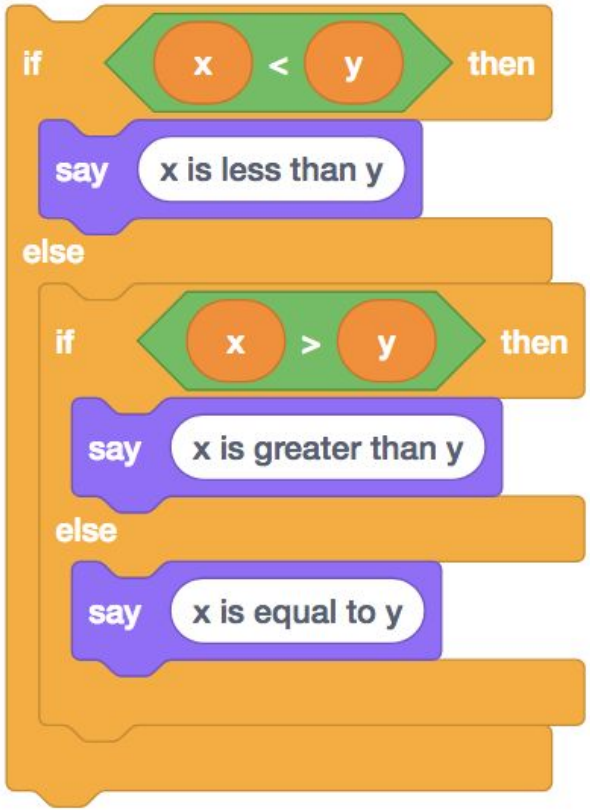

if x < y then
say x is less than y
else
if x > y then
say x is greater than y
else
if x = y then
say x is equal to y




```
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");
}
```

loops

forever
say
meow

```
while (true)
{

}
```

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

repeat
3
say
meow

```
int counter = 0;
while (counter < 3)
{


}
```

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

}
```

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

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

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

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

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

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

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

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

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

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

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

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

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

repeat
3
say
meow

```
for (int i = 0; i < 3; i++)
{

}
```

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

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

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

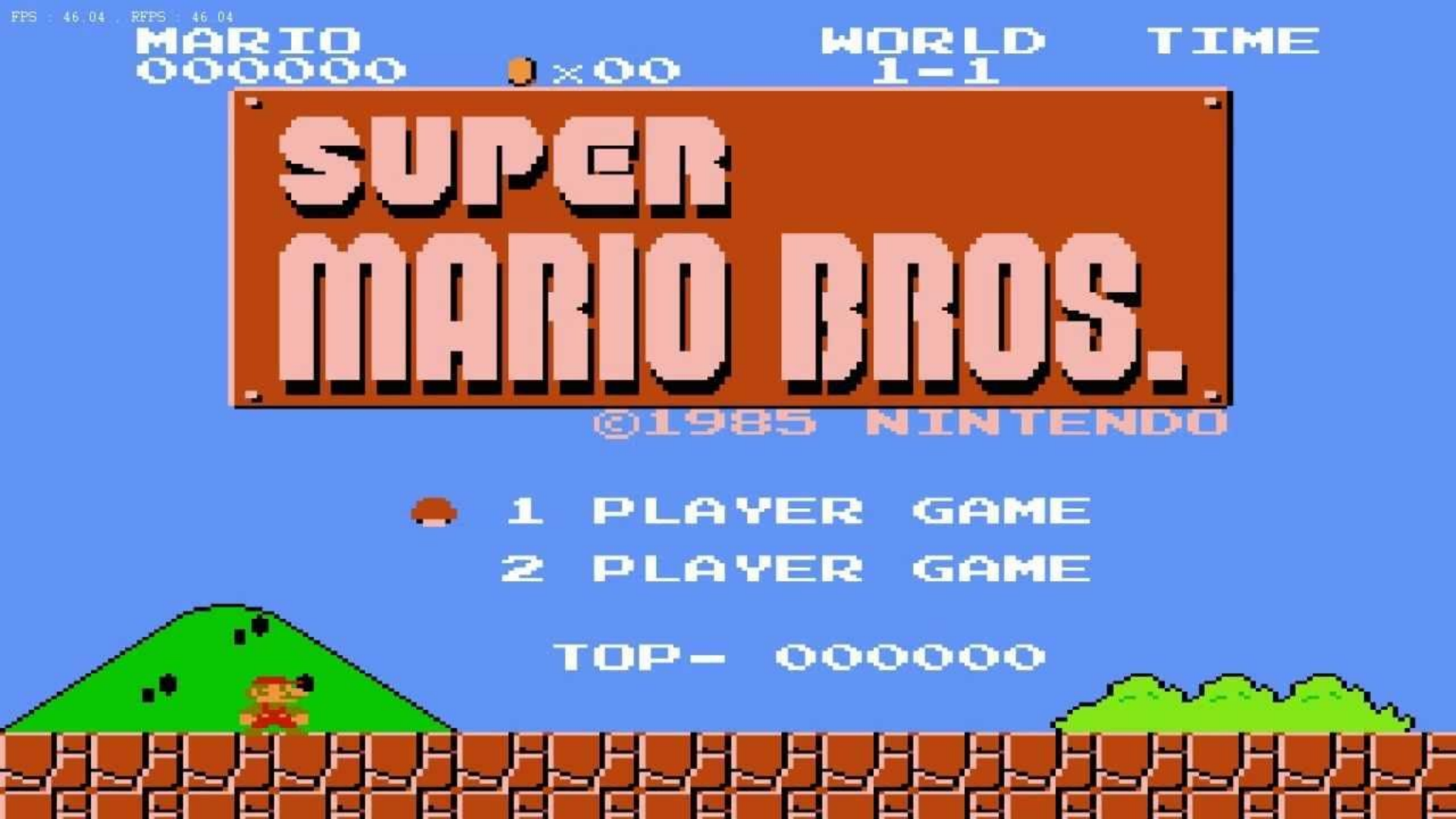
```
for (int i = 0; i < 3; i++)
{
    printf("meow\n");
}
```

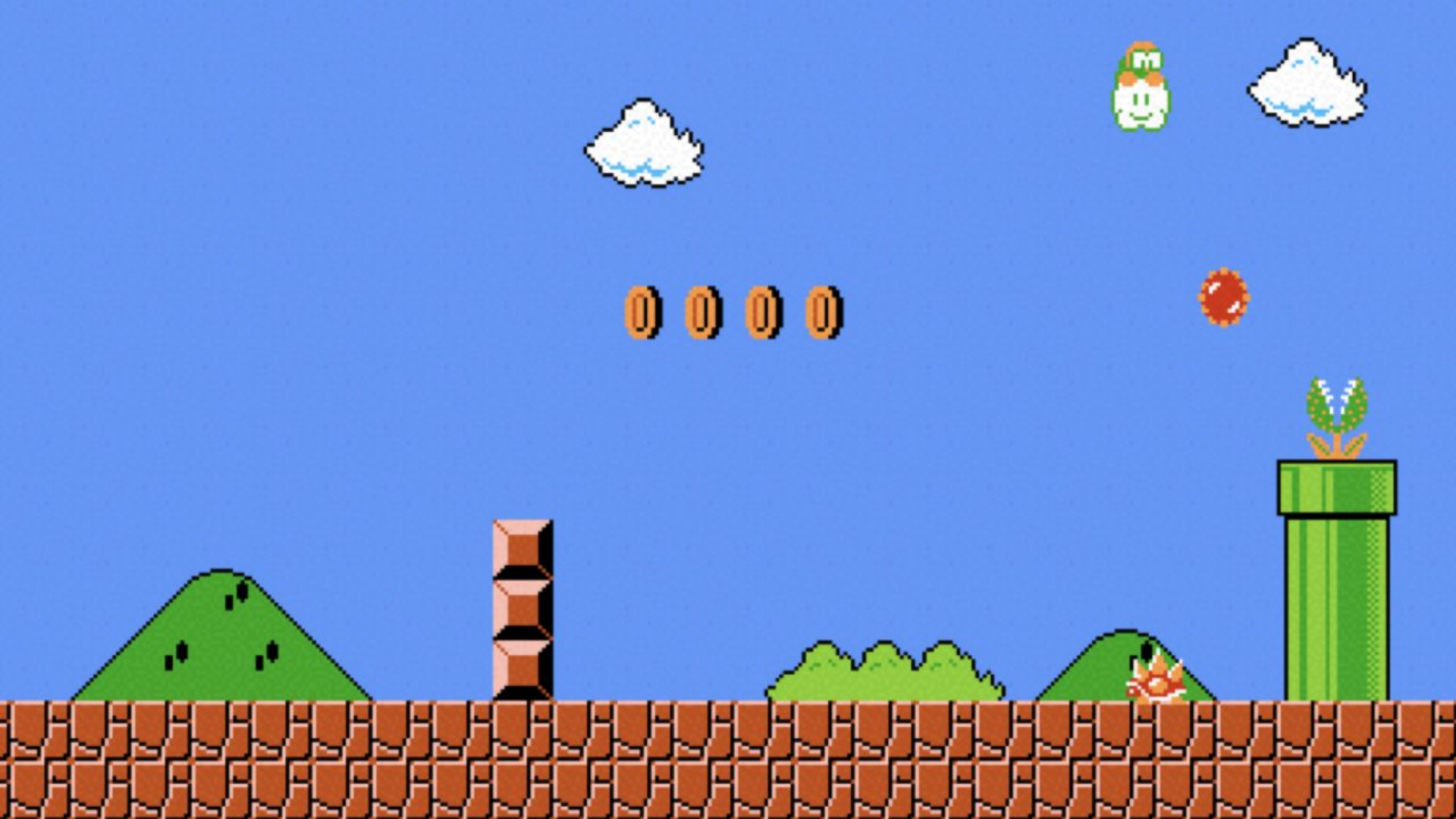
```
for (int i = 0; i < 3; i++)
{
    printf("meow\n");
}
```

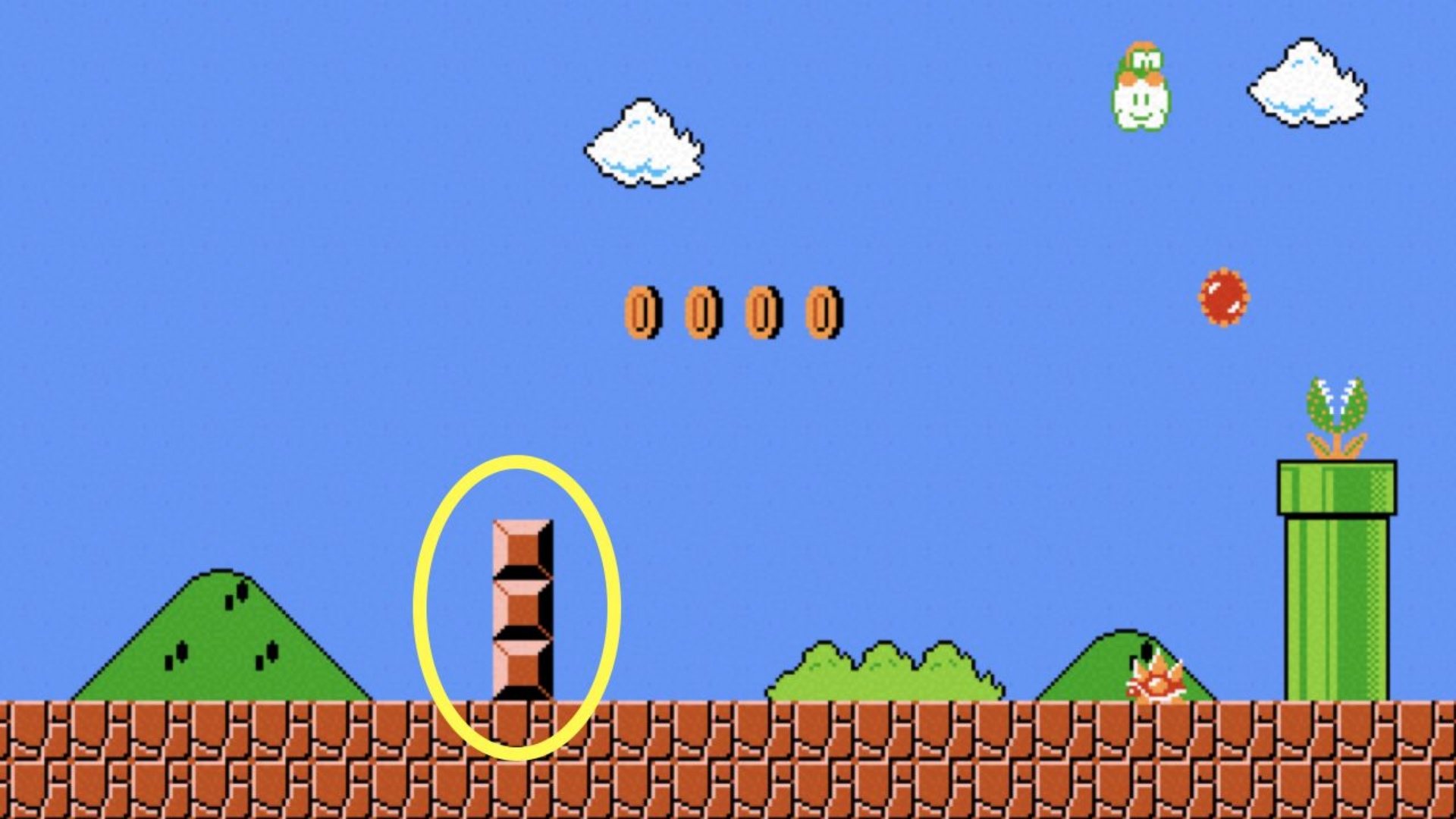
```c
for (int i = 0; i < 3; i++)
{
    printf("meow\n");
}
```

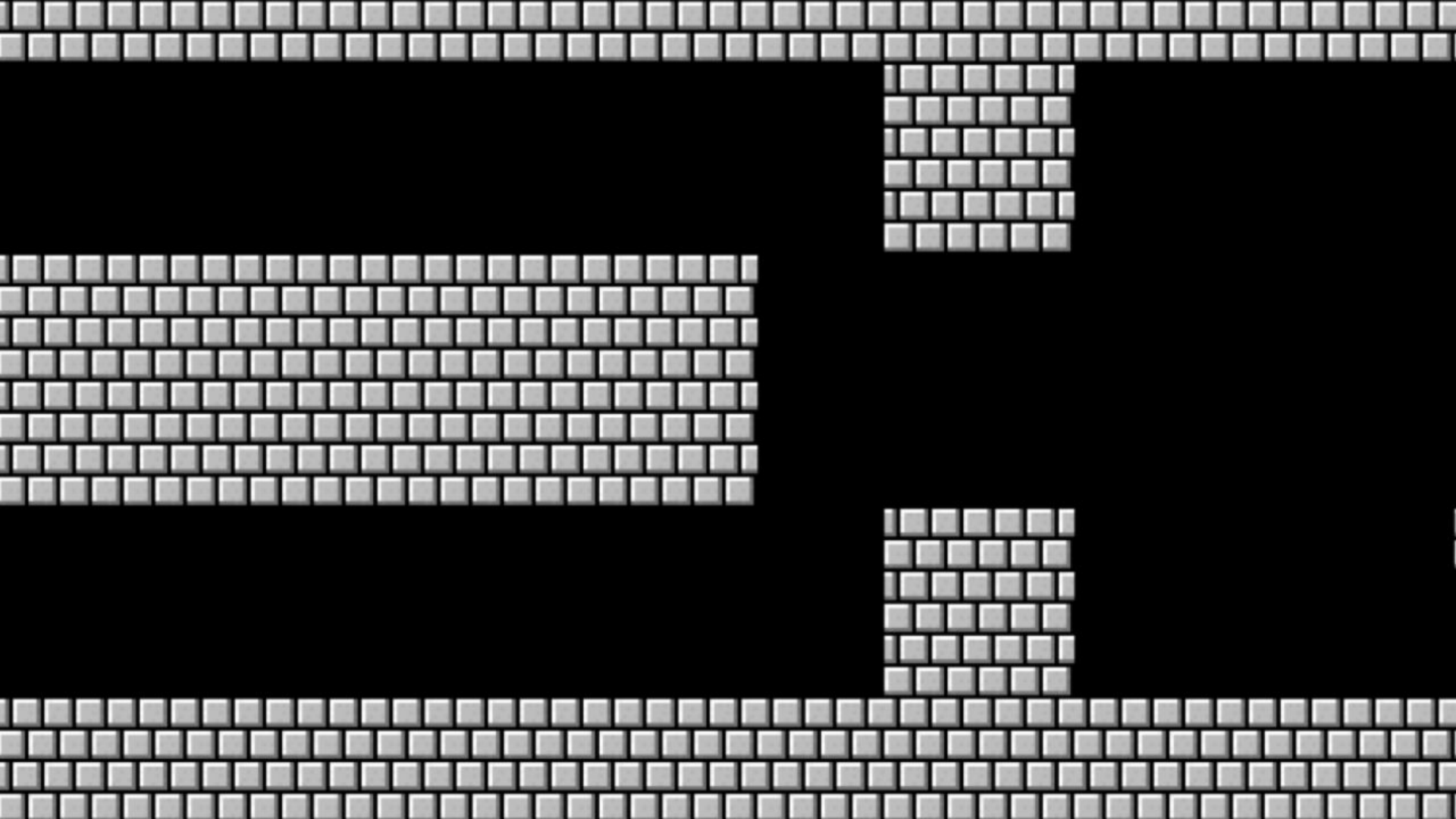
```
for (int i = 0; i < 3; i++)
{
    printf("meow\n");
}
```

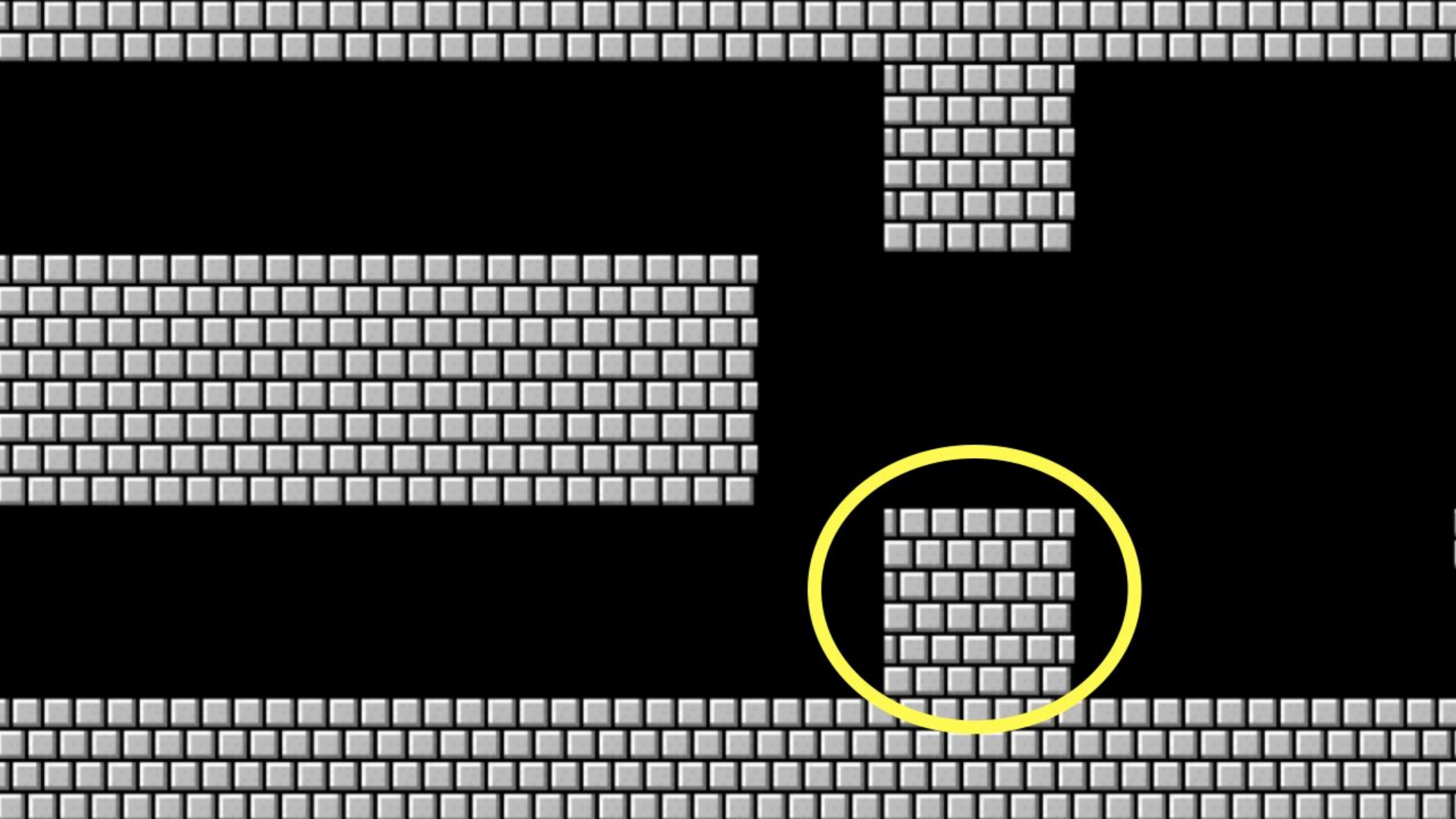
FPS : 46.04 , RFPS : 46.04











crucial
by Micron
8BBI2
D9HXT
8BBI2
D9HXT
4G85
4G85
4G85
47J6

amazon.com

floating-point imprecision

integer overflow

000

001

010

011

100

101

110

111

1000

000

1 January 2000

1999

1999

1900

19 January 2038

2147483647

01111111111111111111111111111111

01111111111111111111111111111111

1

0111111111111111111111111111111110

1

01111111111111111111111111111100

1

01111111111111111111111111111000

01111111111111111111111110000

01111111111111111111111111100000

01111111111111111111111111000000

01111111111111111111111110000000

01111111111111111111111100000000

01111111111111111111111000000000

01111111111111111111110000000000

01111111111111111111000000000000

01111111111111111110000000000000

01111111111111111100000000000000

01111111111111111000000000000000

01111111111111111000000000000000

01111111111111110000000000000000

01111111111111100000000000000000

01111111111110000000000000000000

01111111111100000000000000000000

01111111111000000000000000000000

01111111110000000000000000000000

01111111110000000000000000000000

01111111000000000000000000000000

01111111000000000000000000000000

01111110000000000000000000000000

01111100000000000000000000000000

01111000000000000000000000000000

01110000000000000000000000000000

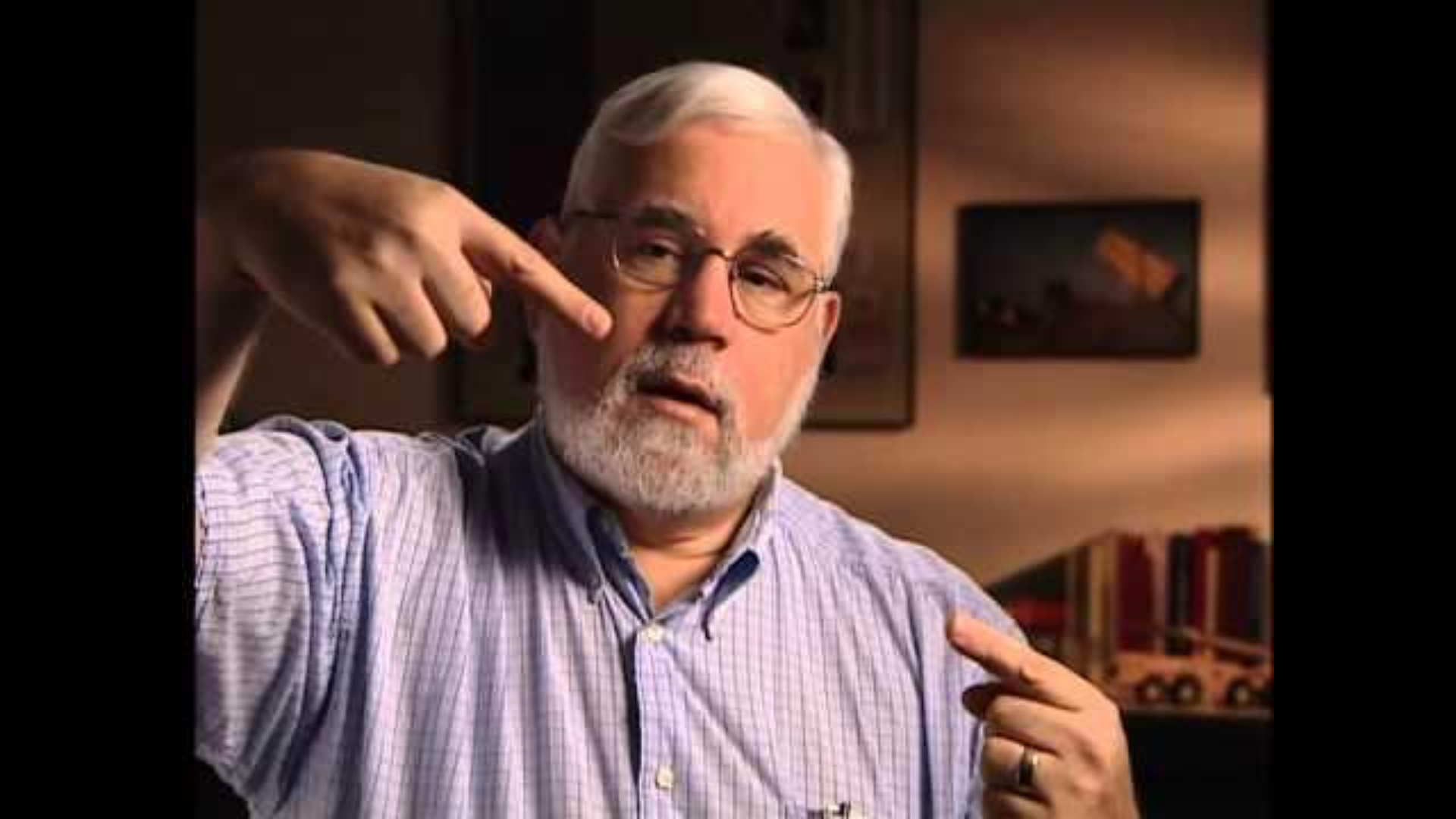
01100000000000000000000000000000

01000000000000000000000000000000

10000000000000000000000000000000

-2147483648

13 December 1901



This is CS50