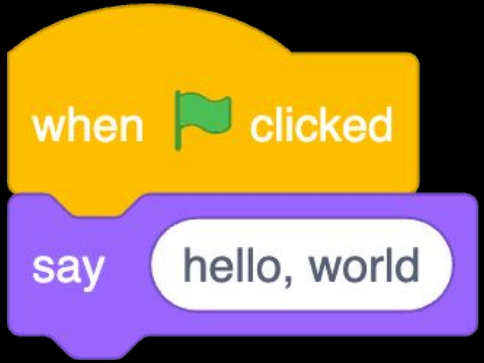
This is CS50

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



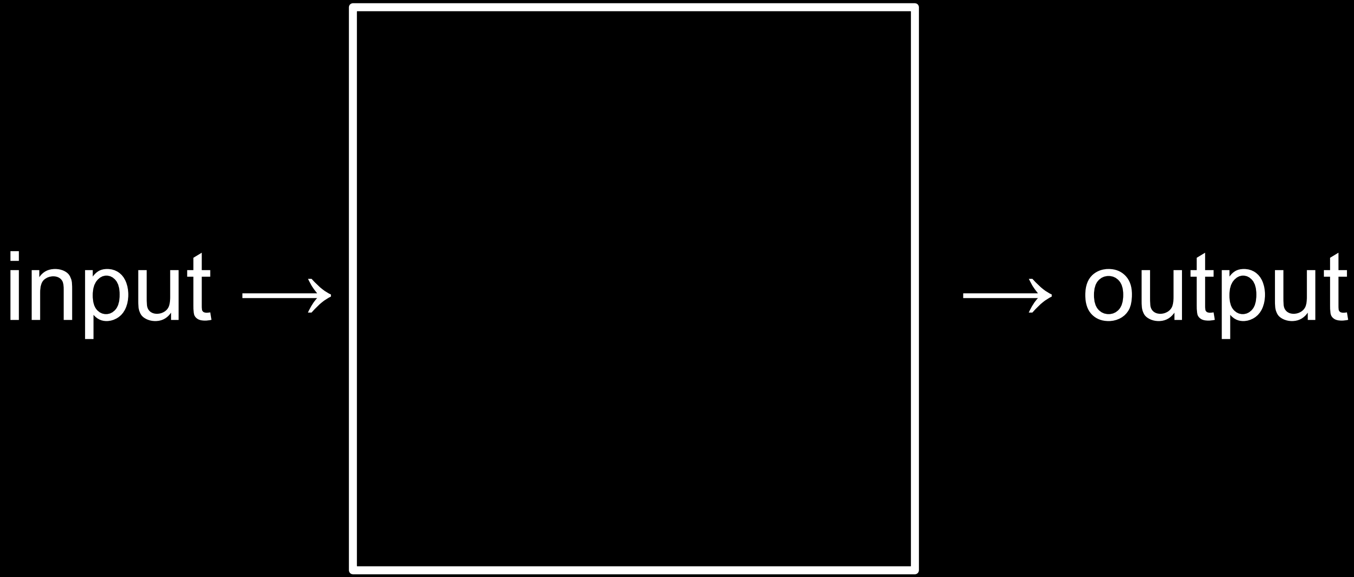
```
when ⚑ clicked
say hello, world
```

```c
#include <stdio.h>

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

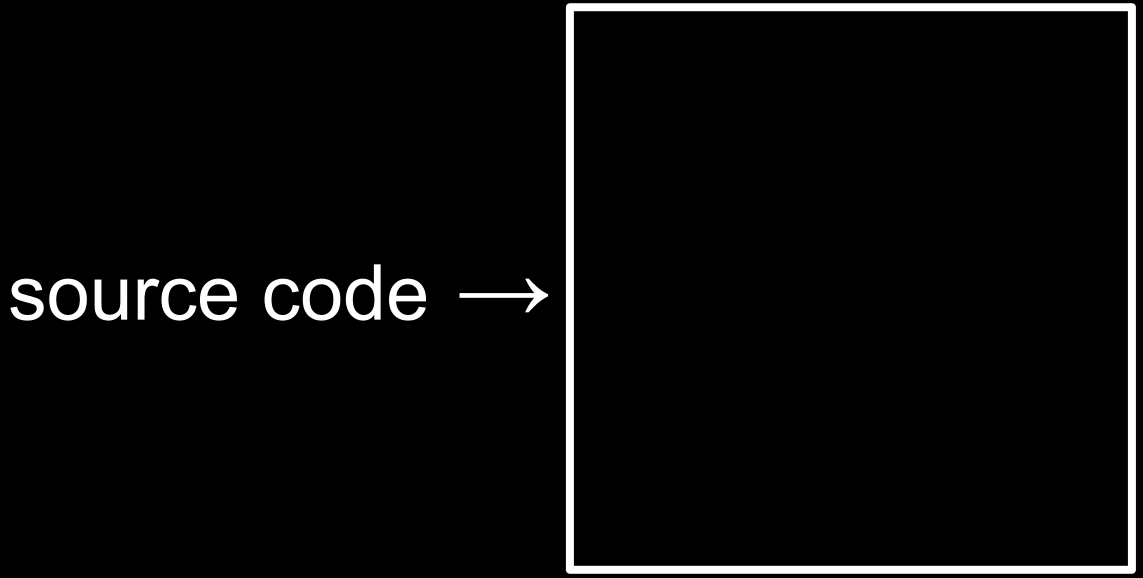
source code

```
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
...
```

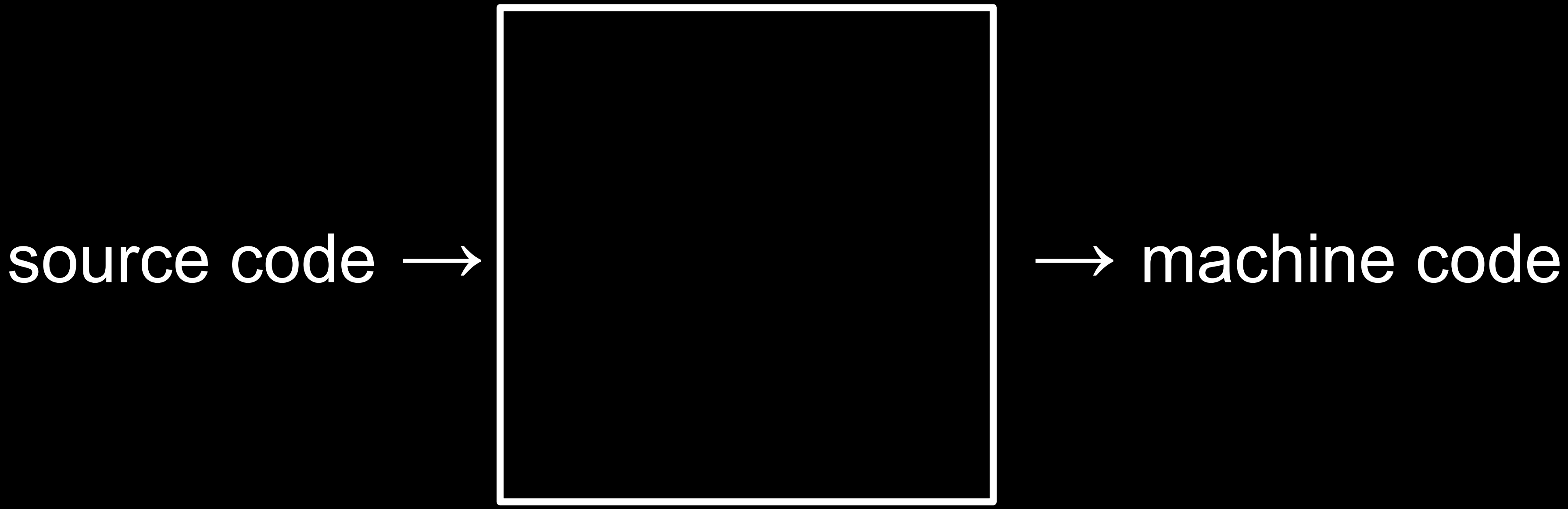
machine code



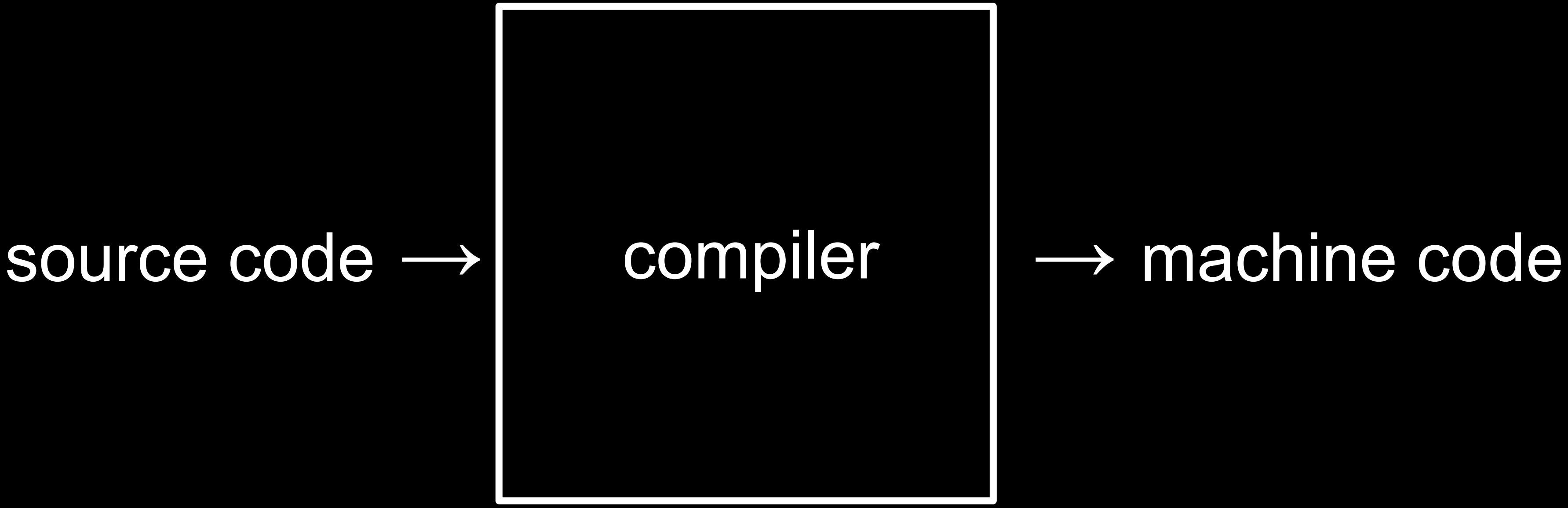
input → → output



source code →



source code → → machine code

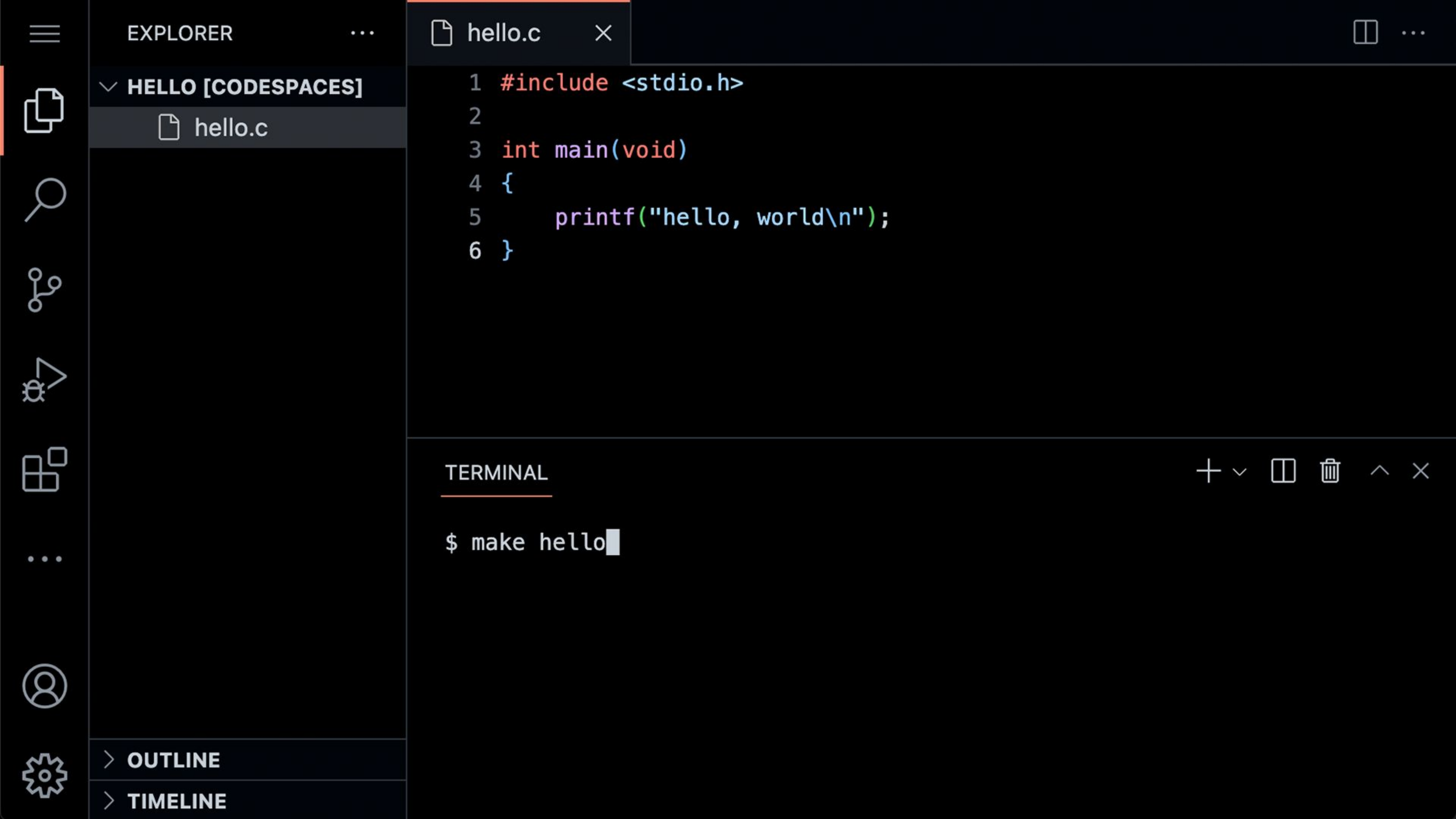


source code → compiler → machine code

correctness, design, style

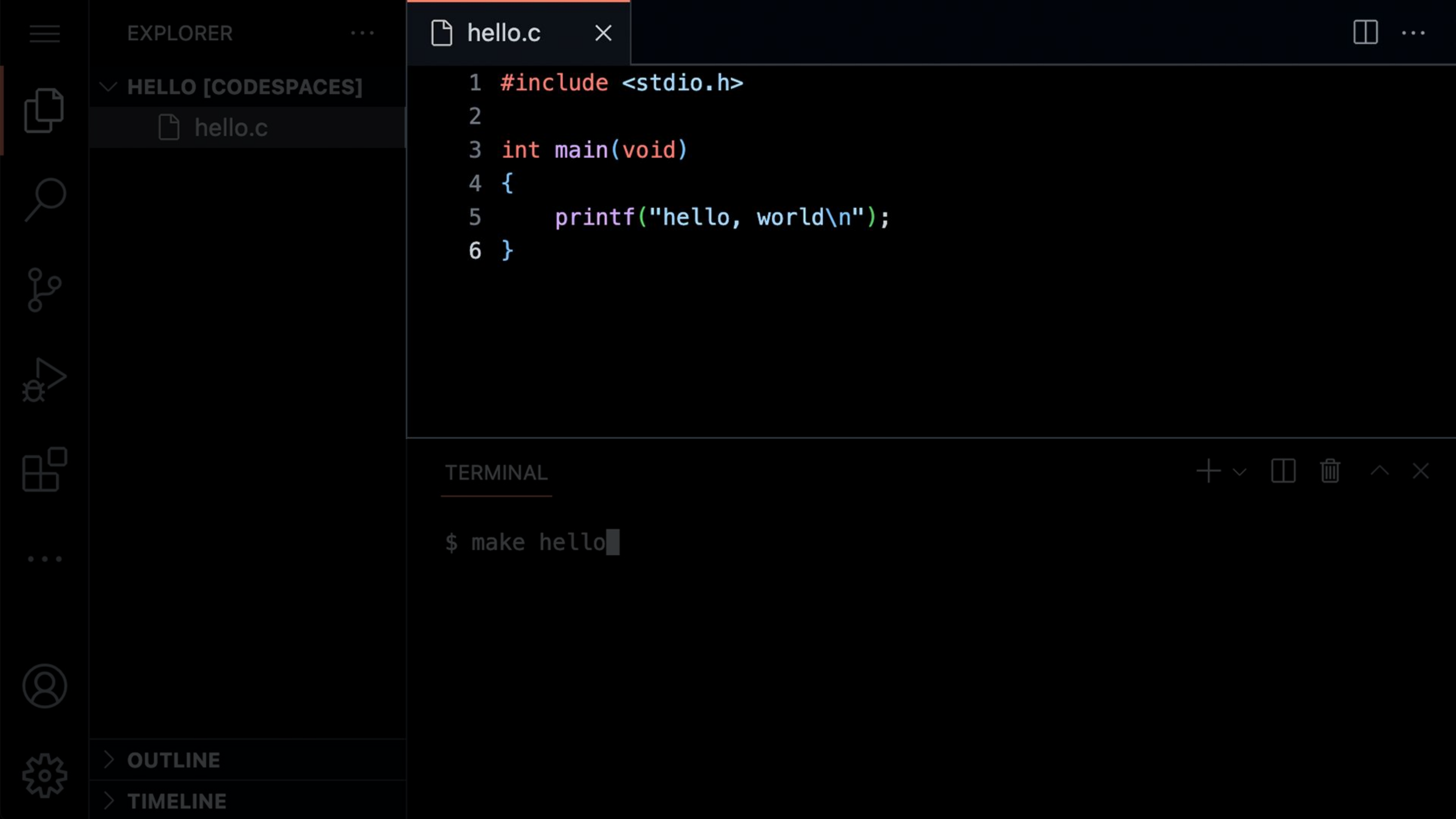
Visual Studio Code

code.cs50.io



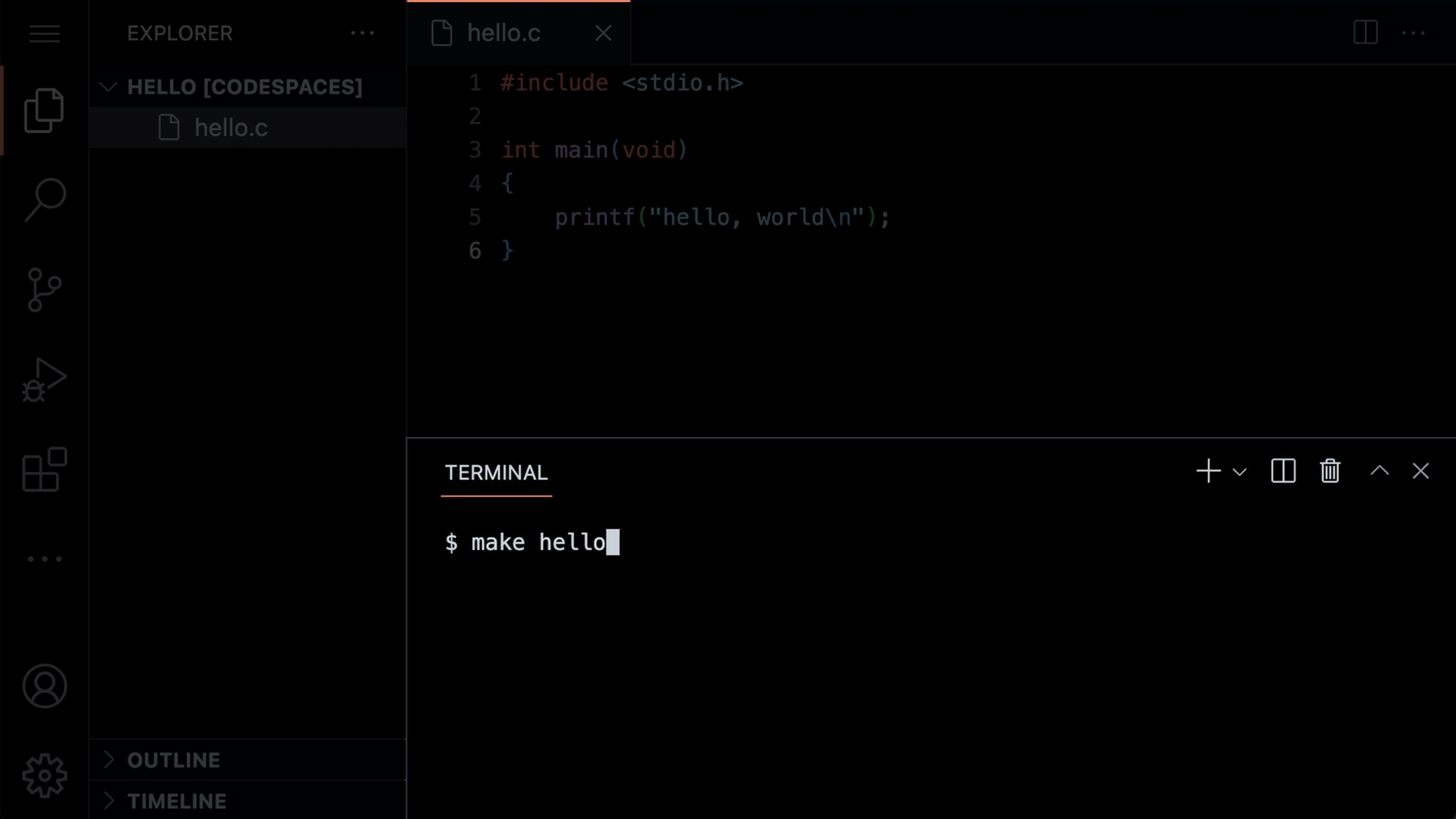
```c
#include <stdio.h>

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

```
$ make hello
```



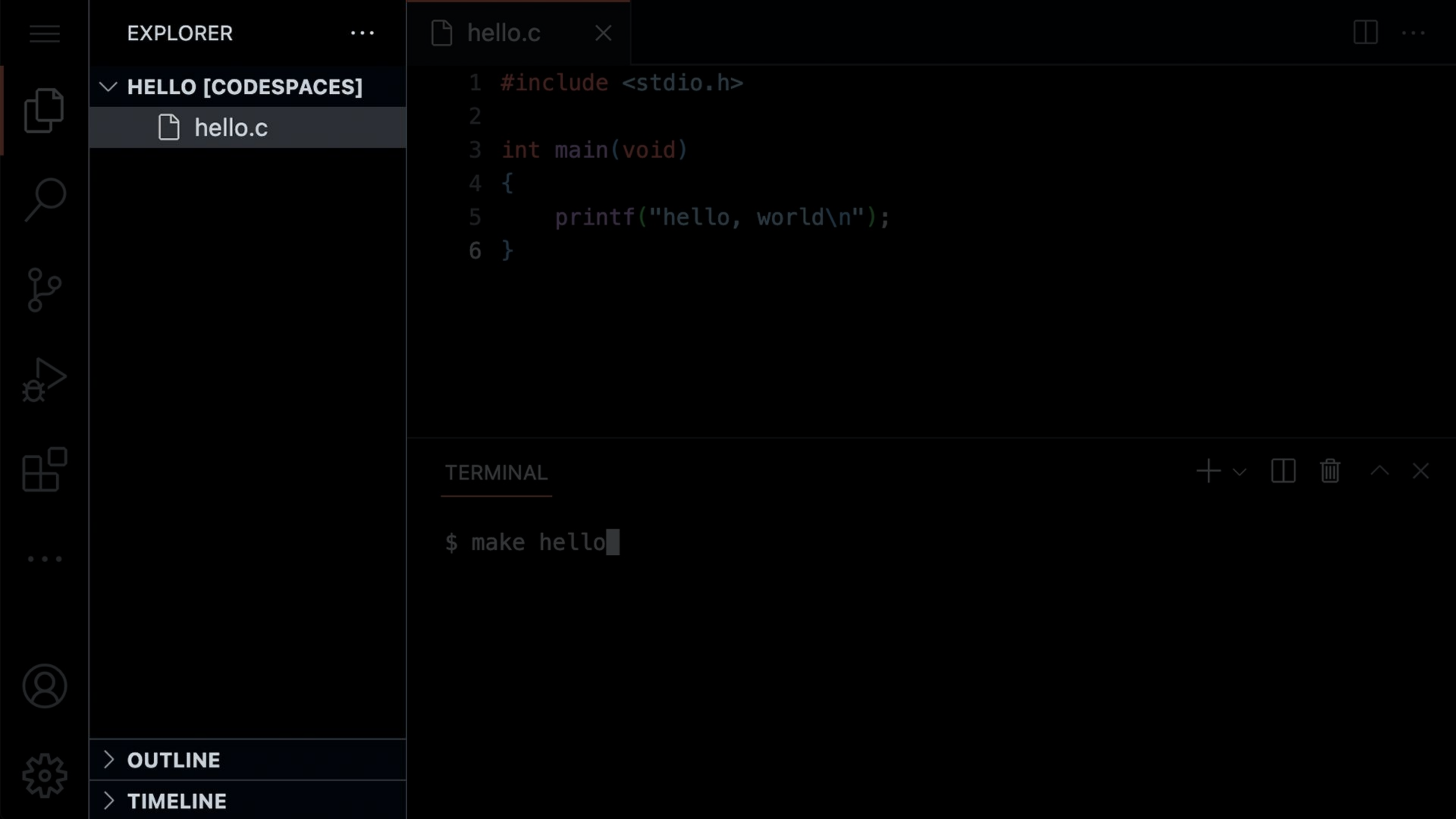
```c
#include <stdio.h>

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

```
$ make hello
```



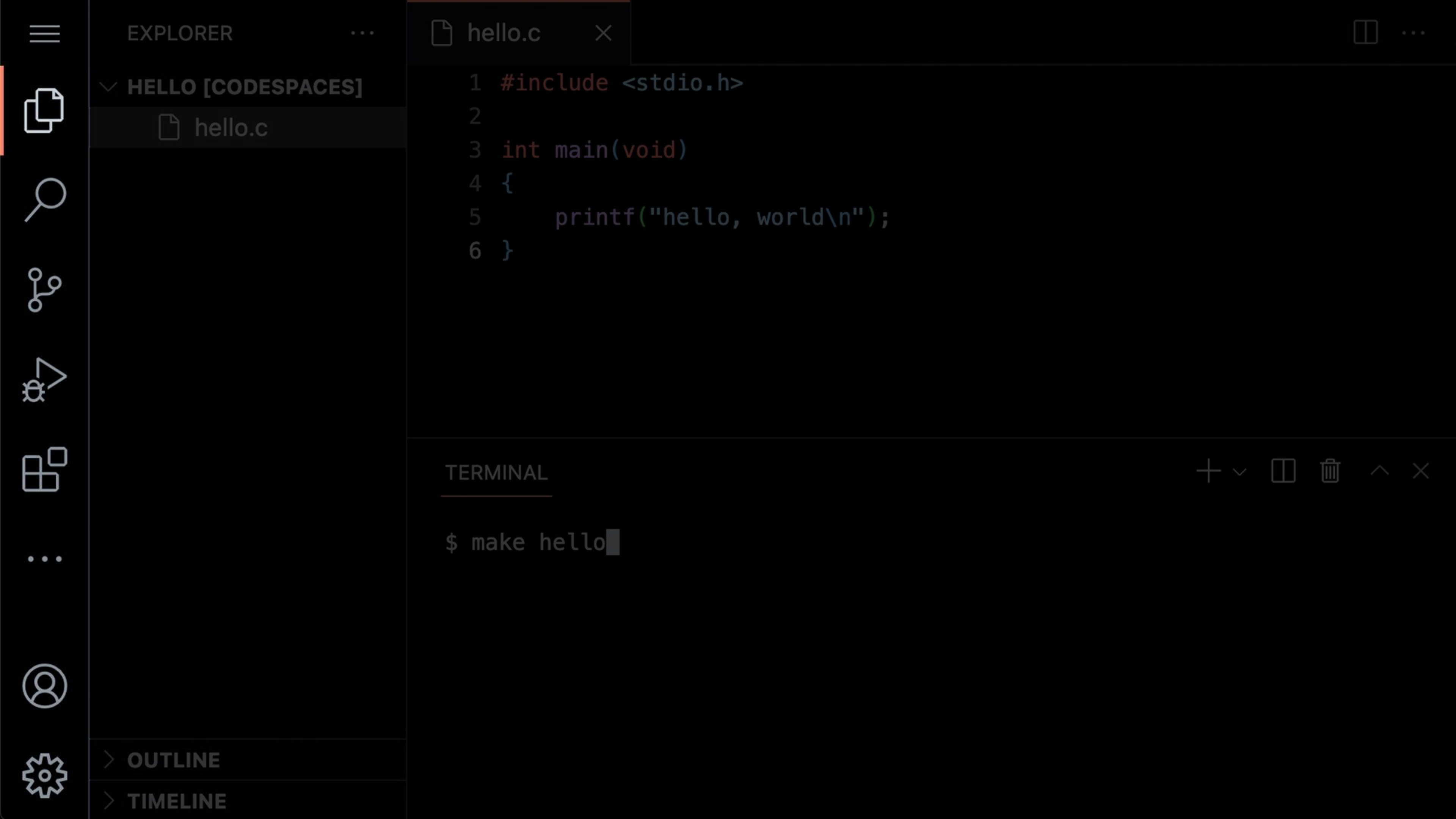
```c
#include <stdio.h>

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

```
$ make hello
```



```c
#include <stdio.h>

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

```
$ make hello
```



```c
#include <stdio.h>

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

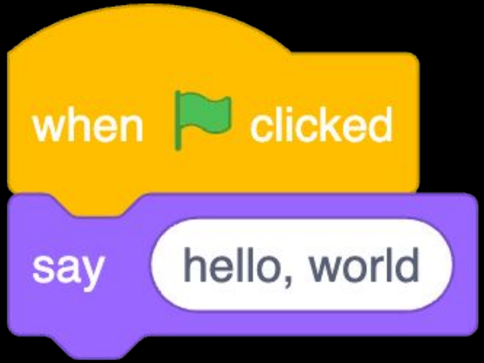
```
$ make hello
```

syntax highlighting

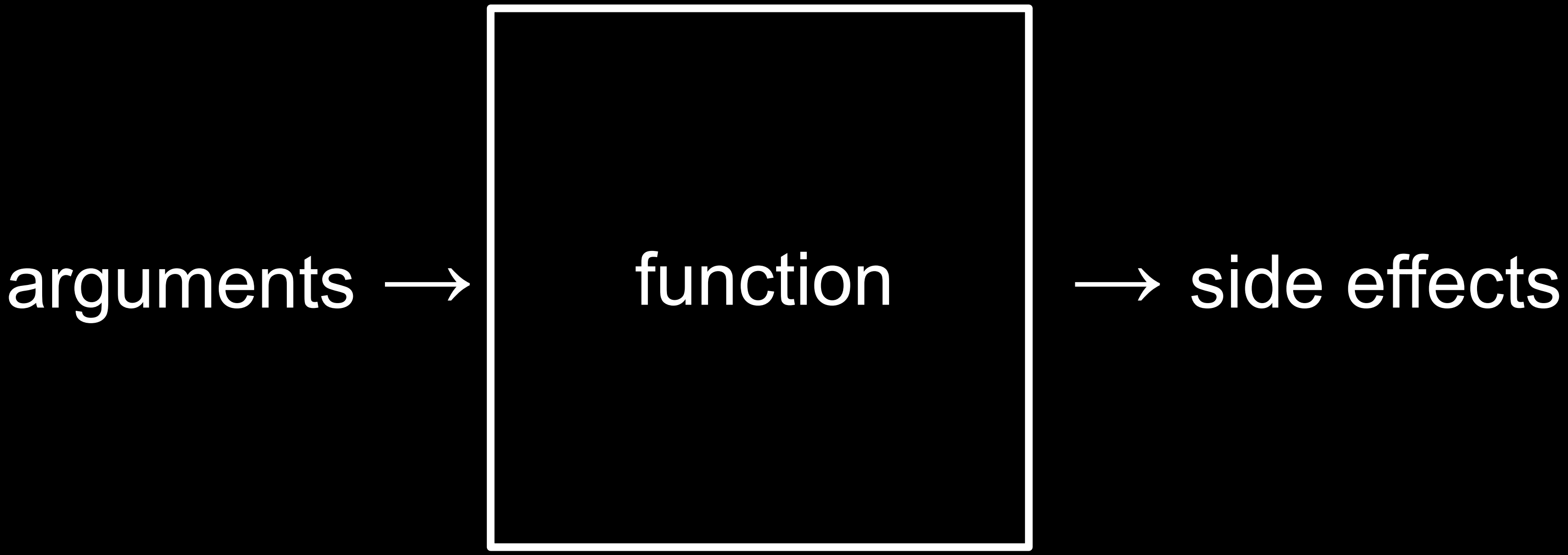
```
make hello

./hello
```

```
#include <stdio.h>

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

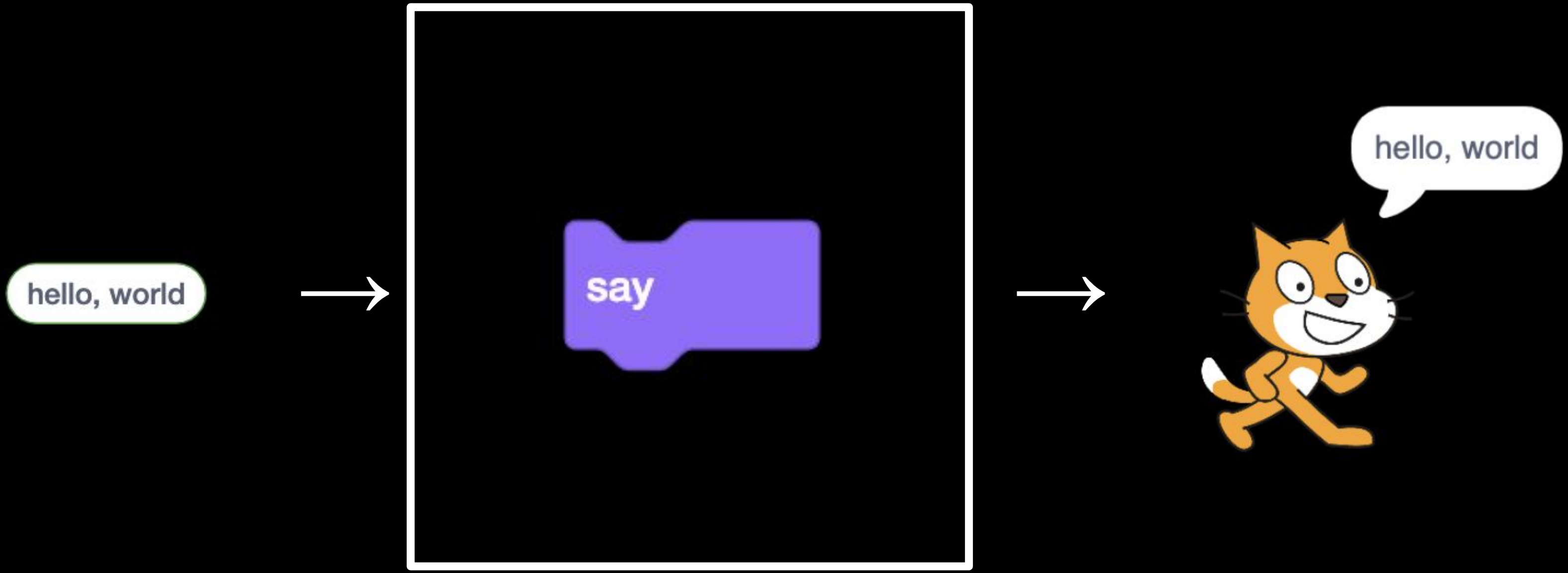
```c
#include <stdio.h>

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

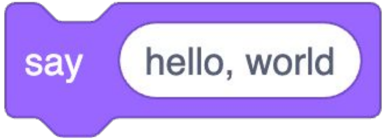


```
when ⚑ clicked
say (hello, world)
```

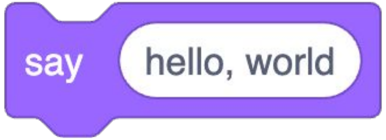


arguments → function → side effects


hello, world
say
hello, world

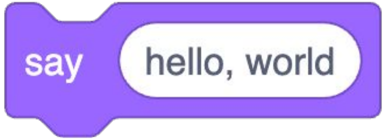


say hello, world

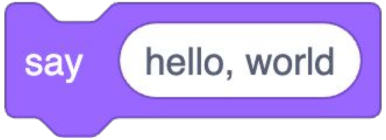


say hello, world

```
print (                    )
```

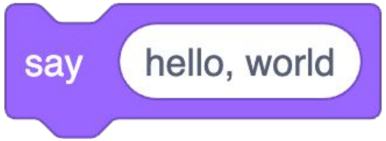


```
printf(                    )
```



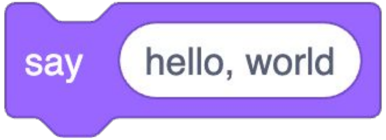
say hello, world

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



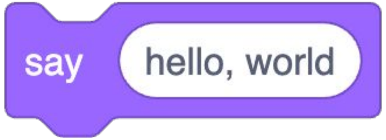
say hello, world

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



say hello, world

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



say hello, world

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

escape sequences

header files

libraries

Manual Pages

manual.cs50.io

stdio.h

manual.cs50.io/#stdio.h

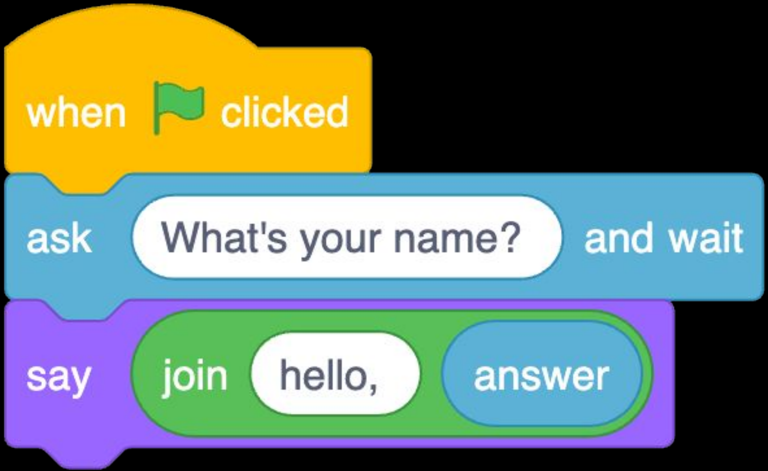
```
printf

...
```

manual.cs50.io/3/printf

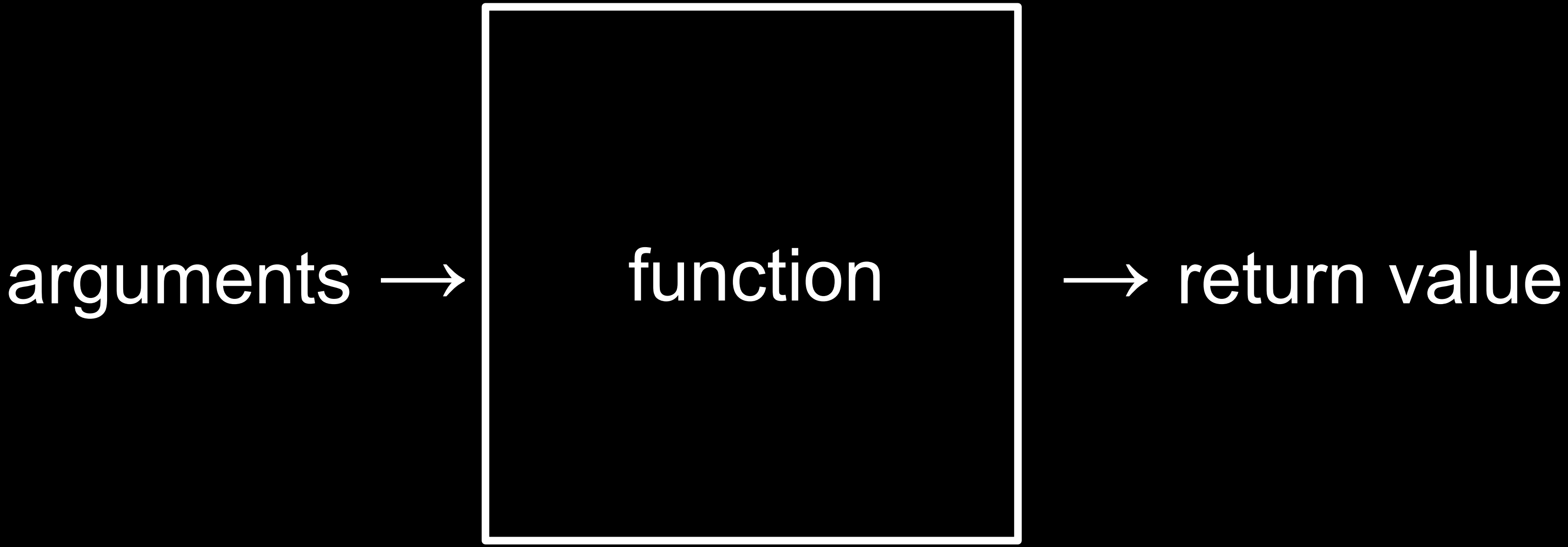
cs50.h

manual.cs50.io/#cs50.h

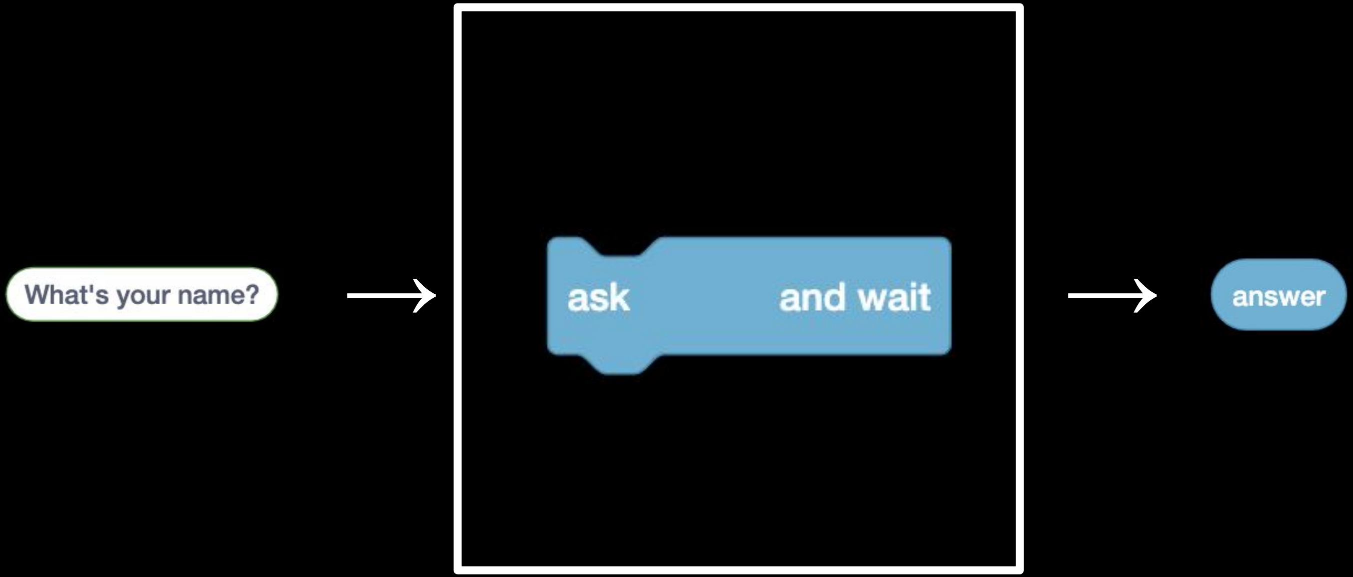
```
get_char

get_double

get_float

get_int

get_long

get_string

...
```

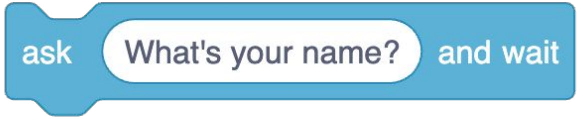


```
when 🏴 clicked
ask (What's your name?) and wait
say (join (hello,) (answer))
```

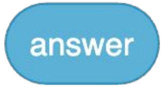


arguments → function → return value

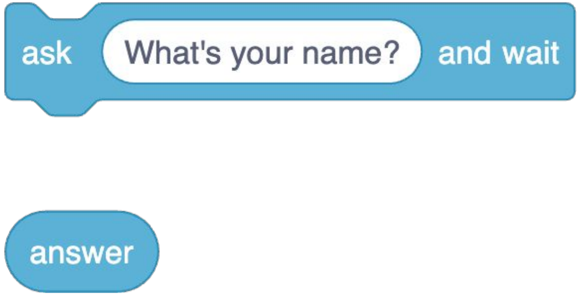

What's your name?
ask and wait
answer



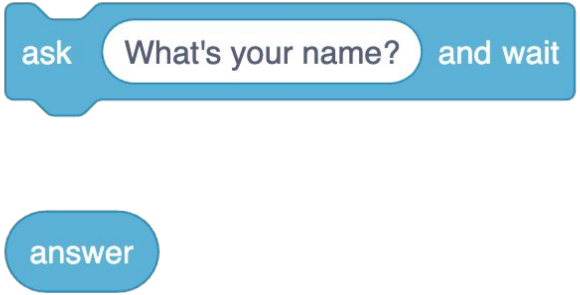
ask What's your name? and wait



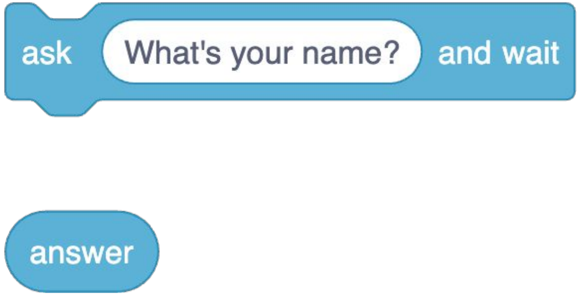
answer



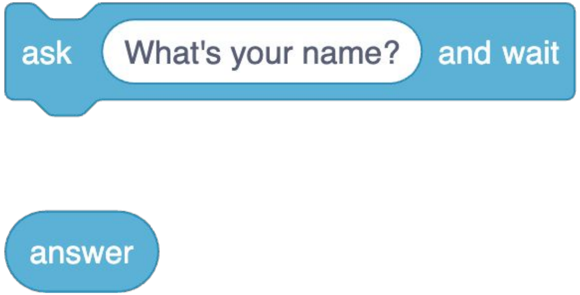
```
get_string(                    )
```



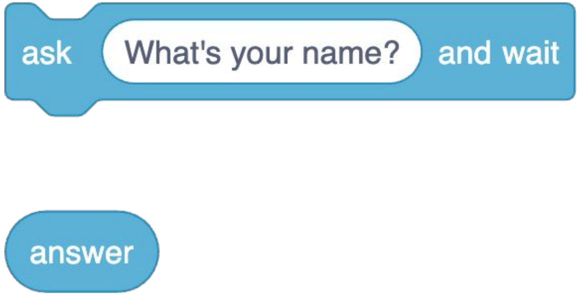
```
get_string( What's your name?  )
```



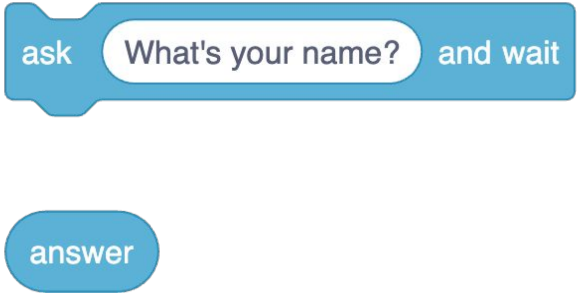
```
get_string("What's your name? ")
```



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



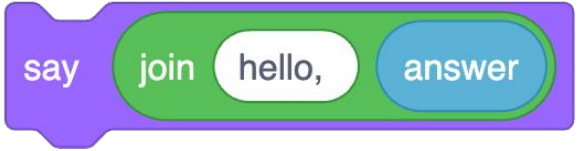
```
string answer = get_string("What's your name? ")
```



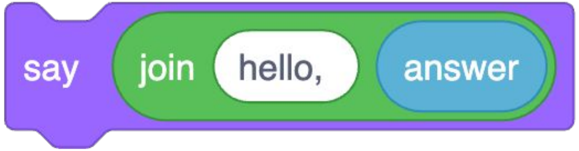
ask What's your name? and wait

answer

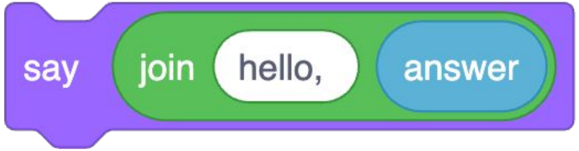
```
string answer = get_string("What's your name? ");
```



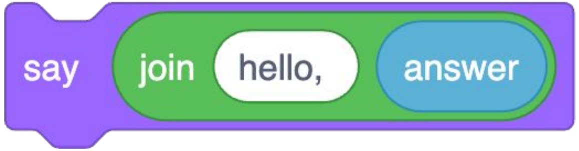
```
say join hello, answer
```



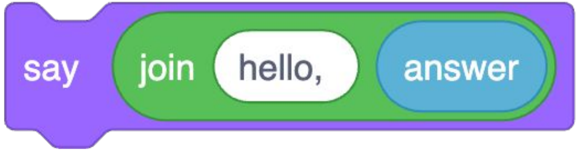
```
printf(                               );
```



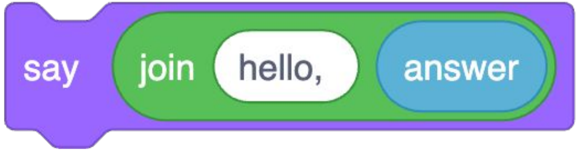
```
printf( hello, %s           );
```



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



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



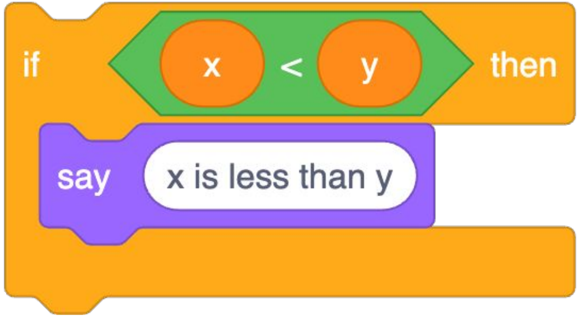
say join hello, answer

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

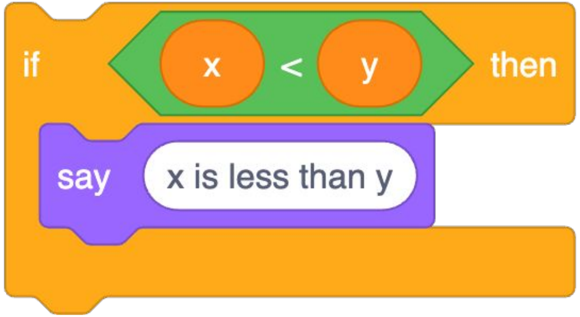
types

```
bool

char

double

float

int

long

string

...
```

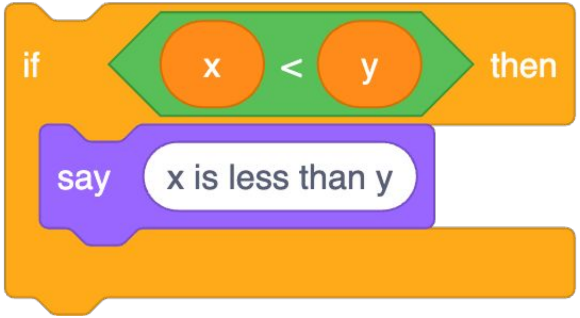
conditionals



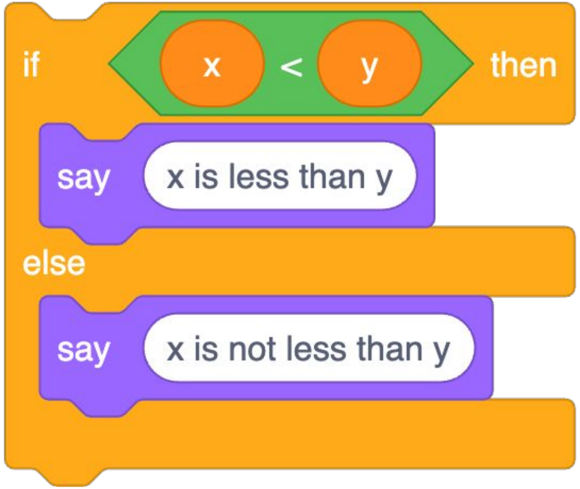
```
if x < y then
    say x is less than y
```



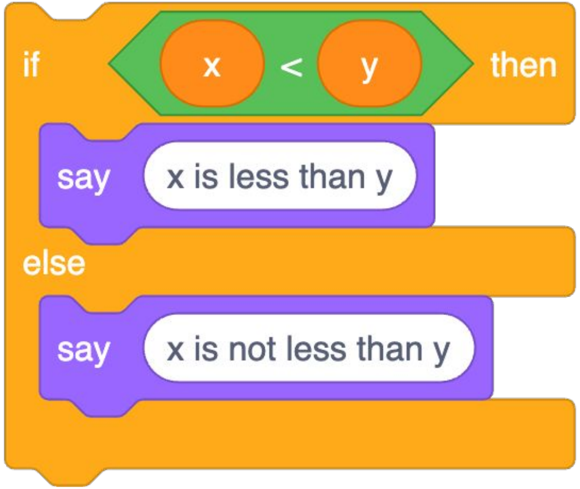
```
if (x < y)
{

}
```



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

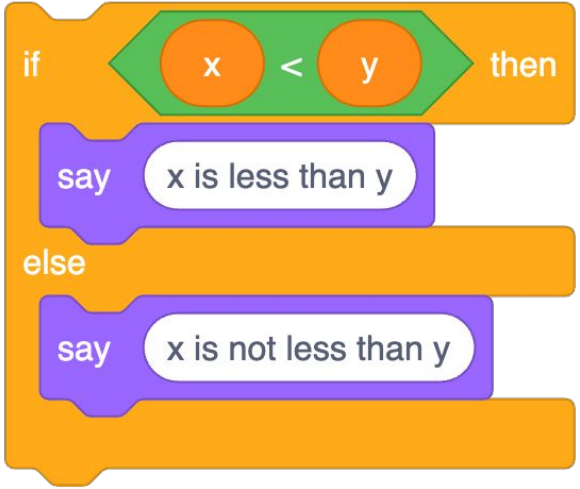


```
if x < y then
  say x is less than y
else
  say x is not less than y
```

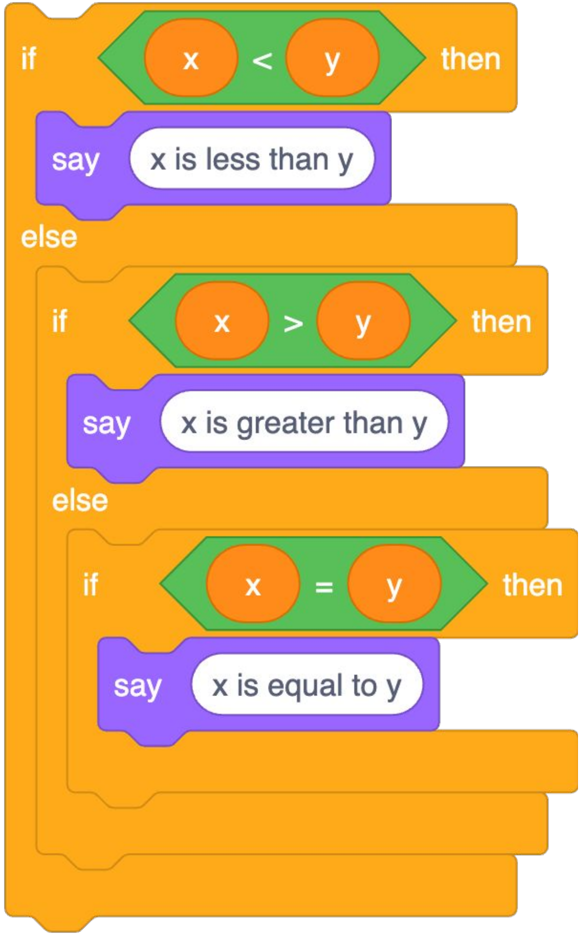


```
if x < y then
    say x is less than y
else
    say x is not less than y
```

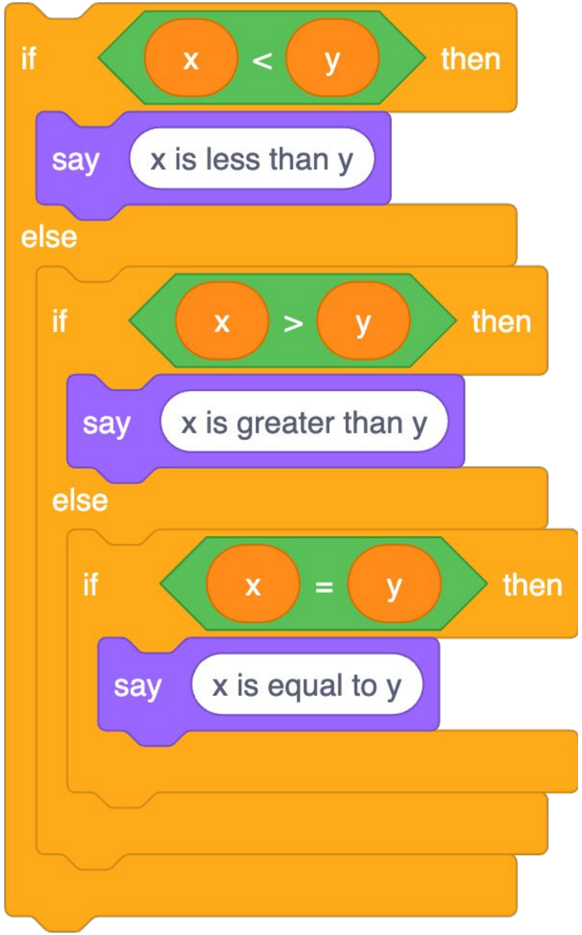
```
if (x < y)
{

}
else
{

}
```



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

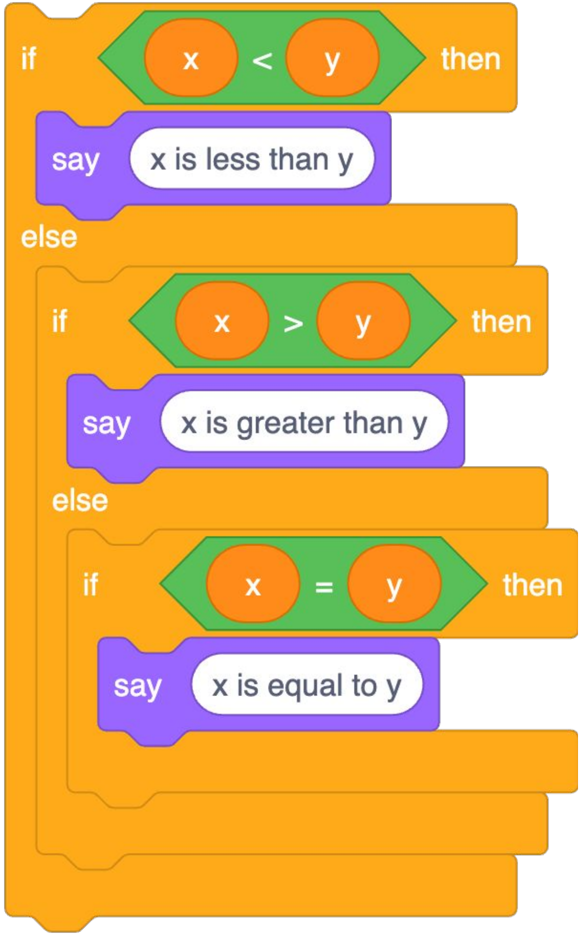


```
if <(x) < (y)> then
    say [x is less than y]
else
    if <(x) > (y)> then
        say [x is greater than y]
    else
        if <(x) = (y)> then
            say [x is equal to y]
        end
    end
end
```



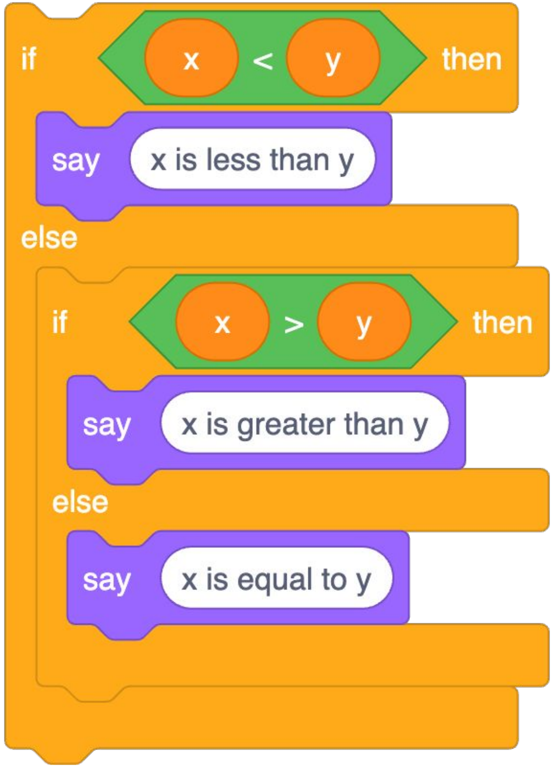
```
if x < y then
    say x is less than y
else
    if x > y then
        say x is greater than y
    else
        if x = y then
            say x is equal to y
```

```
if (x < y)
{

}
else if (x > y)
{

}
else if (x == y)
{

}
```



```
if x < y 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
```

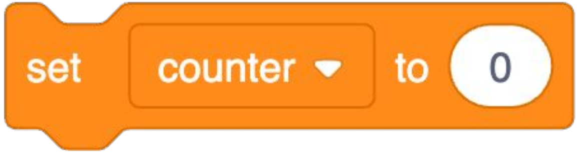
```
if (x < y)
{
    printf("x is less than y\n");
}
else if (x > y)
{
    printf("x is greater than y\n");
}
else if (x == y)
{
    printf("x is equal to y\n");
}
```



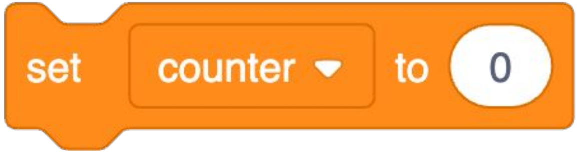
```
if x < y then
  say x is less than y
else
  if x > y then
    say x is greater than y
  else
    say x is equal to y
```

```
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");
}
```

variables

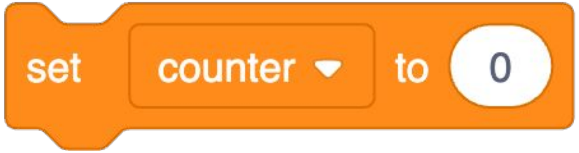


```
set counter ▾ to 0
```



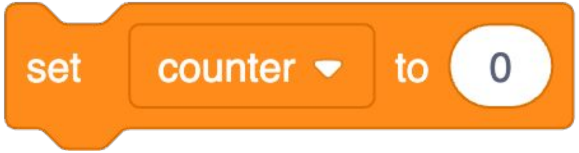
set counter ▾ to 0

```
counter = 0
```

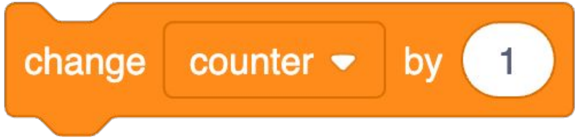


set counter ▾ to 0

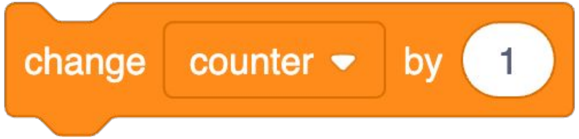
```
int counter = 0
```



```
int counter = 0;
```

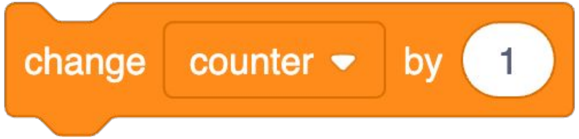


```
change counter ▾ by 1
```



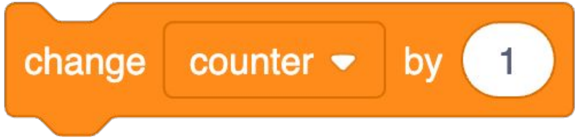
```
change counter by 1
```

```
counter = counter + 1
```

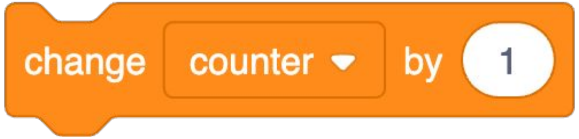


change counter ▾ by 1

```
counter += 1;
```

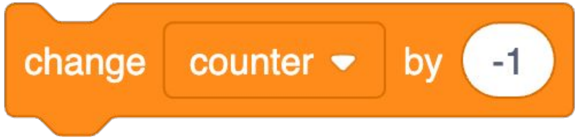


```
counter++;
```



change counter ▾ by 1

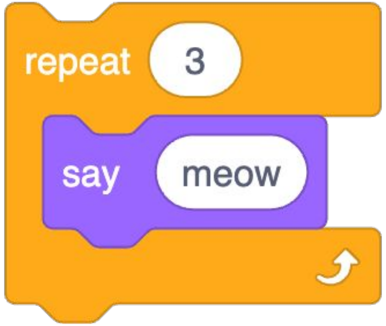
```
counter = counter + 1;
```



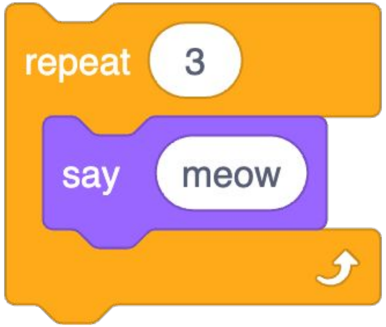
change counter ▾ by -1

```
counter--;
```

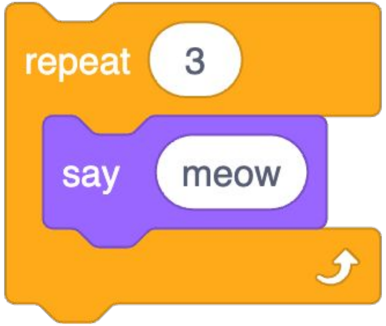
loops



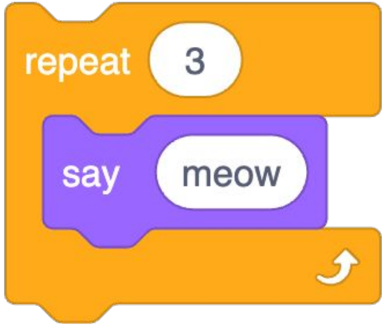
```
repeat 3
    say meow
```



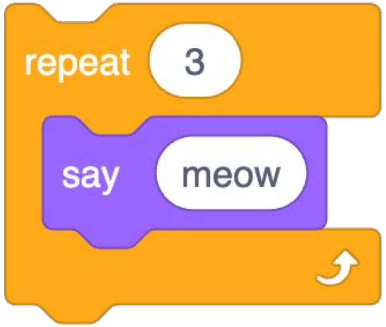
```
int counter = 3;
while (counter > 0)
{
    printf("meow\n");
    counter = counter - 1;
}
```



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



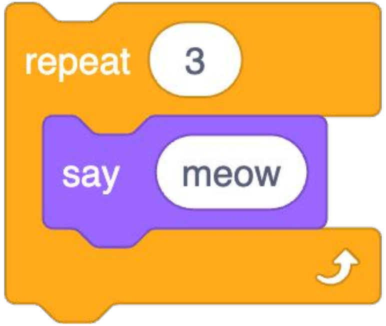
```
int i = 3;
while (i > 0)
{
    printf("meow\n");
    i -= 1;
}
```



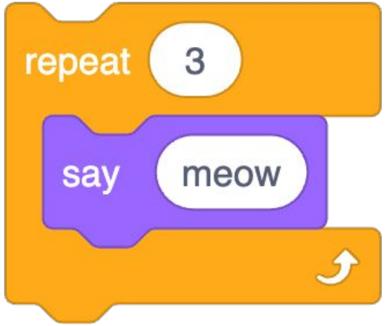
repeat 3

say meow

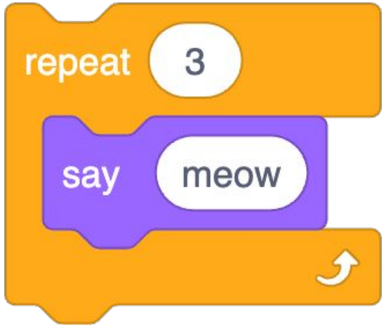
```
int i = 3;
while (i > 0)
{
    printf("meow\n");
    i--;
}
```



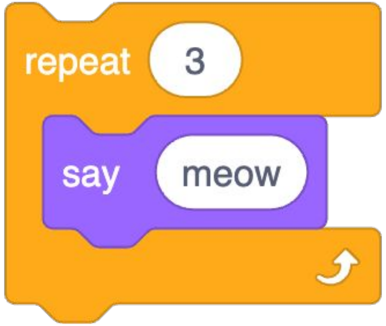
```
int i = 3;
while (i > 0)
{
    printf("meow\n");
    i--;
}
```



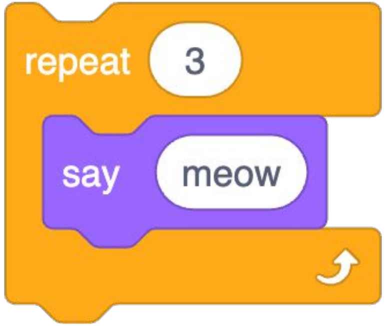
```
int i = 3;
while (i > 0)
{
    printf("meow\n");
    i--;
}
```



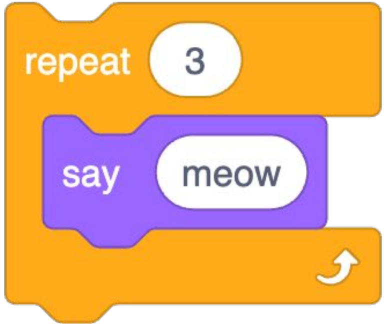
```
int i = 3;
while (i > 0)
{
    printf("meow\n");
    i--;
}
```



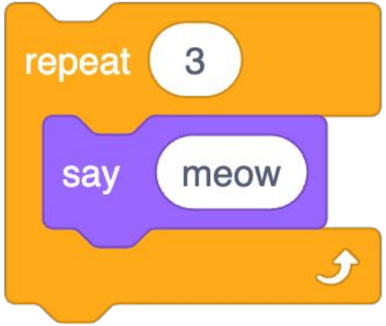
```
int i = 3;
while (i > 0)
{
    printf("meow\n");
    i--;
}
```



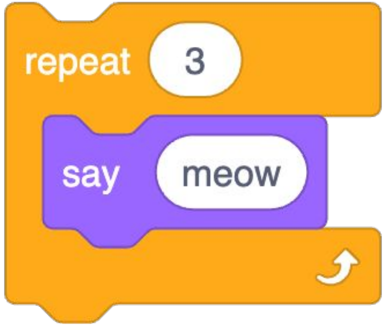
```
int i = 3;
while (i > 0)
{
    printf("meow\n");
    i--;
}
```



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

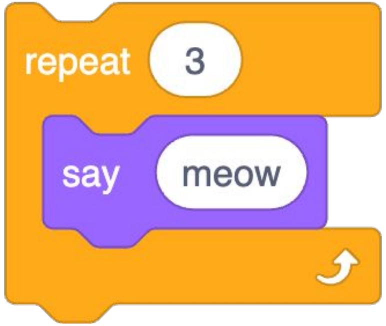


```
int i = 1;
while (i <= 3)
{
    printf("meow\n");
    i++;
}
```

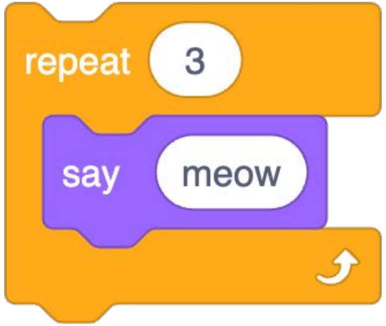


```
repeat 3
    say meow
```

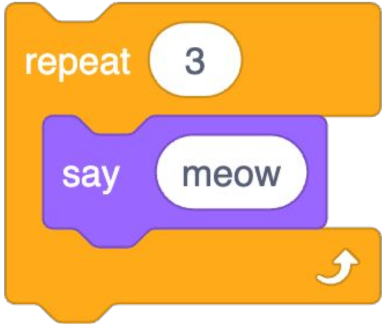
```
int i = 0;
while (i < 3)
{
    printf("meow\n");
    i++;
}
```



```
repeat 3
    say meow
```

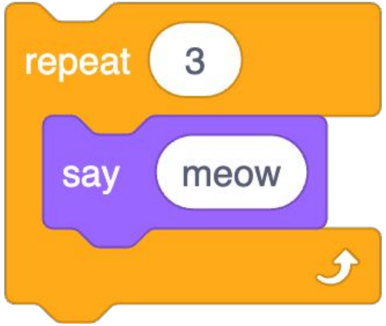


```
for (int i = 0; i < 3; i++)
{

}
```

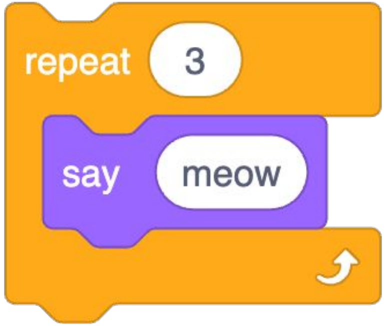


```
repeat 3
    say meow
```

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

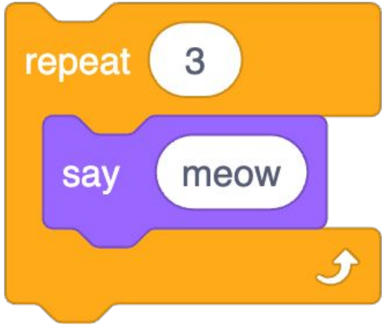


```
for (int i = 0; i < 3; i++)
{
    printf("meow\n");
}
```

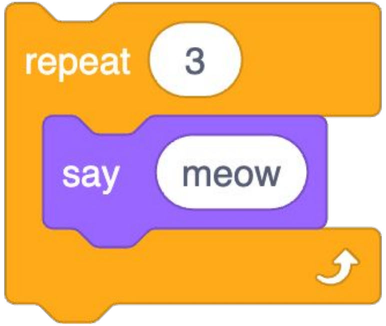


```
repeat 3
    say meow
```

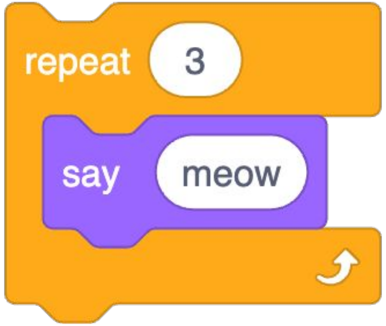
```
for (int i = 0; i < 3; i++)
{
    printf("meow\n");
}
```



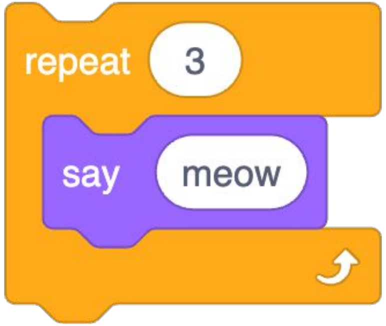
```
for (int i = 0; i < 3; i++)
{
    printf("meow\n");
}
```



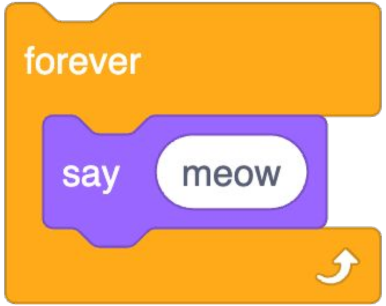
```
for (int i = 0; i < 3; i++)
{
    printf("meow\n");
}
```



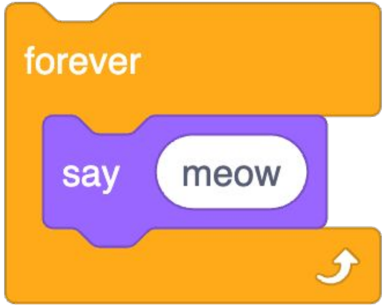
```
for (int i = 0; i < 3; i++)
{
    printf("meow\n");
}
```



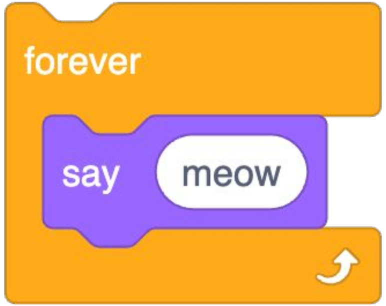
```
for (int i = 0; i < 3; i++)
{
    printf("meow\n");
}
```



```
forever
    say meow
```



```
while (true)
{

}
```



```
while (1)
{
    printf("meow\n");
}
```

Linux

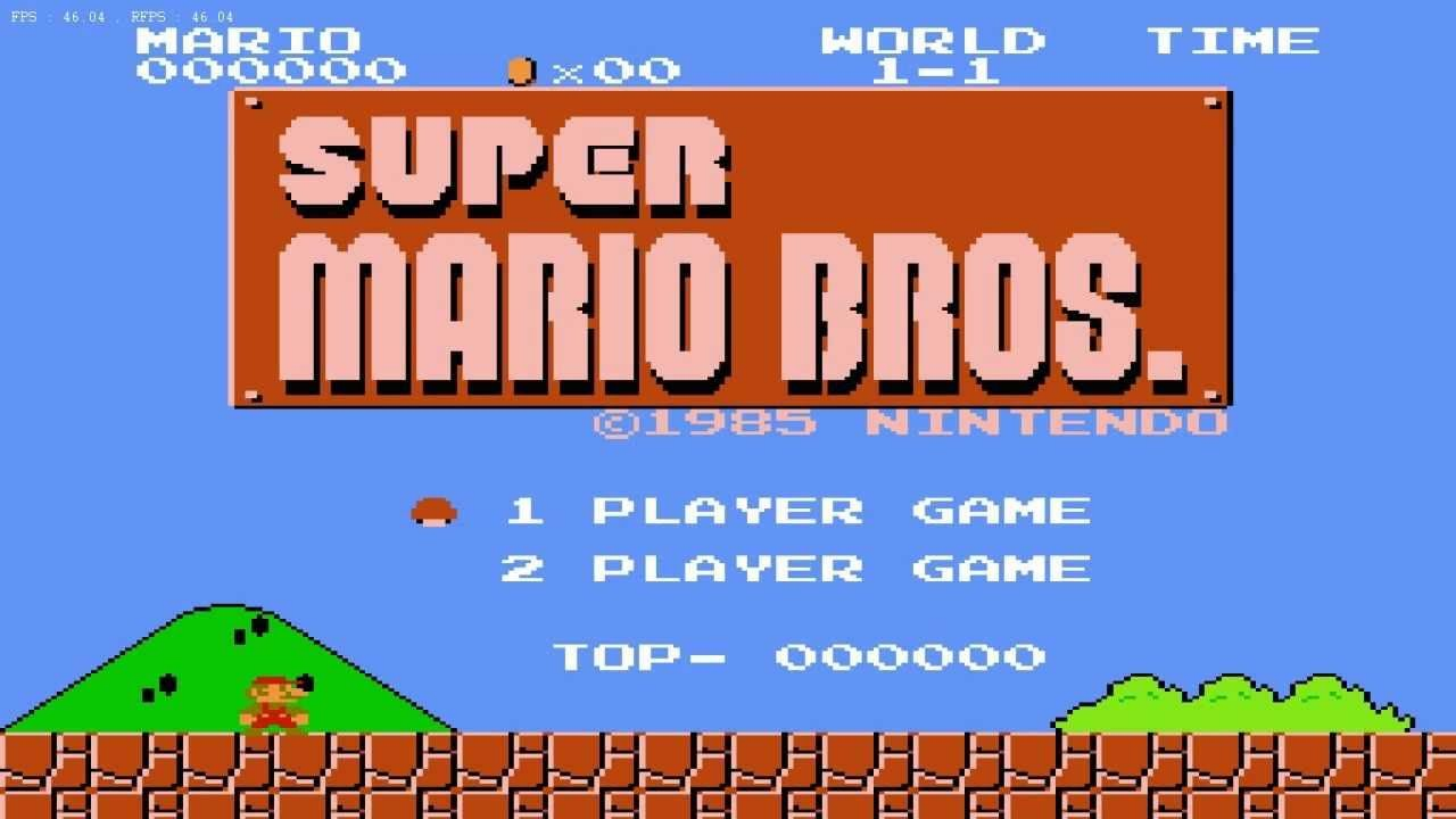
graphical user interface

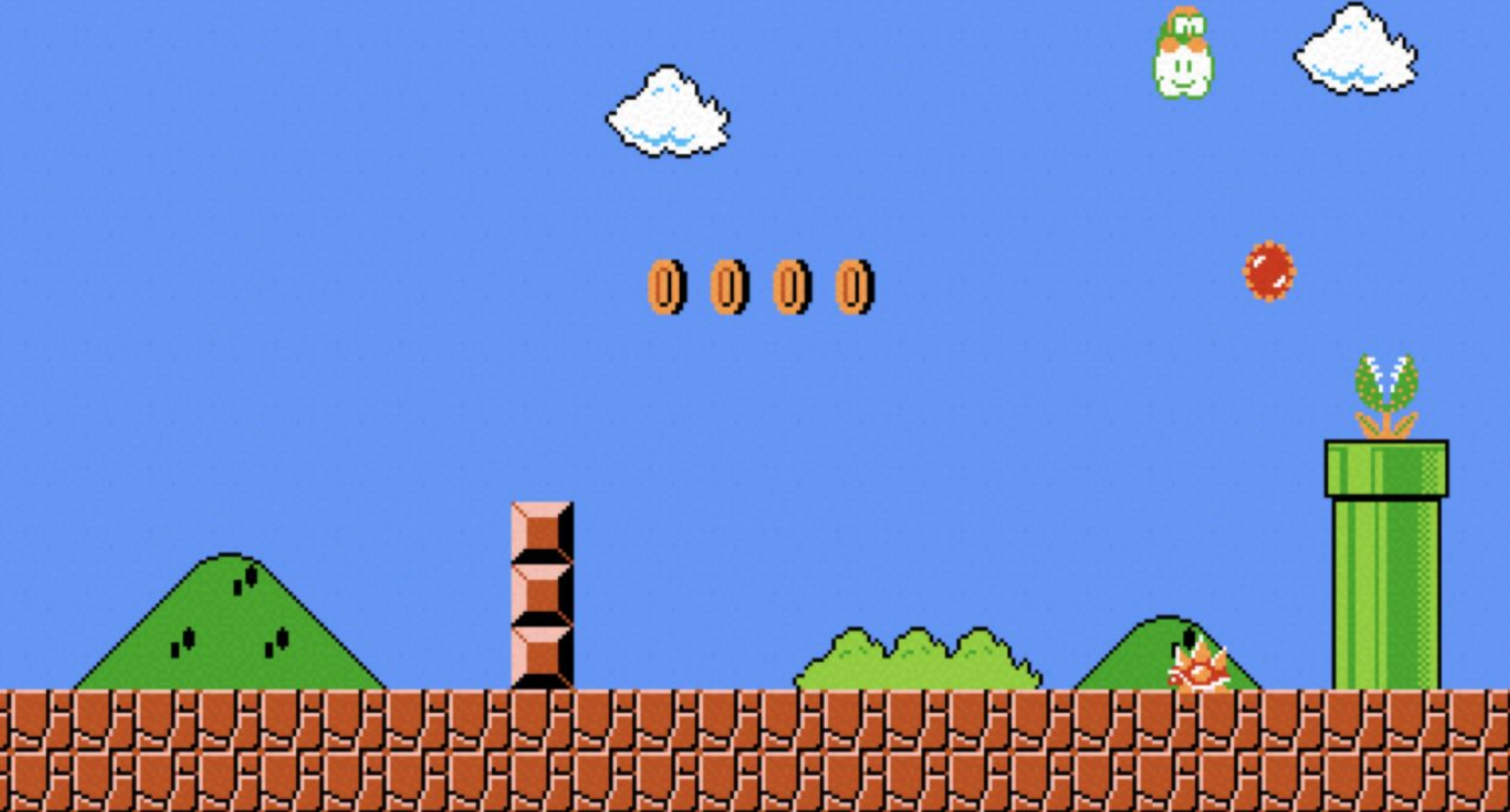
GUI

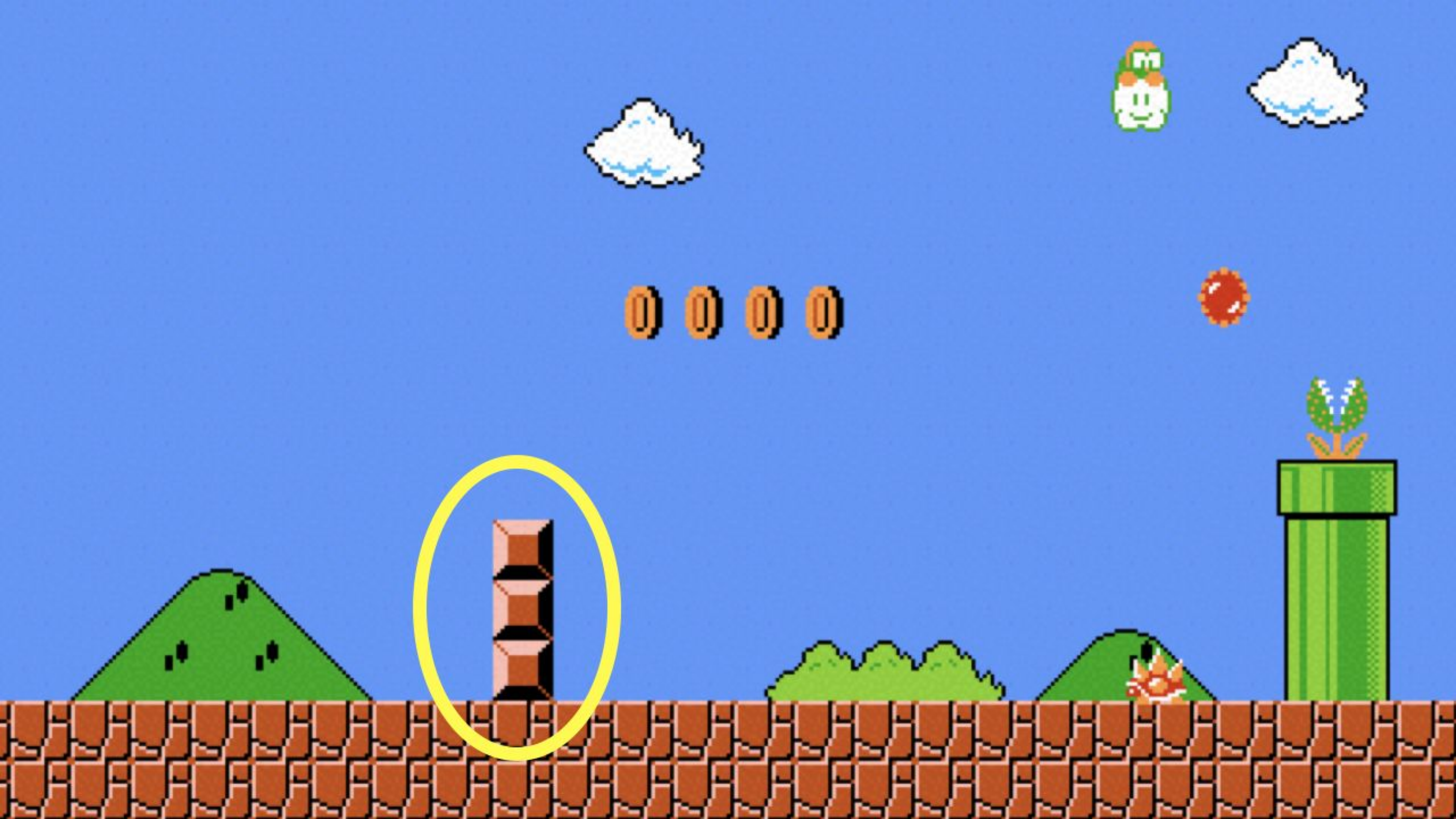
command-line interface

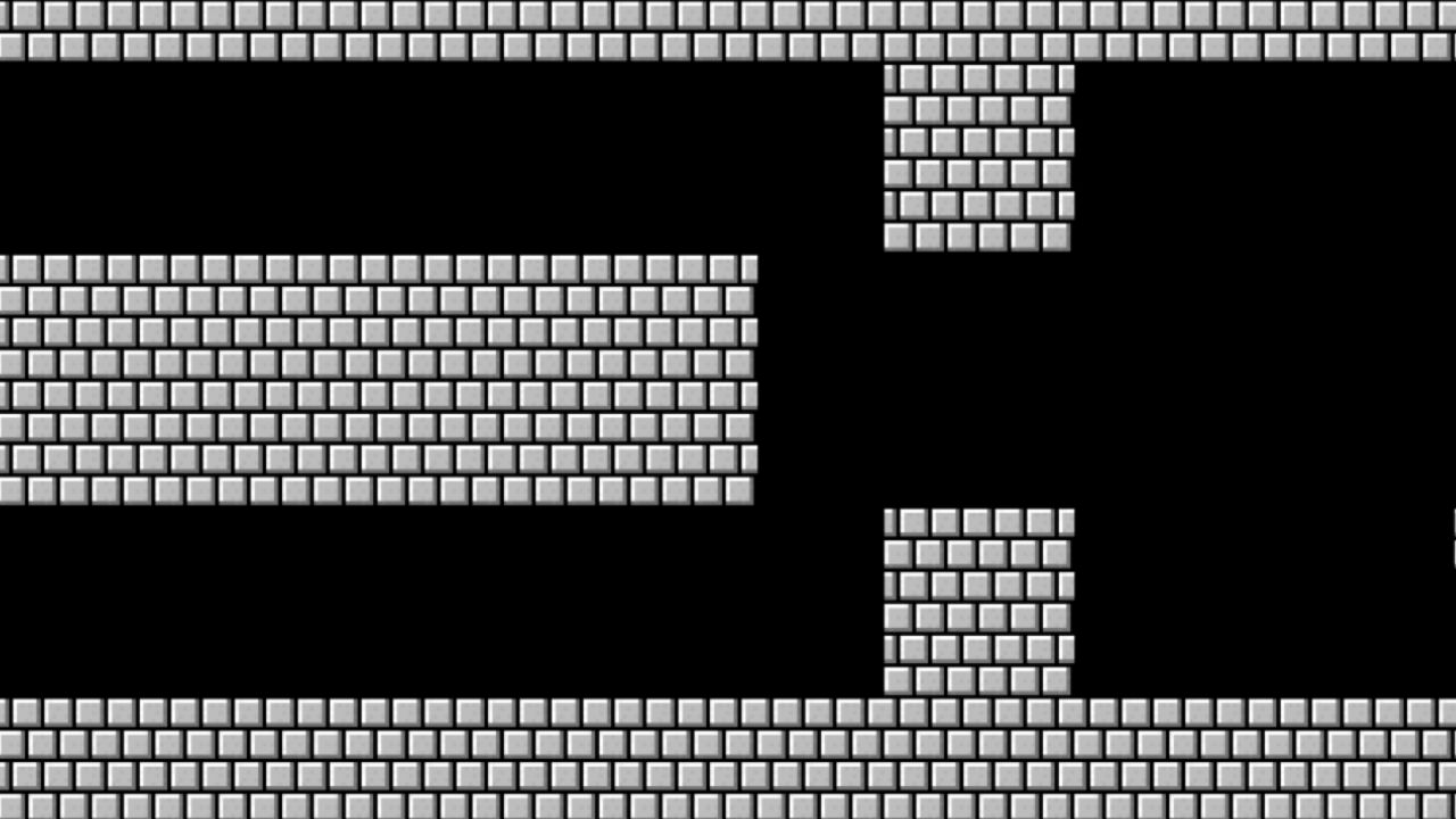
CLI

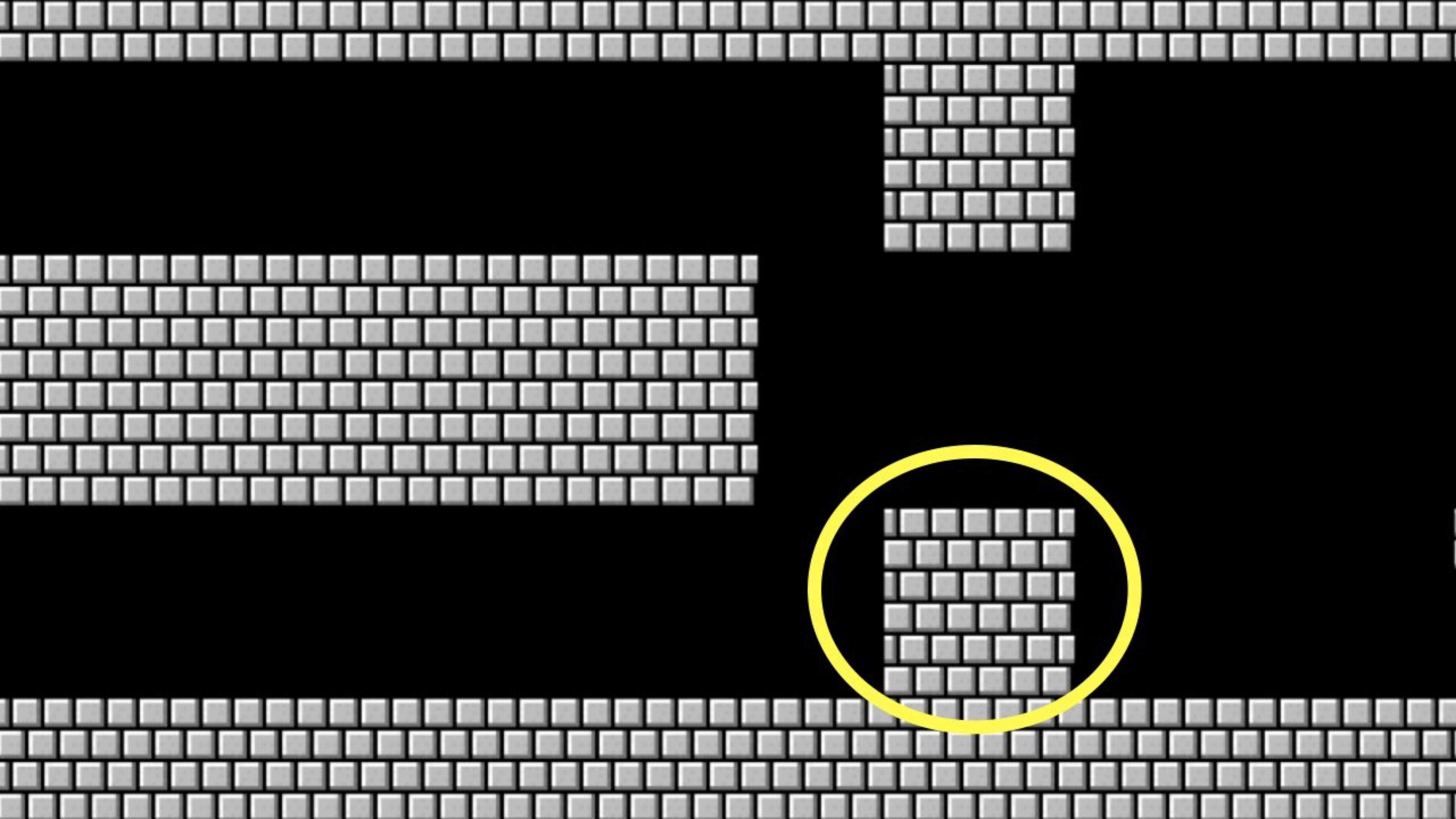
```
cd

cp

ls

mkdir

mv

rm

rmdir

...
```


FPS : 46.04 . RFPS : 46.04
MARIO
000000
×00
WORLD
1-1
TIME
SUPER
MARIO BROS.
©1985 NINTENDO
1 PLAYER GAME
2 PLAYER GAME
TOP- 000000









constants

comments

pseudocode

operators

+

-

*

/

%

crucial™
by Micron
KN-13
94V-0
47J6
4G85
4G85
4G85
8BBI2
D9HXT
8BBI2
D9HXT

00000000000000000000000000000000

11111111111111111111111111111111

4294967295

2147483647

-2147483648

integer overflow

0000

0001

0010

0011

0100

0101

0110

0111

1000

format codes

```
%c

%f

%i

%li

%s
```

9223372036854775807

truncation

type casting

floating-point imprecision

1999

1999

1900

19 January 2038

13 December 1901

This is CS50