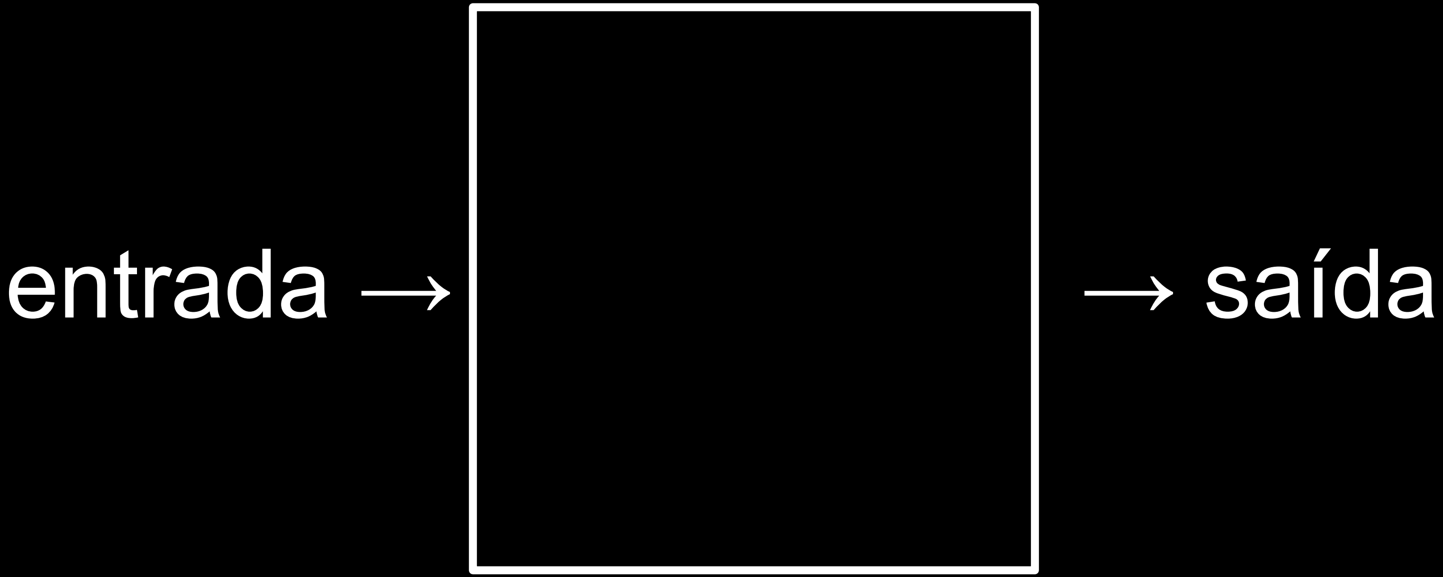
Isso (não) é CS50



entrada →

→ saída

000

001

010

011

100

101

110

111

123

1

123

$$1\overset{10}{2}\overset{1}{3}$$

100 10 1

123

100 10 1

123

100 × 1

100 10 1

123

100 × 1 + 10 × 2

100 10 1

123

100 × 1 + 10 × 2 + 1 × 3

100 10 1

123

100 + 20 + 3

123

100	10	1
#	#	#

10^2 10^1 10^0

###

2^2	2^1	2^0
#	#	#

4 2 1

###

4	2	1
0	0	0

4	2	1
0	0	1

4	2	1
0	1	0

4 2 1

011

4 2 1

100

4	2	1
1	0	1

4	2	1
1	1	0

4 2 1

111

A

65

01000001

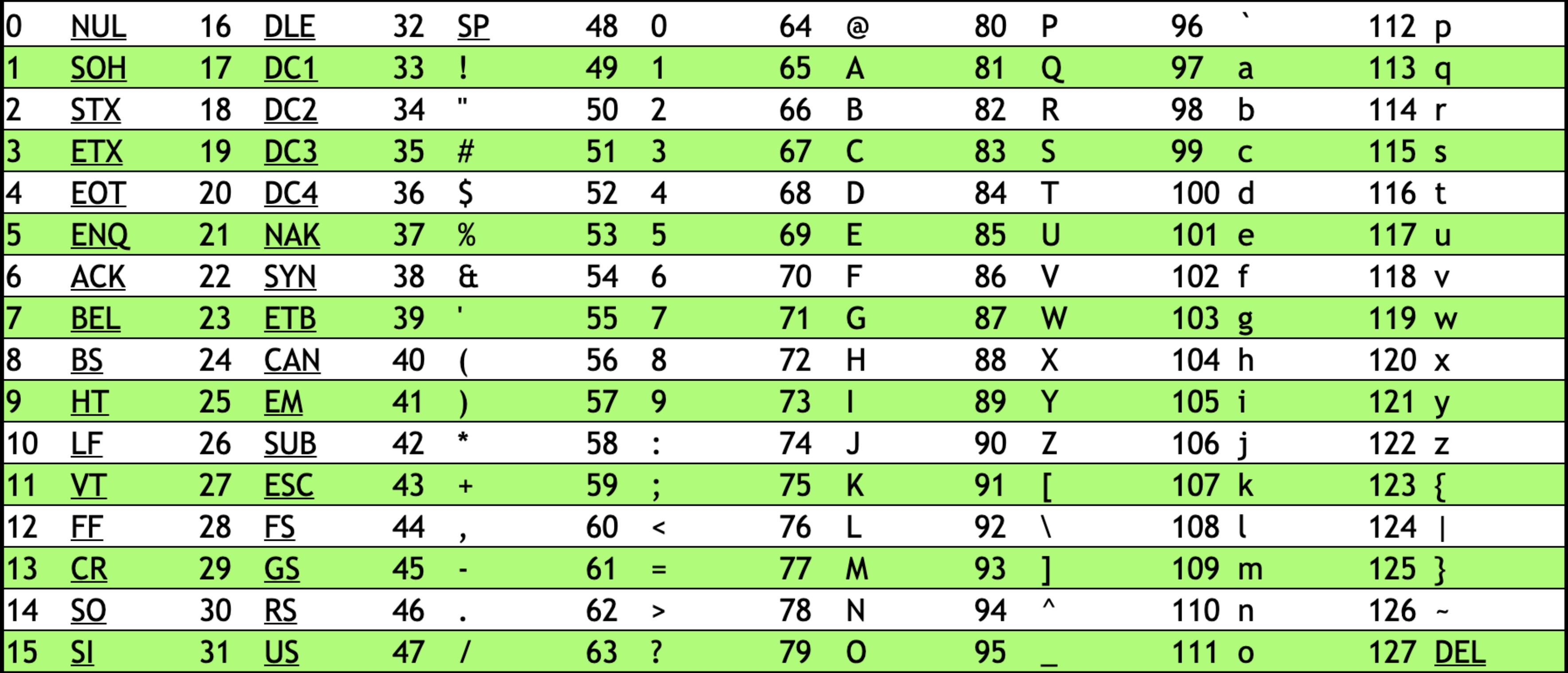
ASCII

...	A	B	C	D	E	F	G	H	I	...
...	65	66	67	68	69	70	71	72	73	...

72 73 33

H	I	
72	73	33

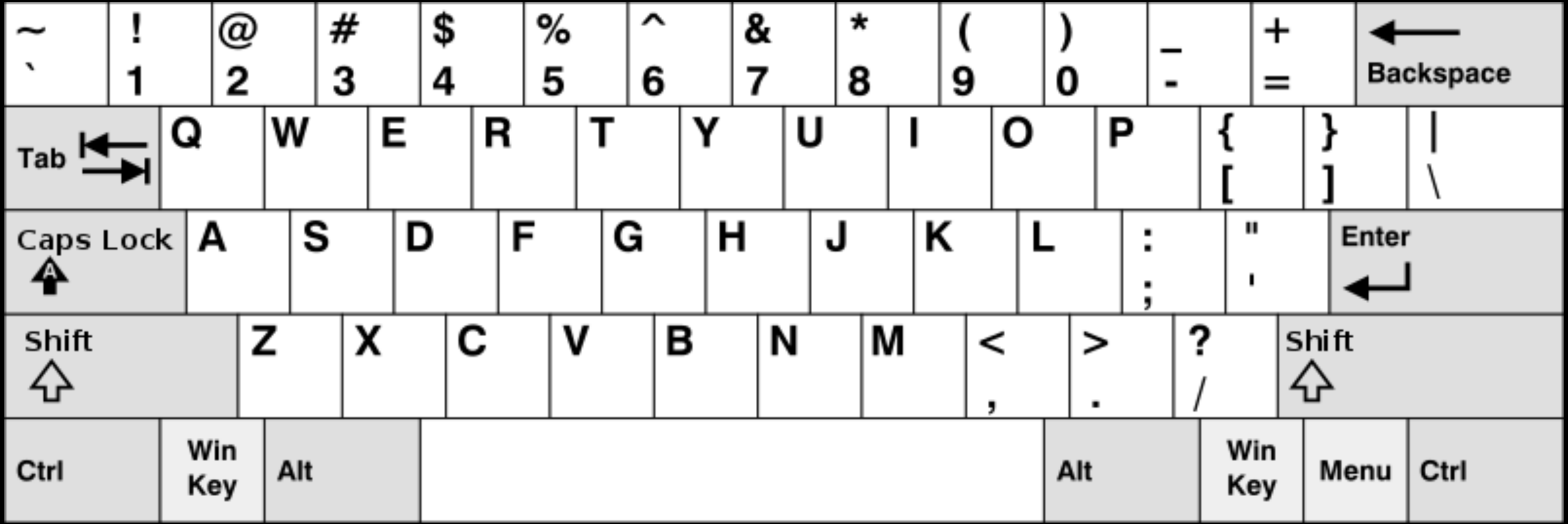
H	I	!
72	73	33

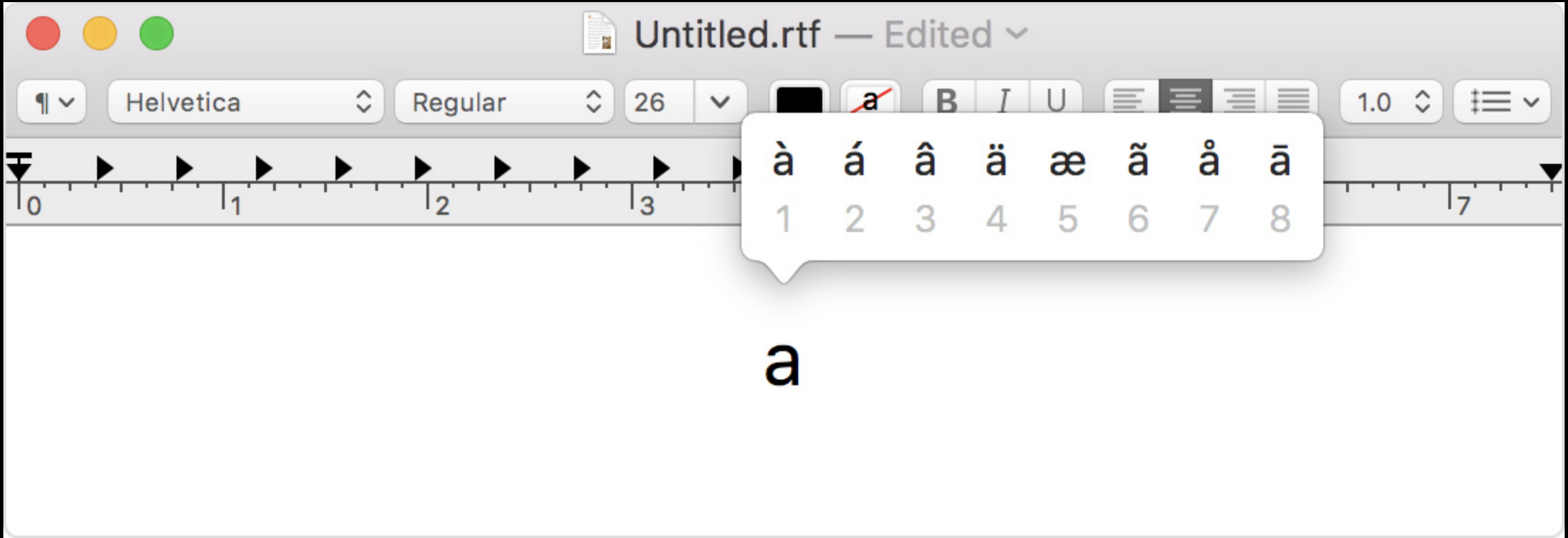


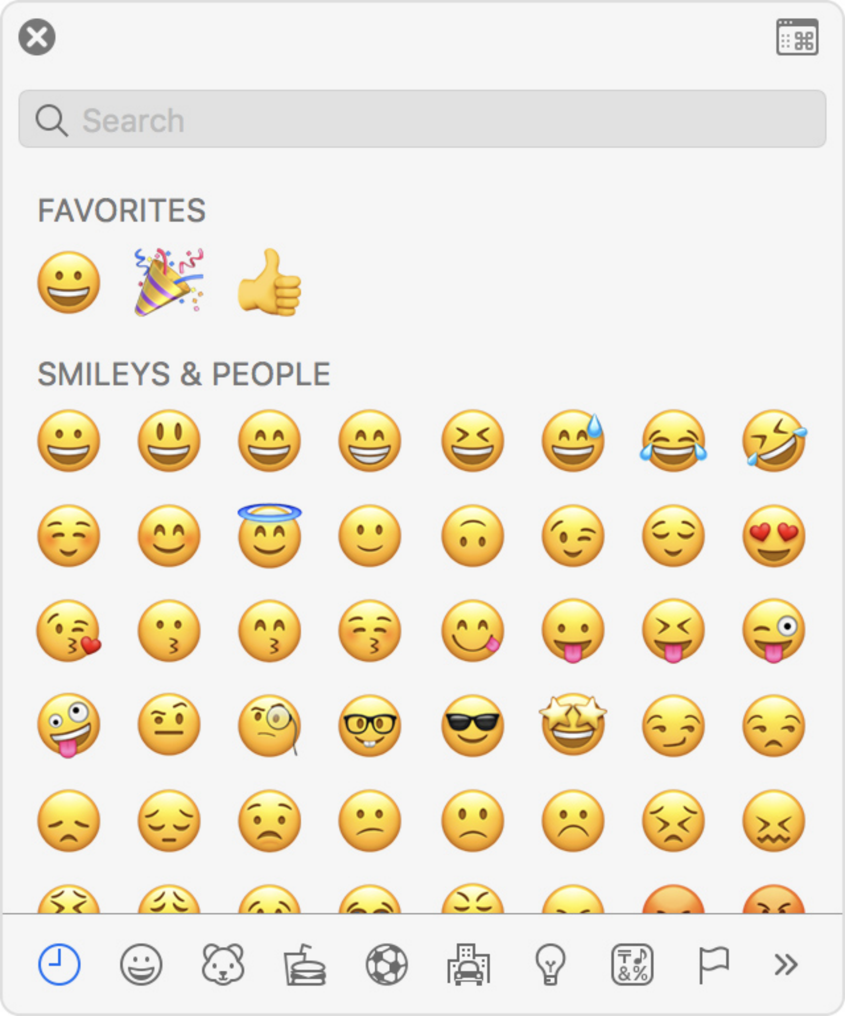
0	NUL	16	DLE	32	SP	48	0	64	@	80	P	96	`	112	p
1	SOH	17	DC1	33	!	49	1	65	A	81	Q	97	a	113	q
2	STX	18	DC2	34	"	50	2	66	B	82	R	98	b	114	r
3	ETX	19	DC3	35	#	51	3	67	C	83	S	99	c	115	s
4	EOT	20	DC4	36	$	52	4	68	D	84	T	100	d	116	t
5	ENQ	21	NAK	37	%	53	5	69	E	85	U	101	e	117	u
6	ACK	22	SYN	38	&	54	6	70	F	86	V	102	f	118	v
7	BEL	23	ETB	39	'	55	7	71	G	87	W	103	g	119	w
8	BS	24	CAN	40	(	56	8	72	H	88	X	104	h	120	x
9	HT	25	EM	41	)	57	9	73	I	89	Y	105	i	121	y
10	LF	26	SUB	42	*	58	:	74	J	90	Z	106	j	122	z
11	VT	27	ESC	43	+	59	;	75	K	91	[	107	k	123	{
12	FF	28	FS	44	,	60	<	76	L	92	\	108	l	124	|
13	CR	29	GS	45	-	61	=	77	M	93	]	109	m	125	}
14	SO	30	RS	46	.	62	>	78	N	94	^	110	n	126	~
15	SI	31	US	47	/	63	?	79	O	95	_	111	o	127	DEL

H	I	!
72	73	33

H	I	!
01001000	01001001	00100001


~ `
! 1
@ 2
# 3
$ 4
% 5
^ 6
& 7
* 8
(9
) 0
_ -
+ =
Backspace
Tab
Q
W
E
R
T
Y
U
I
O
P
{ [
}]
| \
Caps Lock
A
S
D
F
G
H
J
K
L
: ;
" '
Enter
Shift
Z
X
C
V
B
N
M
< ,
> .
? /
Shift
Ctrl
Win Key
Alt
Alt
Win Key
Menu
Ctrl


Untitled.rtf — Edited
¶
Helvetica
Regular
26
B
I
U
1.0
à á â ä æ ã å ā
1 2 3 4 5 6 7 8
0
1
2
3
7
a



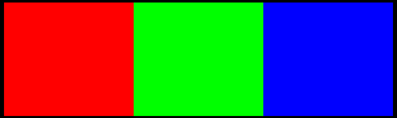
Search
FAVORITES
SMILEYS & PEOPLE

Unicode

4,036,991,159

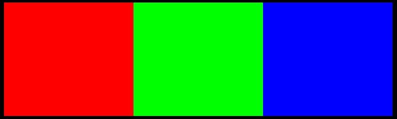
11110000 10011111 10011000 10110111

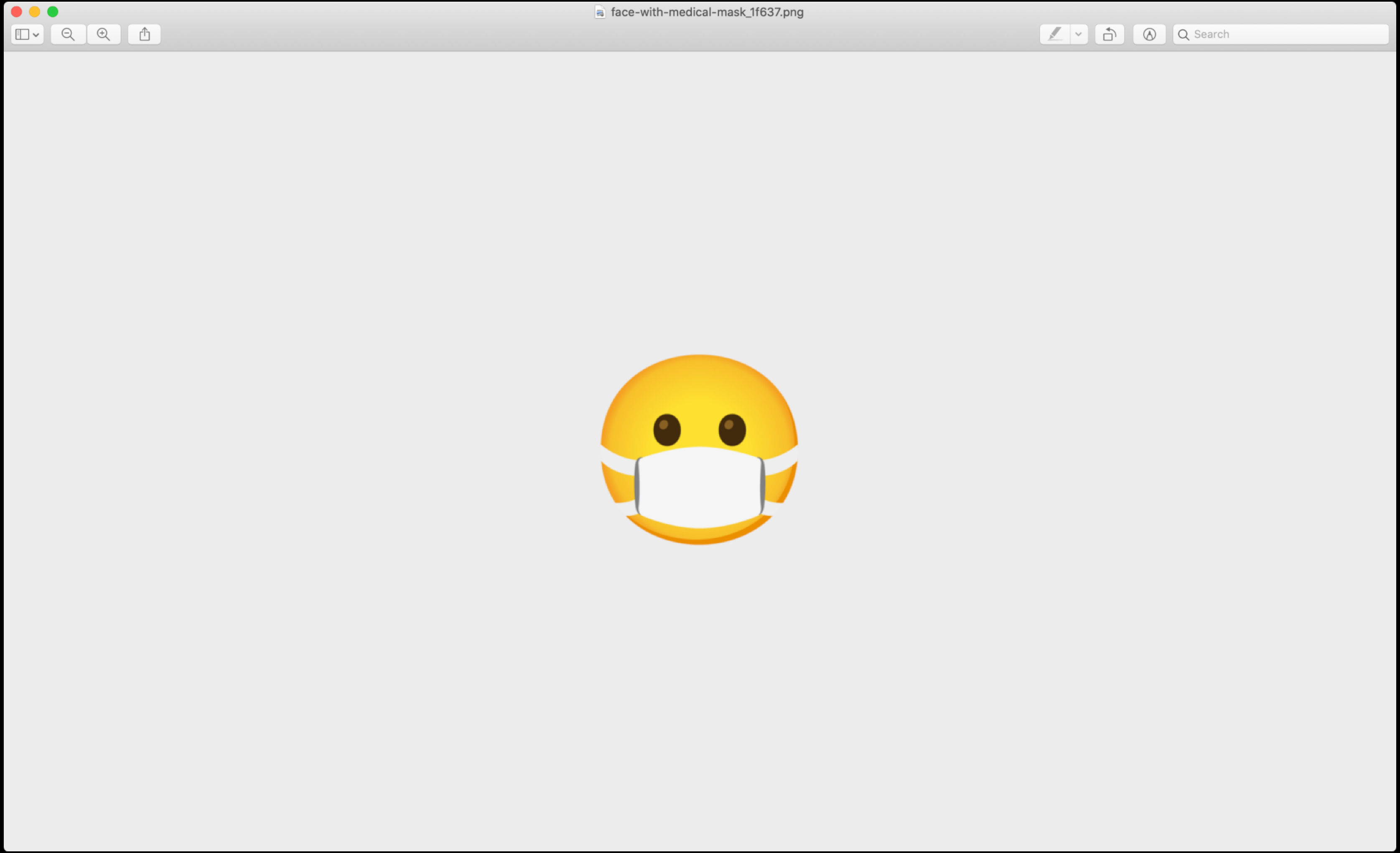
RGB



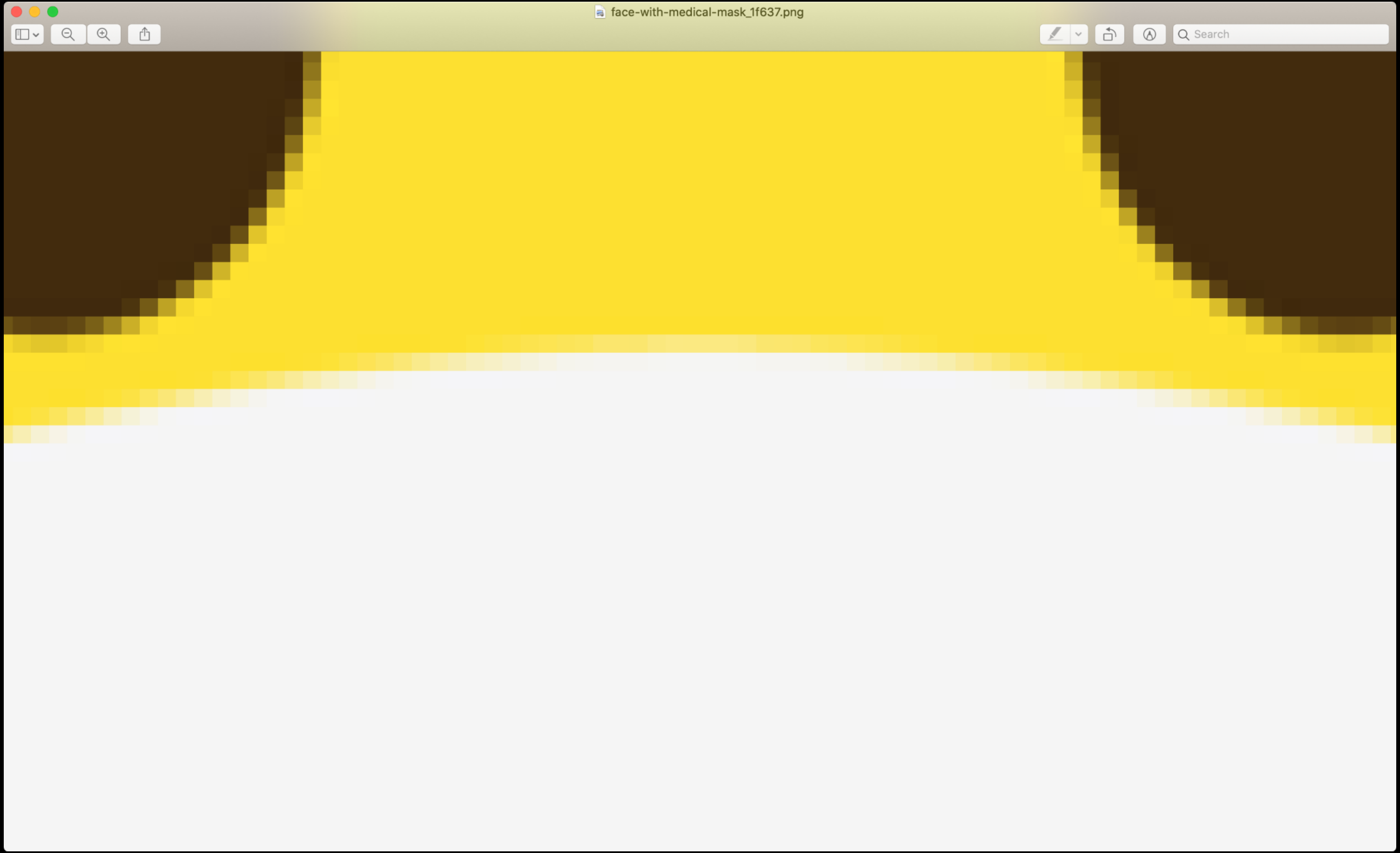
72 73 33

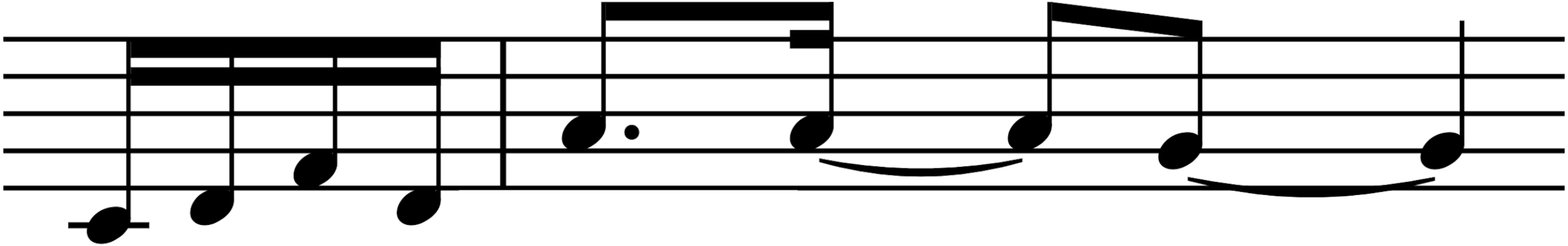
72	73	33



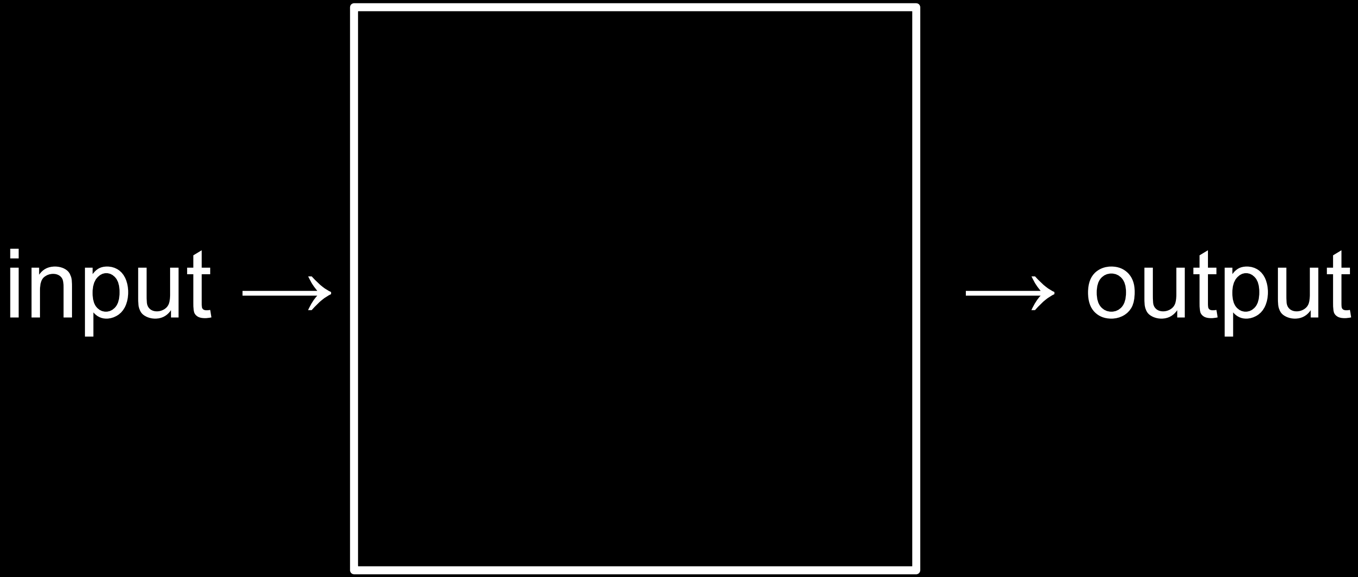

face-with-medical-mask_1f637.png
Search

face-with-medical-mask_1f637.png
Search


face-with-medical-mask_1f637.png
Search



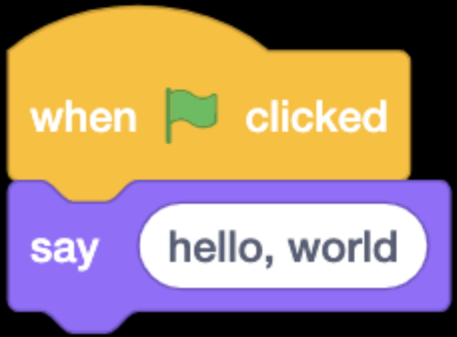
```
144 60 64
128 60 64
144 62 64
128 62 64
144 65 64
128 65 64
144 62 64
128 62 64
144 69 64
128 69 64
144 69 64
128 69 64
144 67 64
128 67 64
```



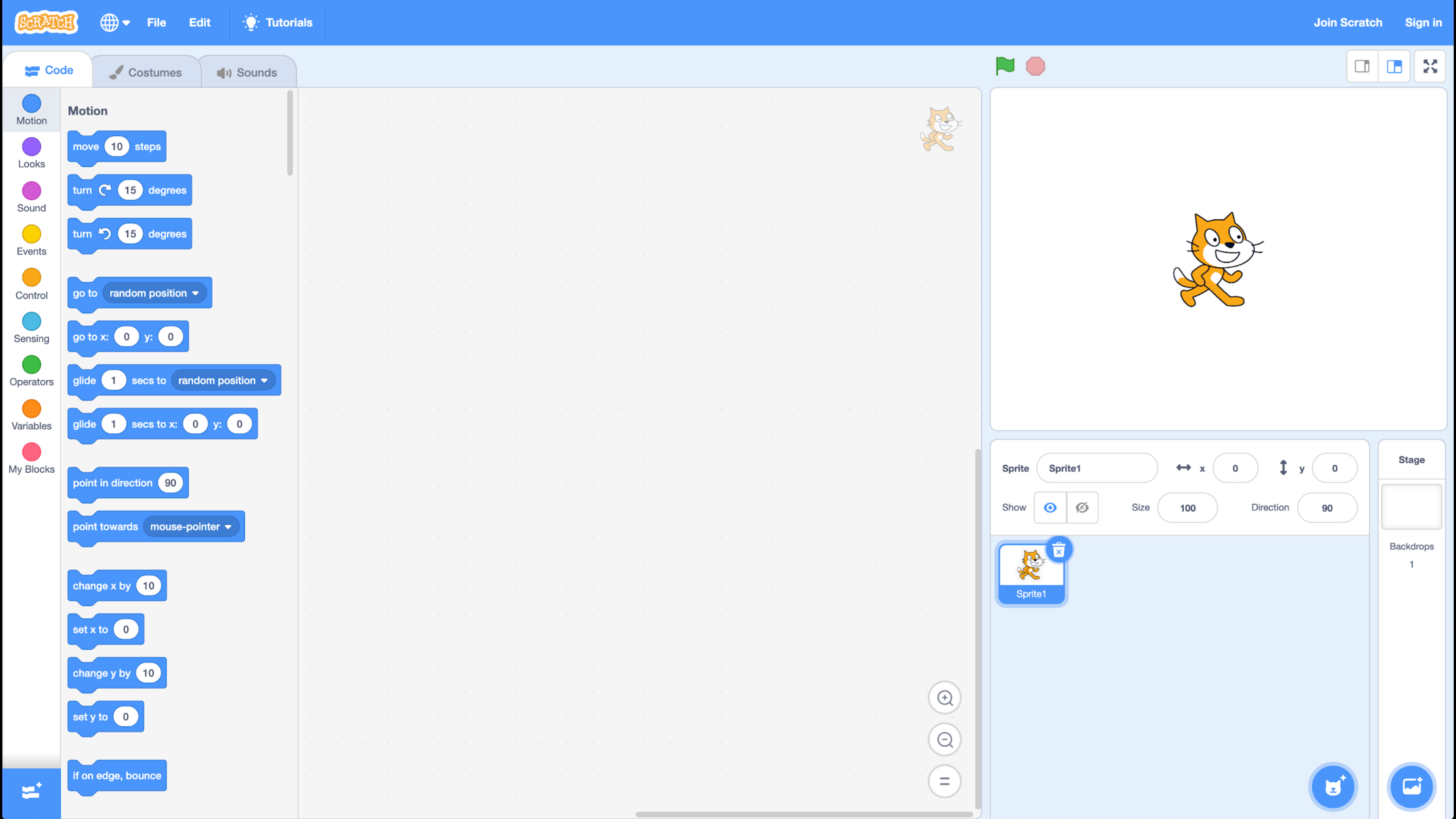
input → → output

```
#include <stdio.h>

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

```
print("hello, world")
```

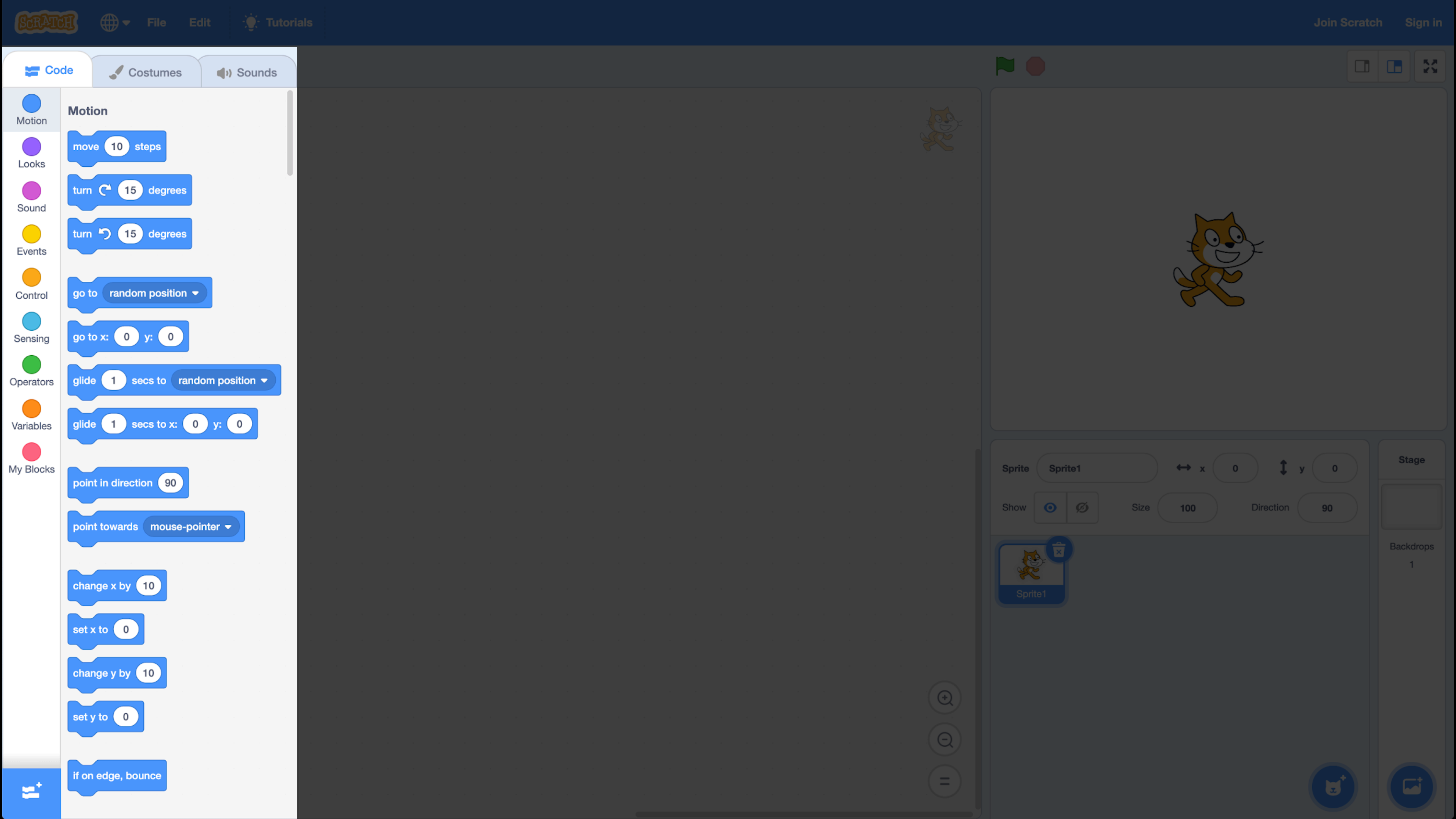


```
when ⚑ clicked
say (hello, world)
```



Sprite Sprite1 ↔ x 0 ↕ y 0

Show Size 100 Direction 90



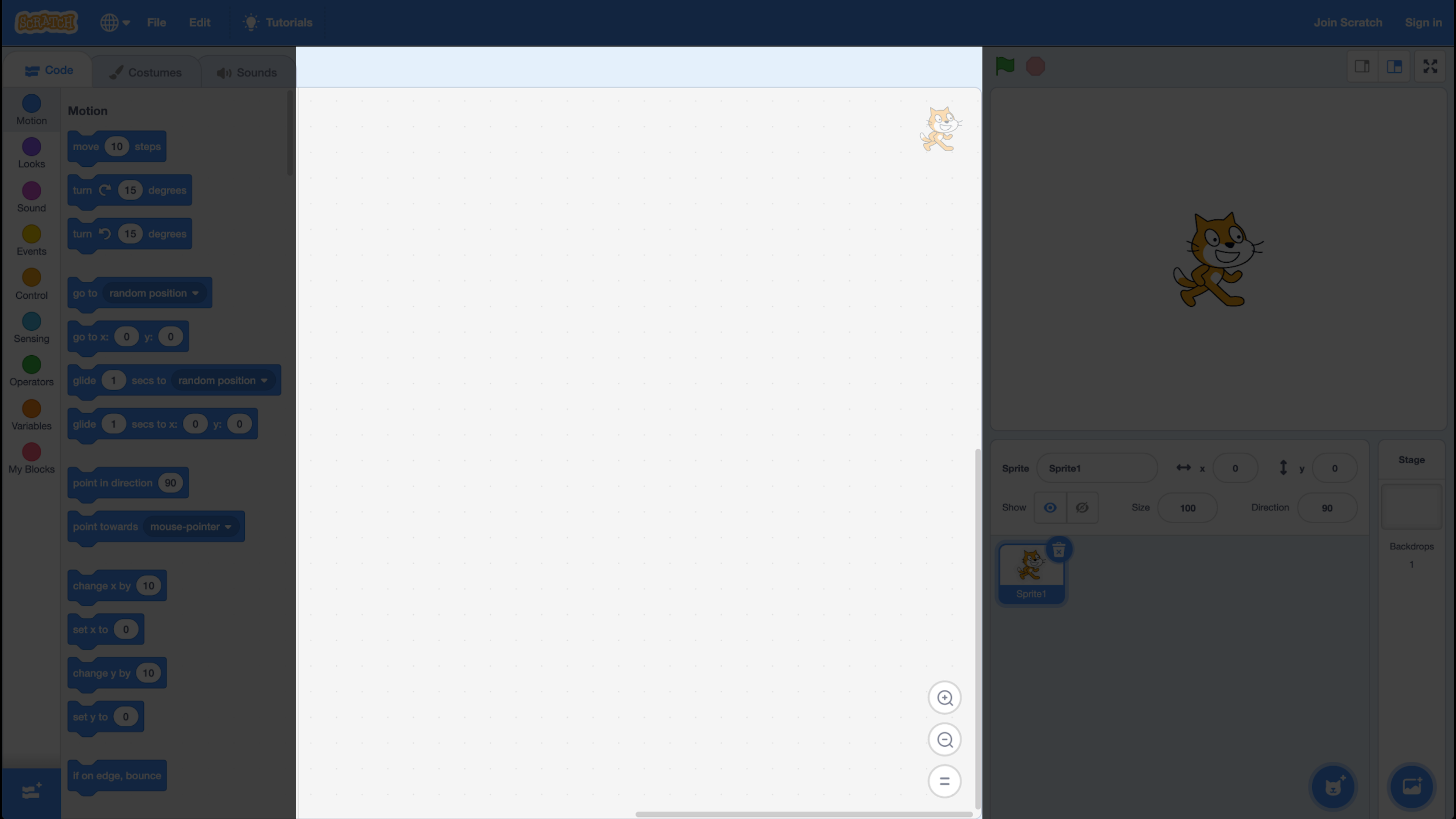
Motion

```
move (10) steps
turn ↻ (15) degrees
turn ↺ (15) degrees
go to (random position ▾)
go to x: (0) y: (0)
glide (1) secs to (random position ▾)
glide (1) secs to x: (0) y: (0)
point in direction (90)
point towards (mouse-pointer ▾)
change x by (10)
set x to (0)
change y by (10)
set y to (0)
if on edge, bounce
```

Sprite: Sprite1 ↔ x: 0 ↕ y: 0

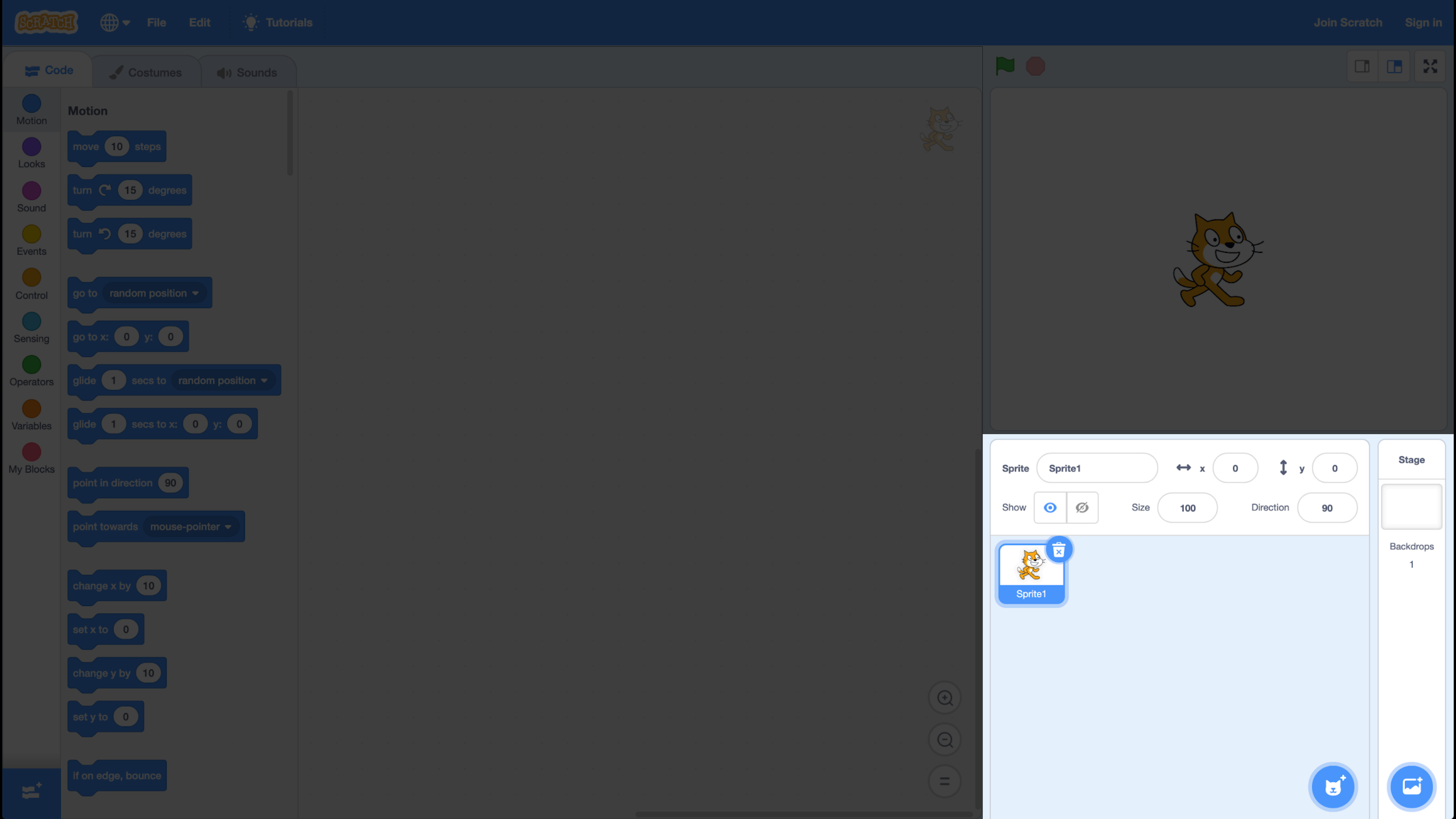
Size: 100 Direction: 90

Sprite1



Sprite Sprite1 ↔ x 0 ↕ y 0

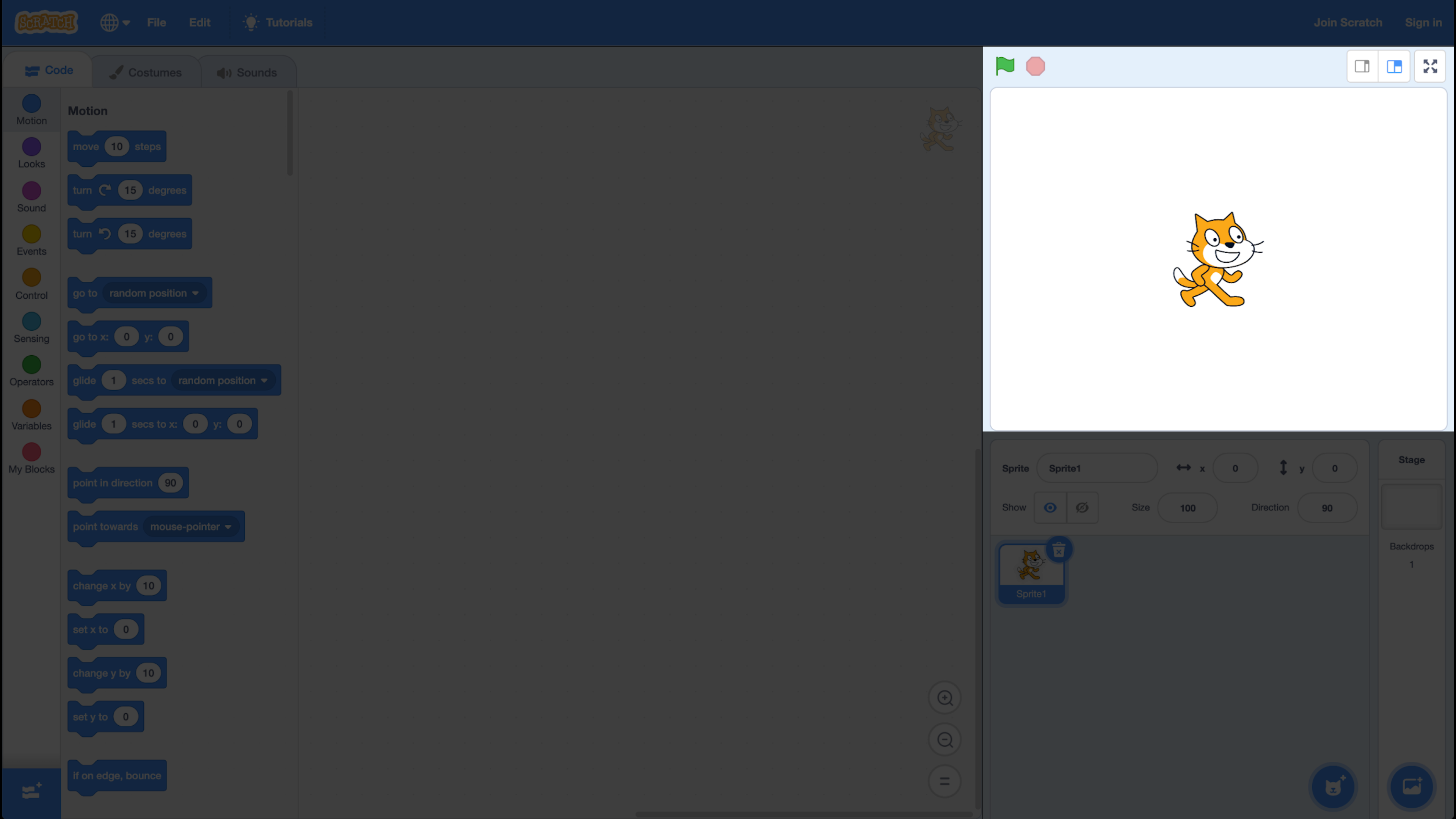
Show Size 100 Direction 90



Sprite Sprite1 x 0 y 0

Show Size 100 Direction 90

Sprite1



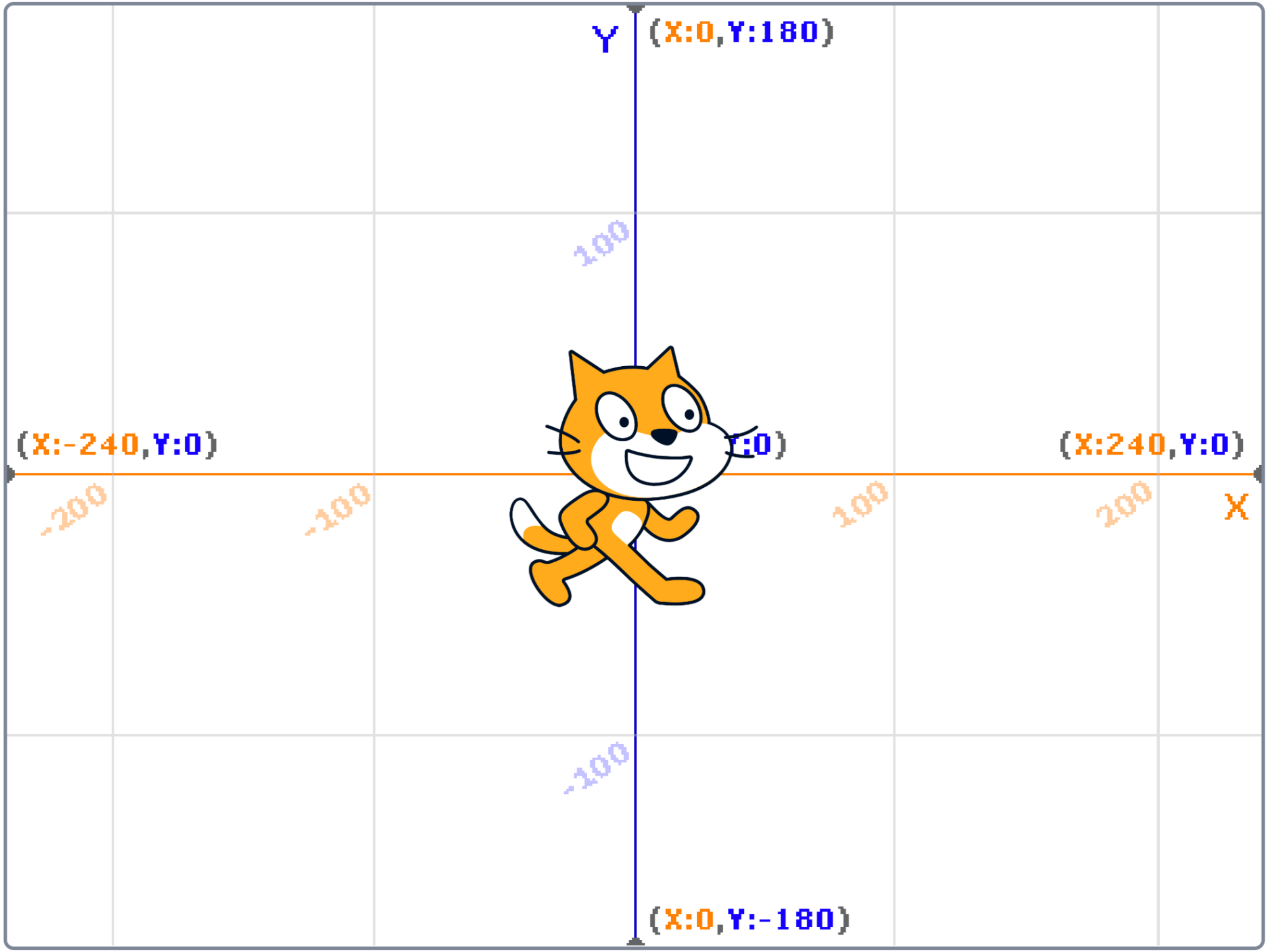
Motion

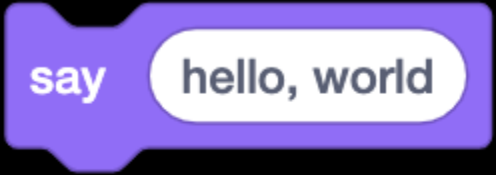
```
move (10) steps
turn ↻ (15) degrees
turn ↺ (15) degrees
go to (random position ▾)
go to x: (0) y: (0)
glide (1) secs to (random position ▾)
glide (1) secs to x: (0) y: (0)
point in direction (90)
point towards (mouse-pointer ▾)
change x by (10)
set x to (0)
change y by (10)
set y to (0)
if on edge, bounce
```

Sprite Sprite1 ↔ x 0 ↕ y 0

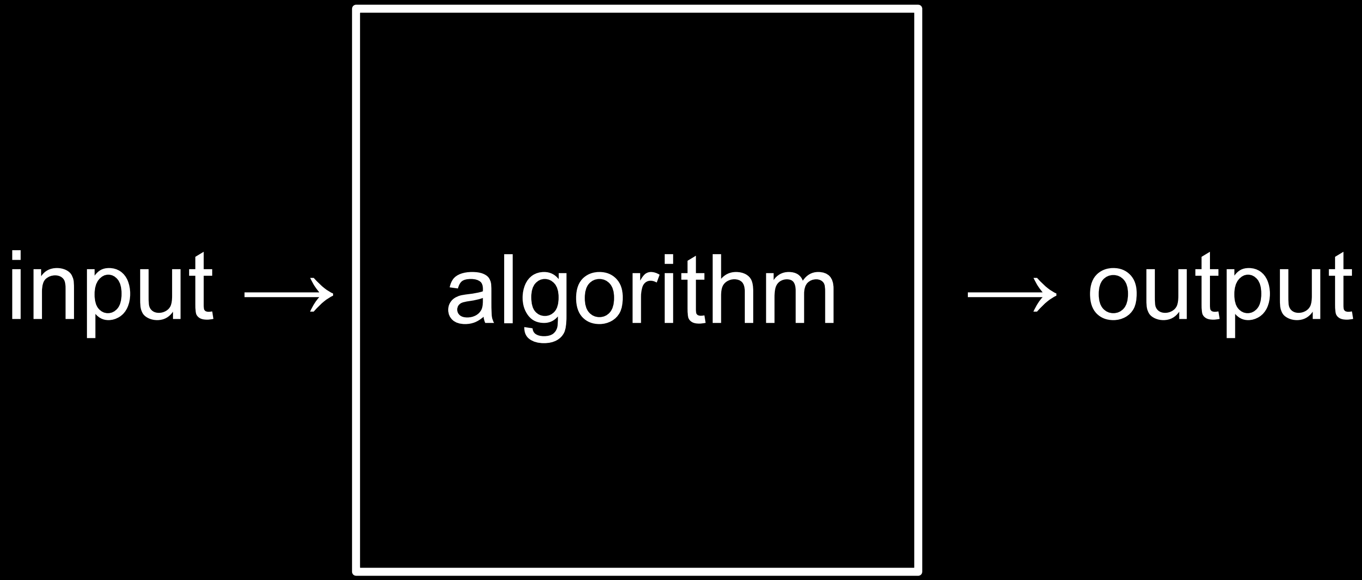
Show Size 100 Direction 90

Sprite1

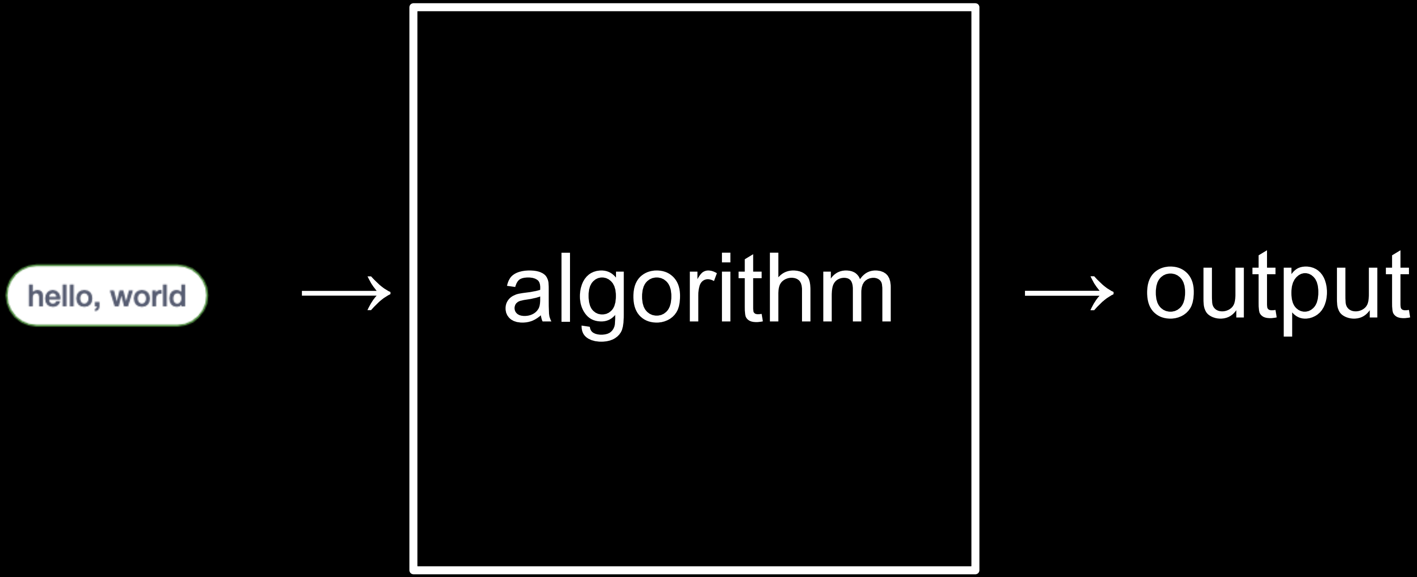

Y
(X:0,Y:180)
100
(X:-240,Y:0)
Y:0)
(X:240,Y:0)
-200
-100
100
200
X
-100
(X:0,Y:-180)



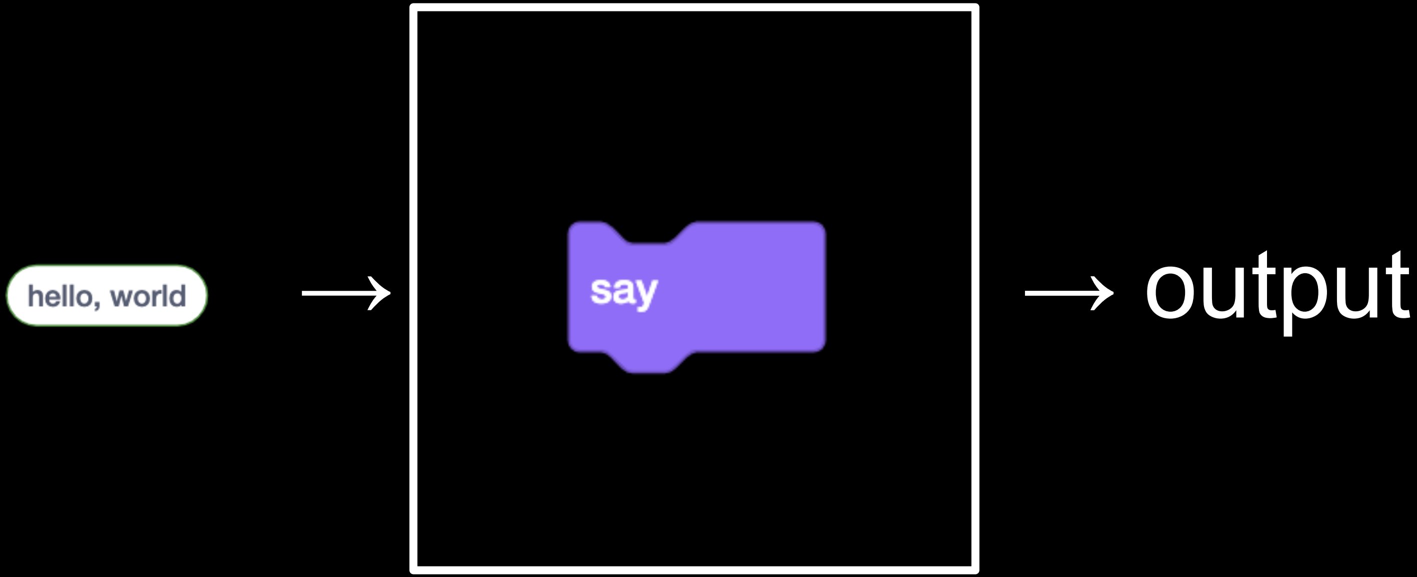
```
say hello, world
```



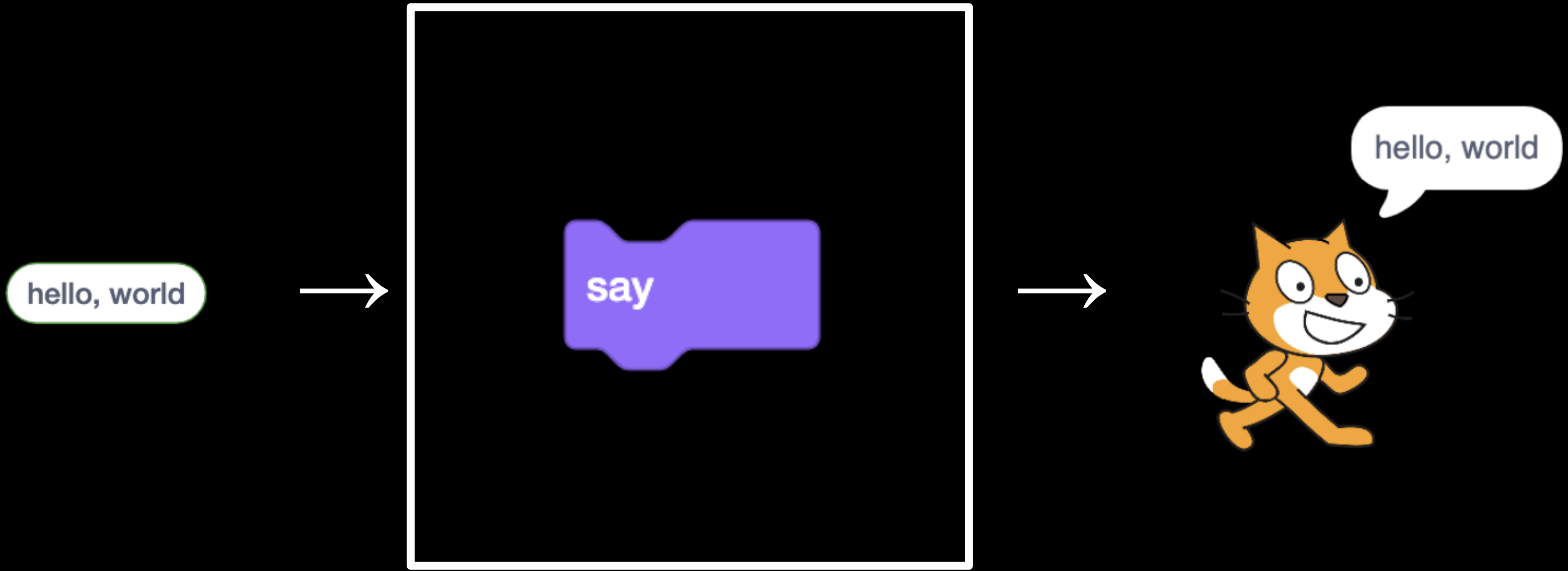
input → algorithm → output

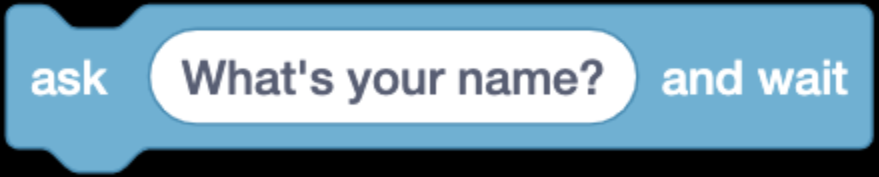


hello, world → algorithm → output

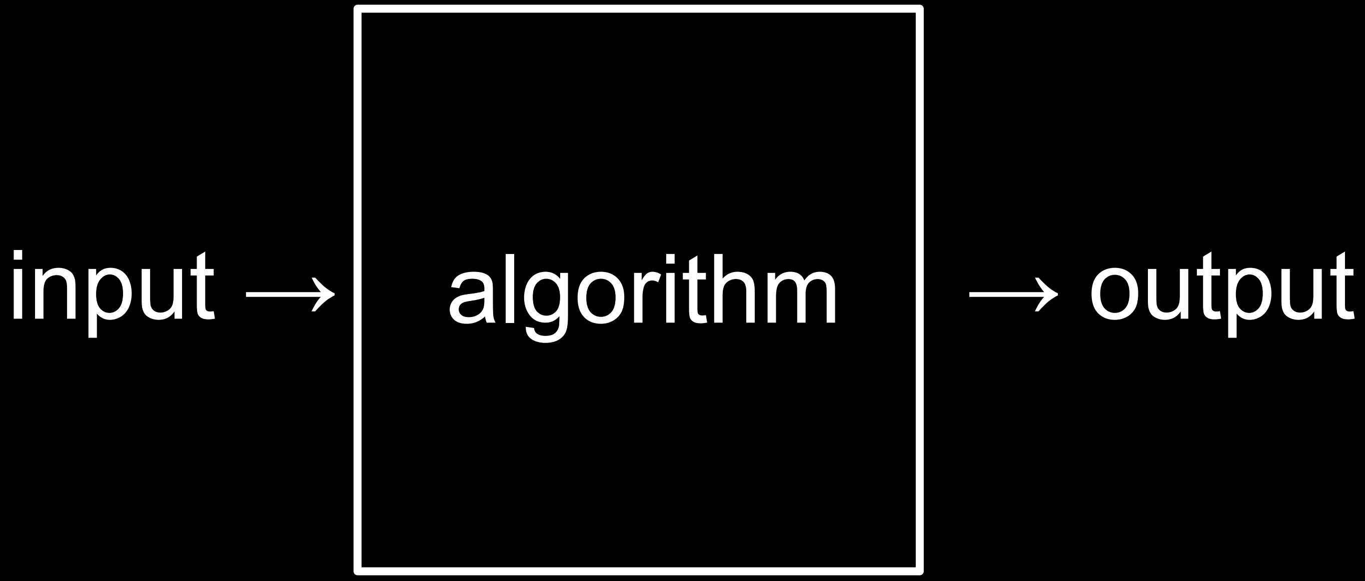


→ output

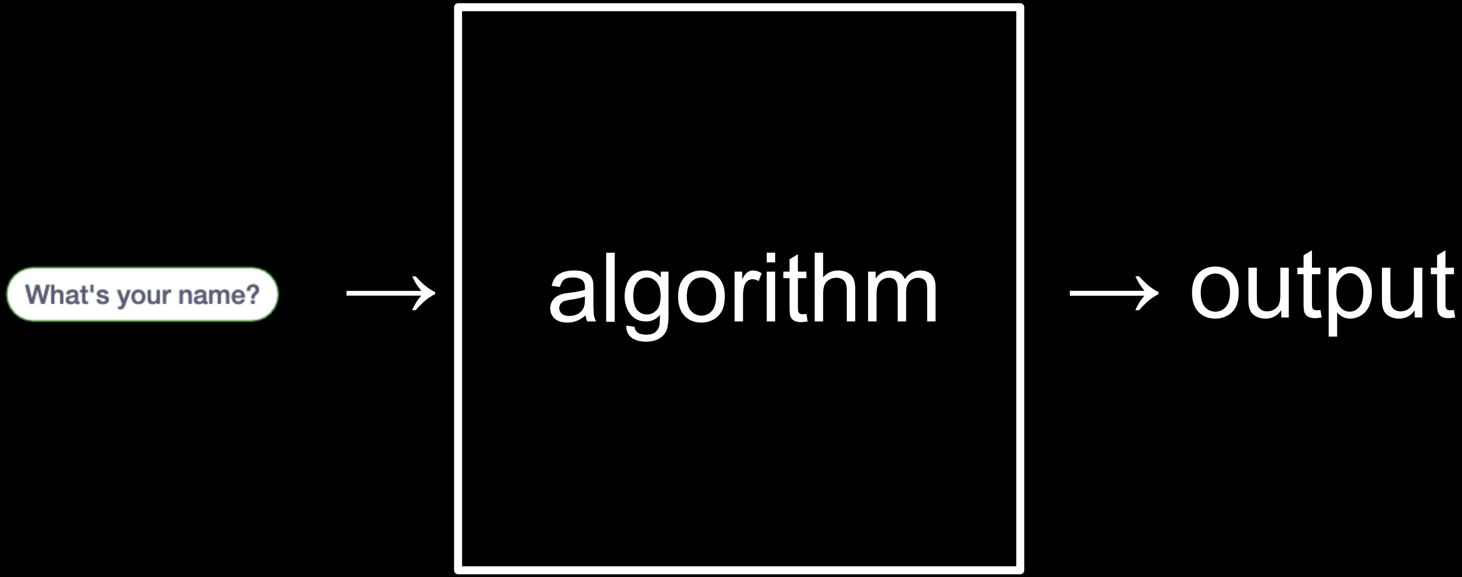

hello, world
say
hello, world



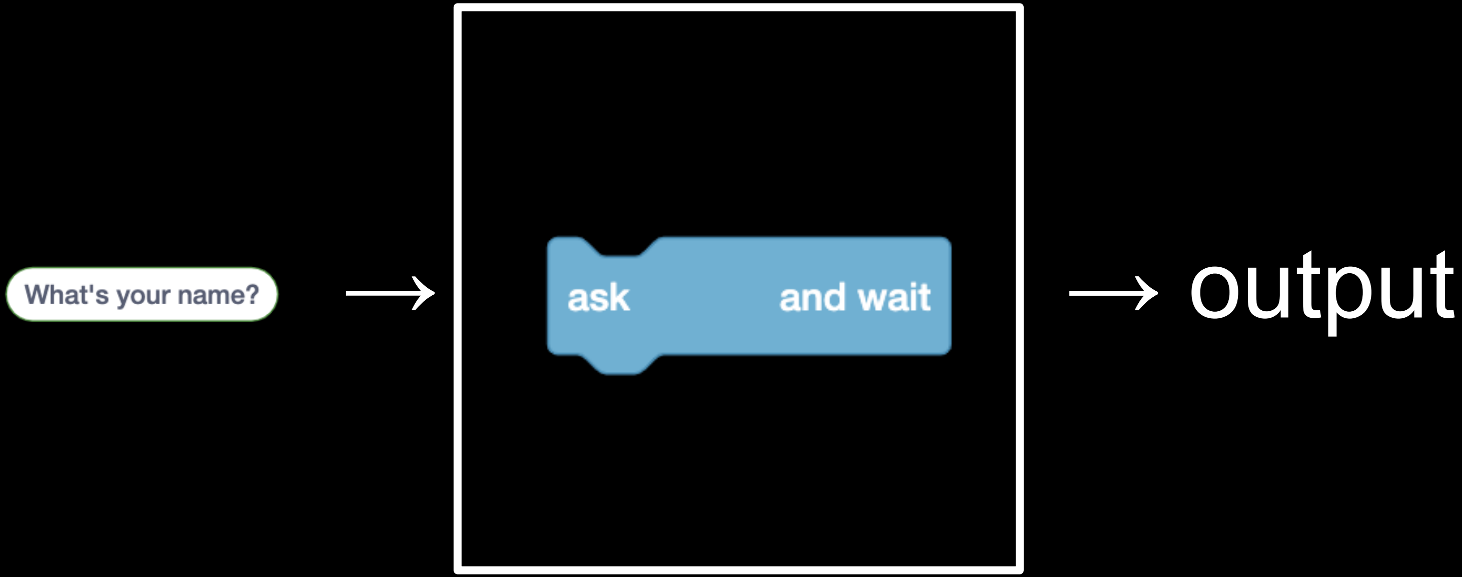
ask What's your name? and wait



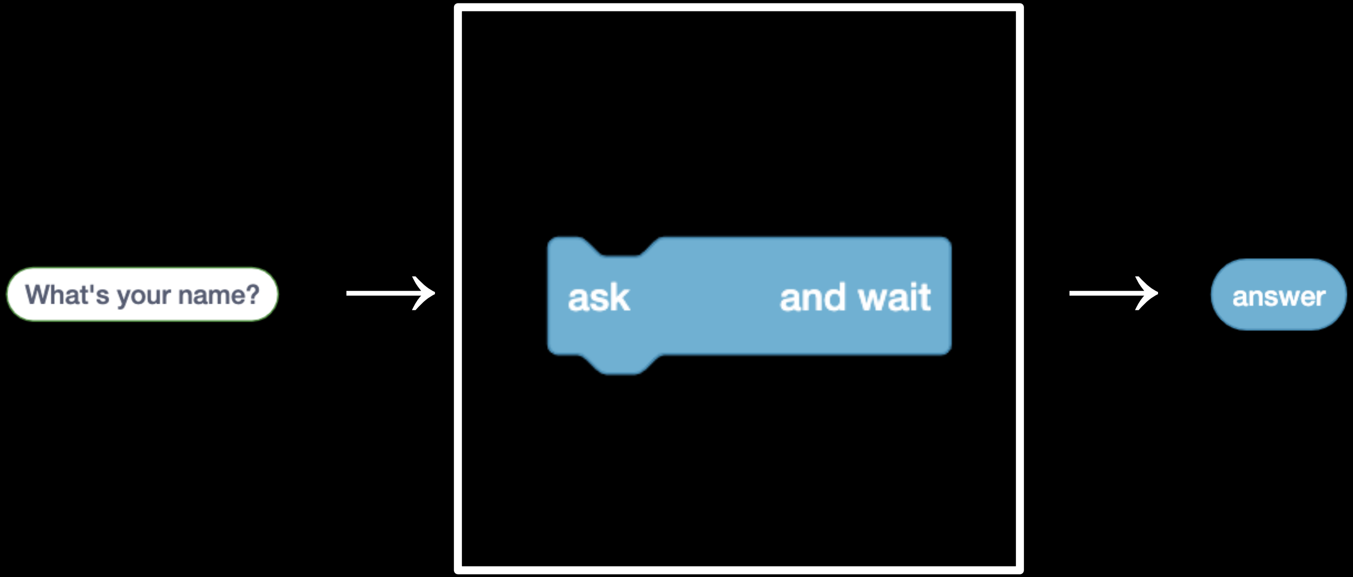
input → algorithm → output



What's your name? → algorithm → output



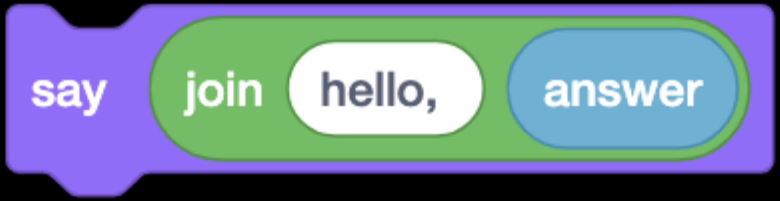
What's your name?
→
ask
and wait
→ output



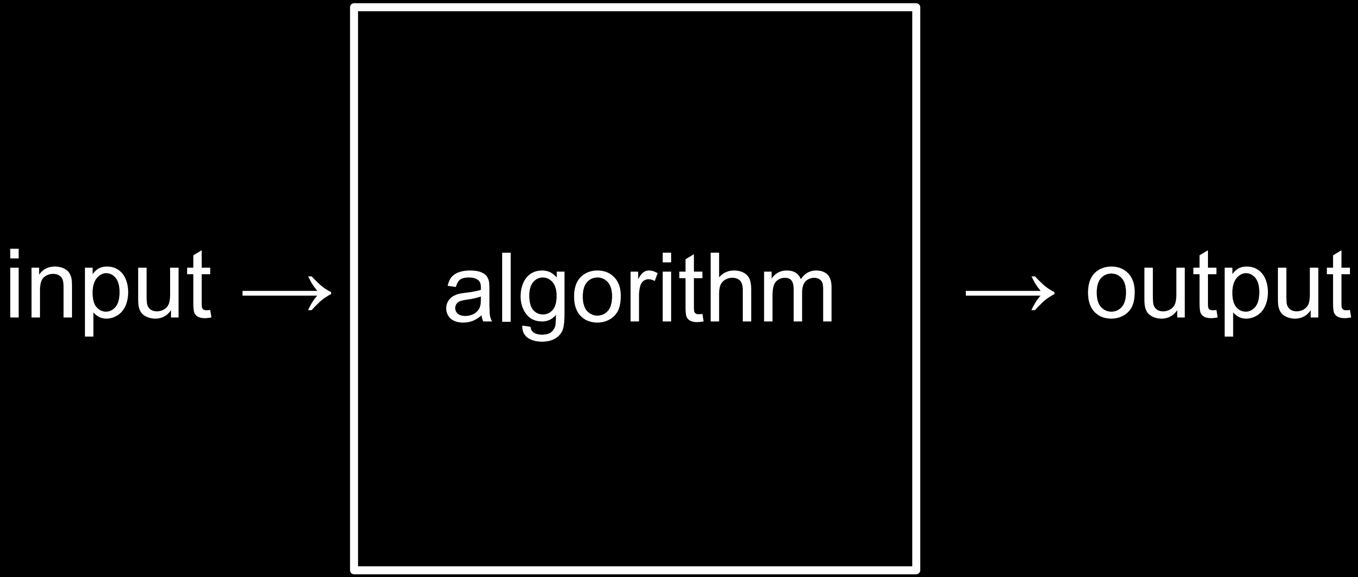
What's your name?

ask and wait

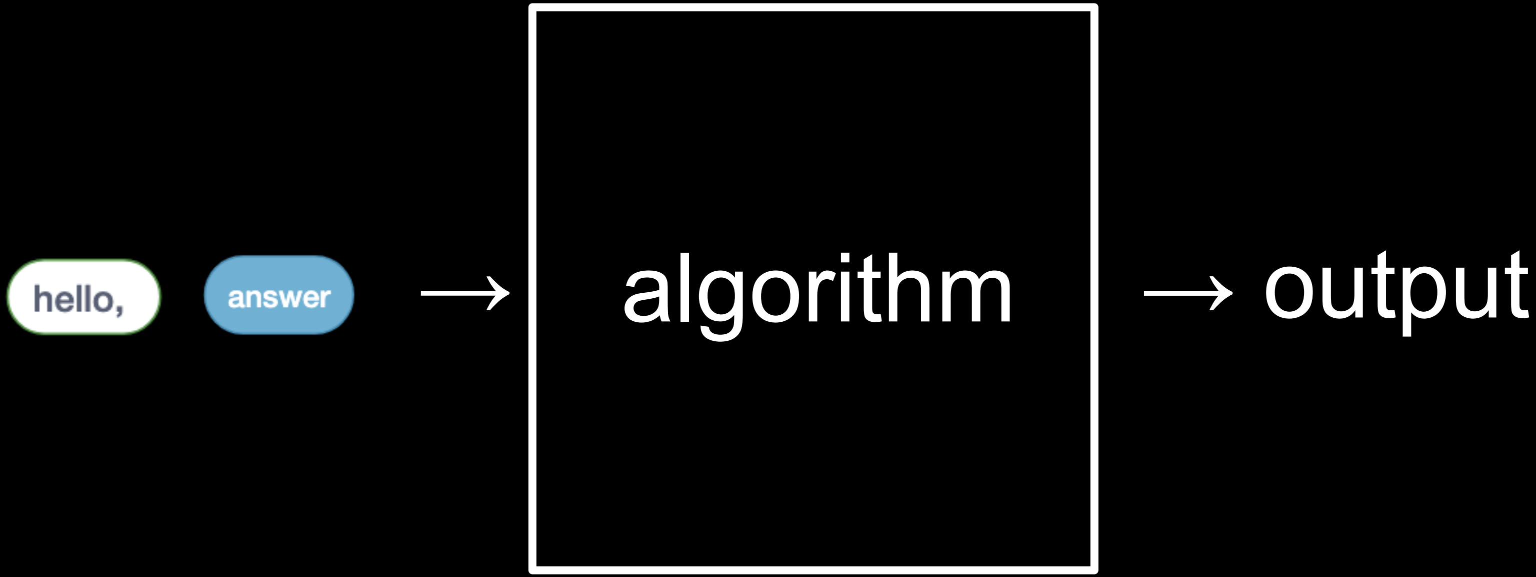
answer



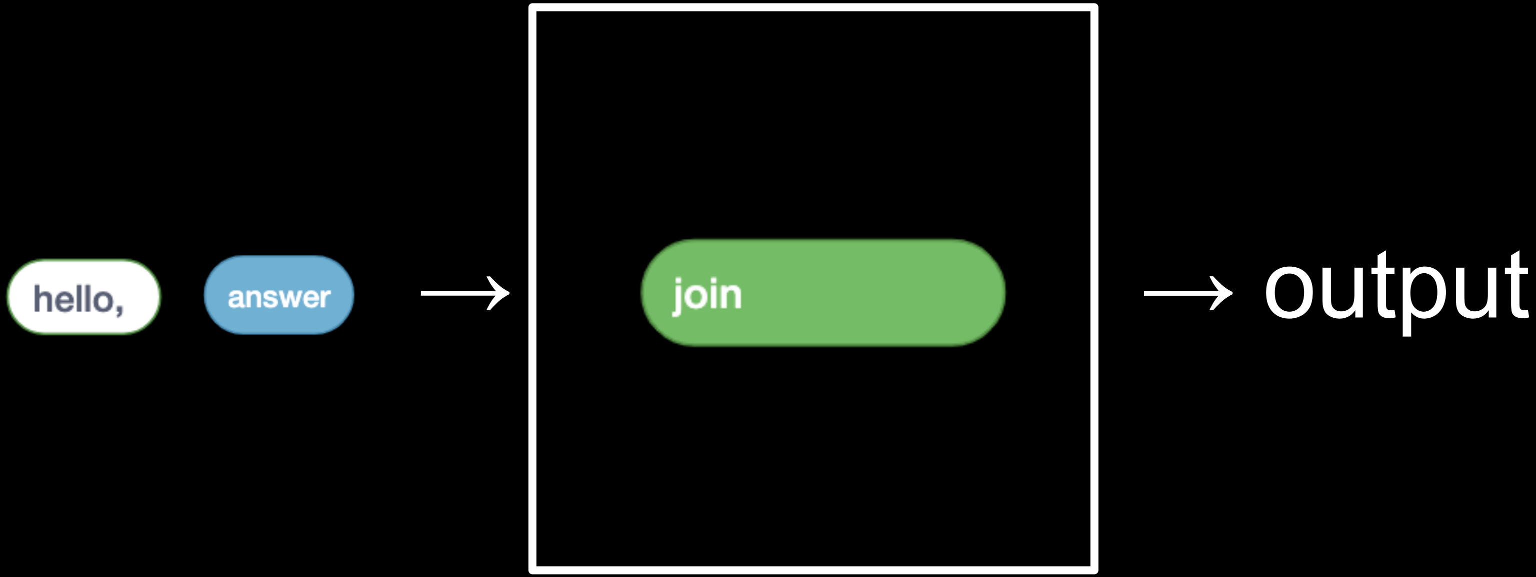
```
say (join (hello, ) (answer))
```



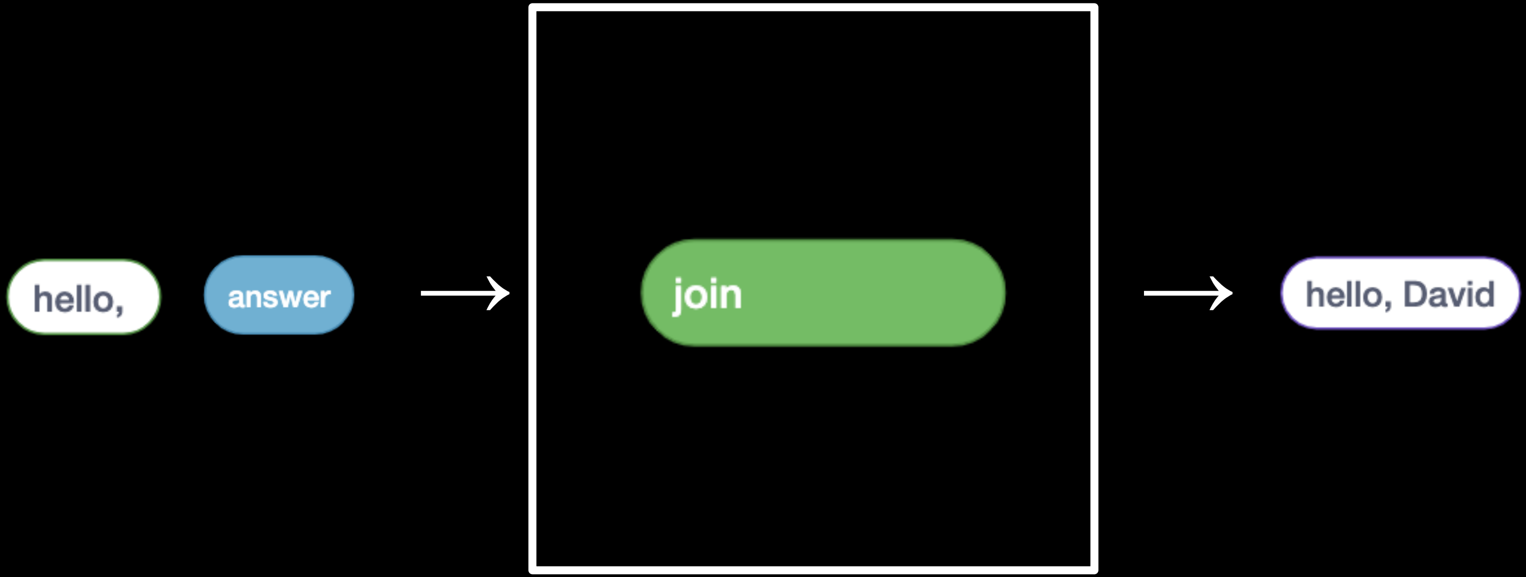
input → algorithm → output



hello, answer → algorithm → output



hello, answer → join → output



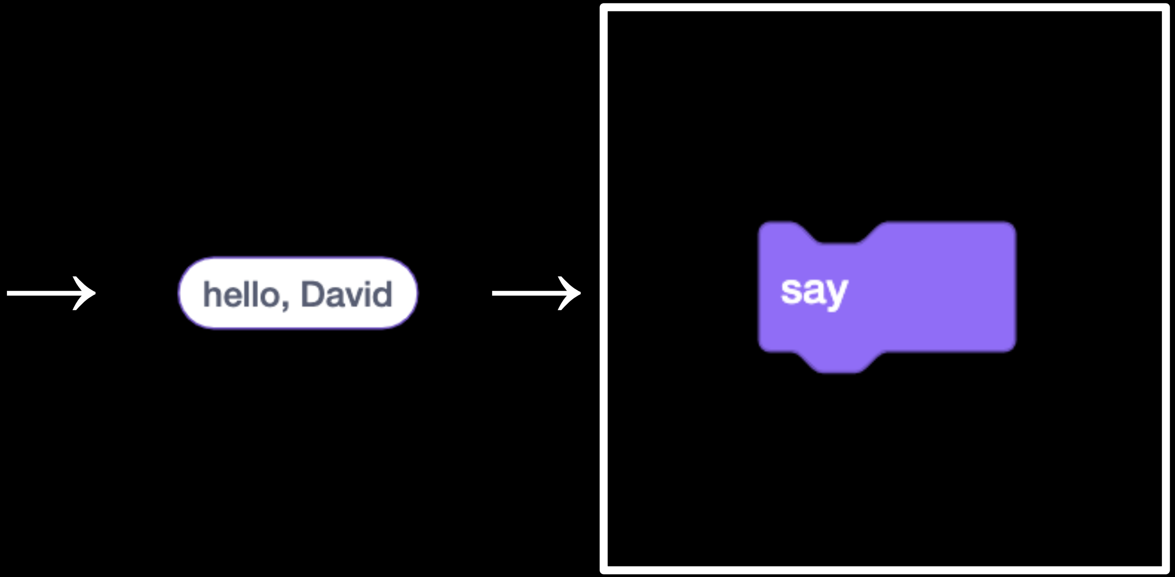
hello,
answer
join
hello, David

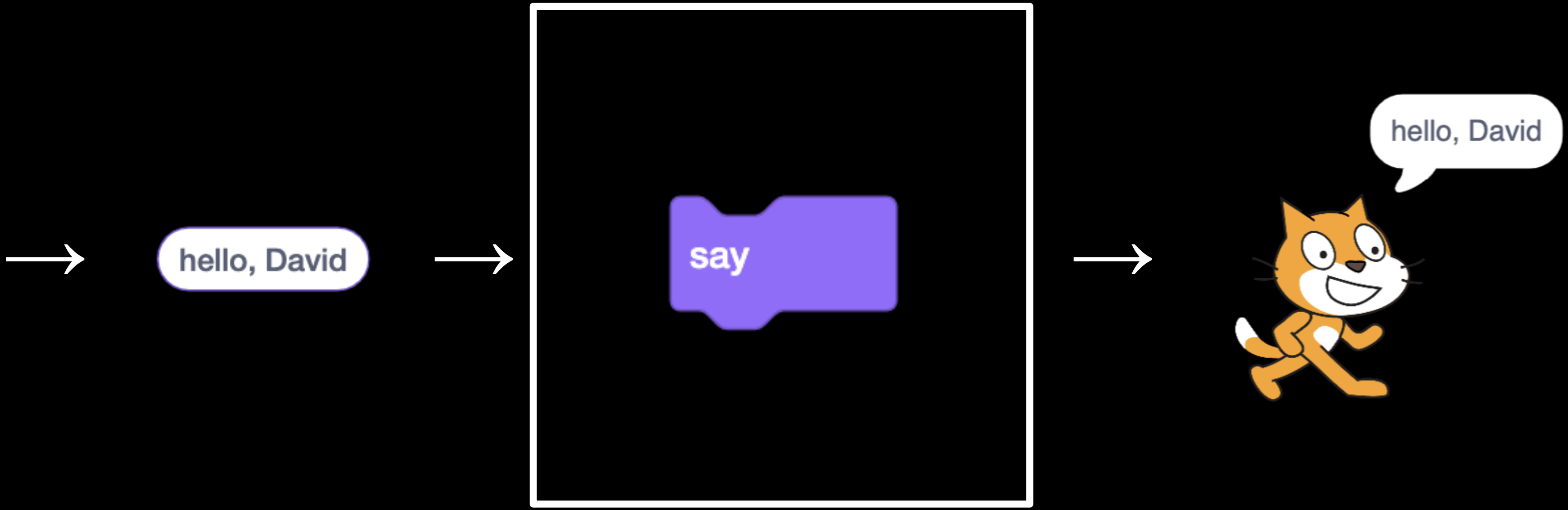
→

hello, David

→

hello, David


hello, David
say



hello, David

say

intervalo

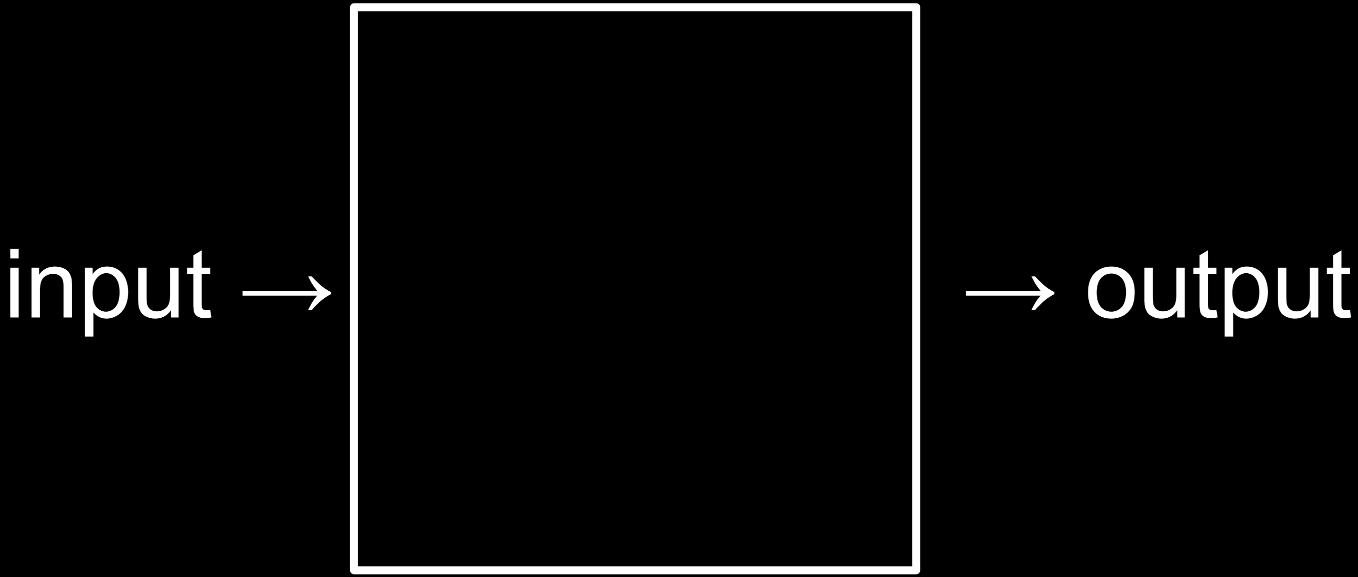
correctness

design

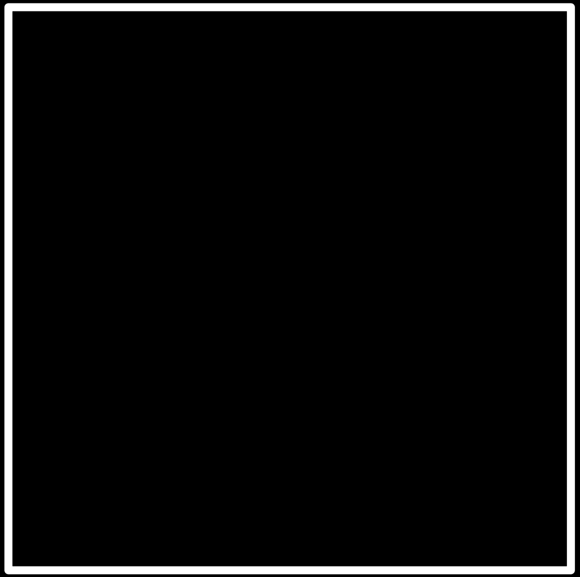
style

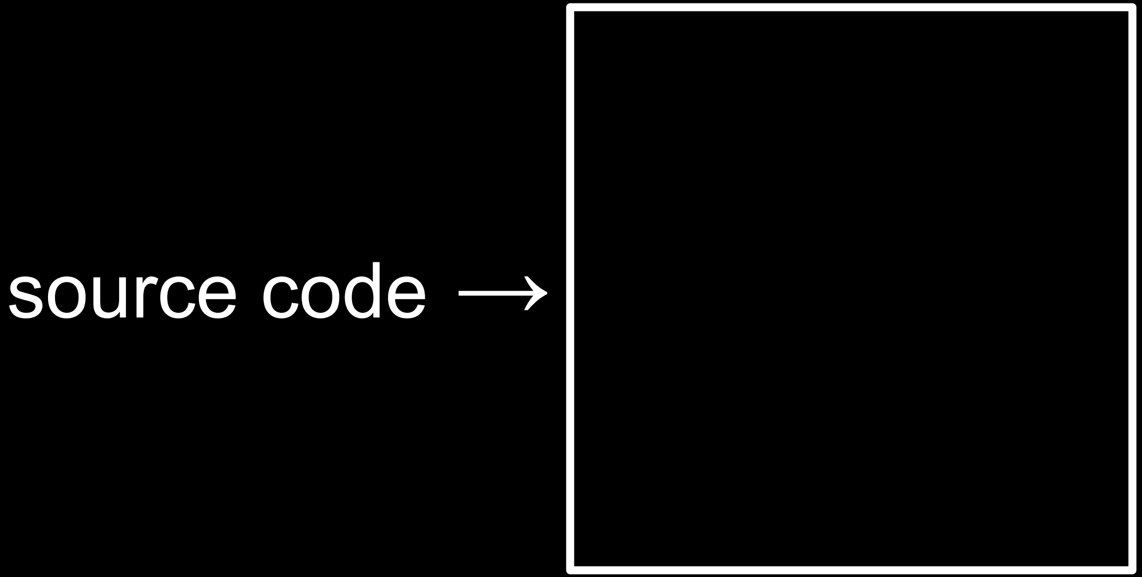
```
#include <stdio.h>

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

```
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
```

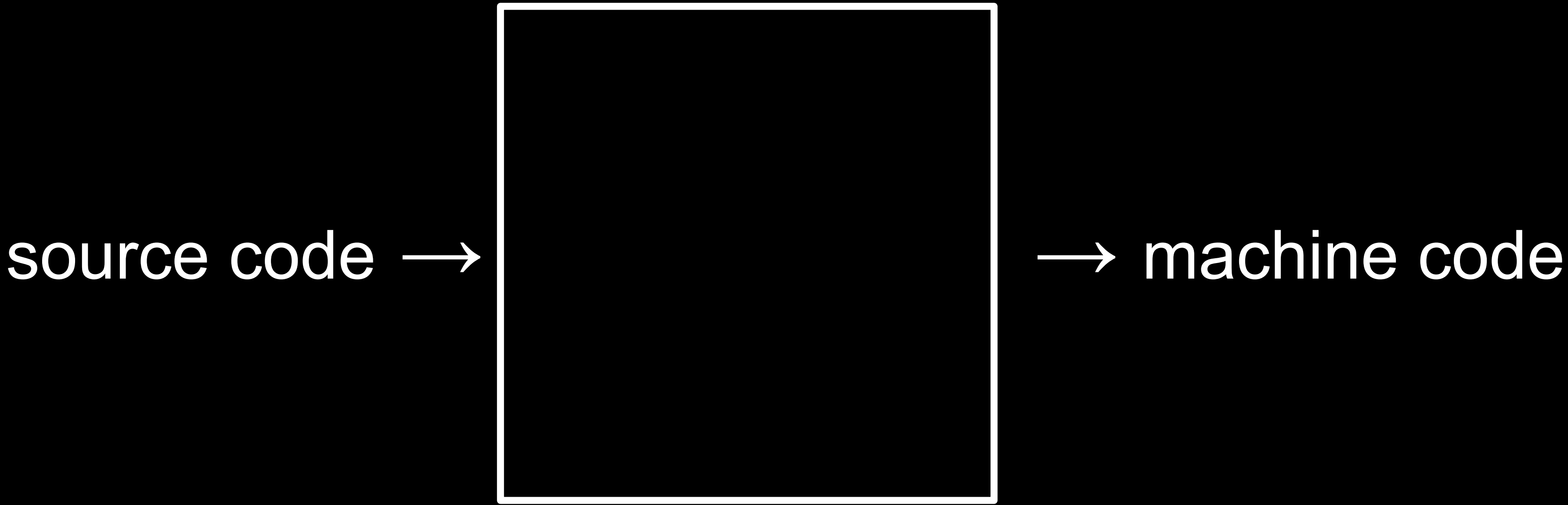


input → → output

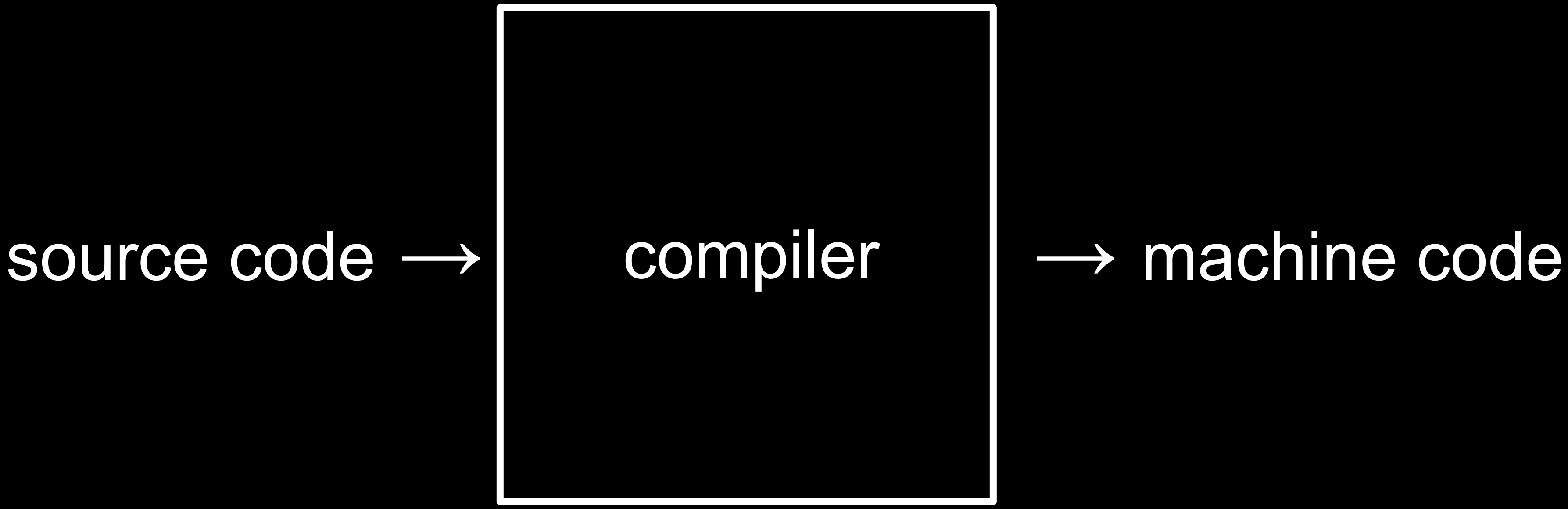




source code →



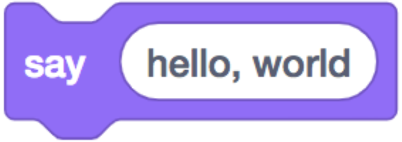
source code → → machine code



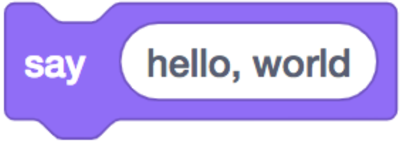
source code → compiler → machine code

```
make hello

./hello
```

functions, arguments

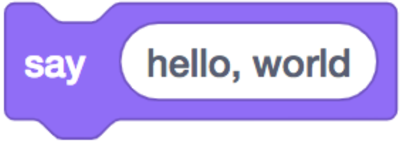


```
say hello, world
```

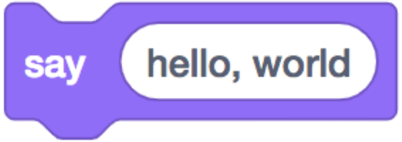


say hello, world

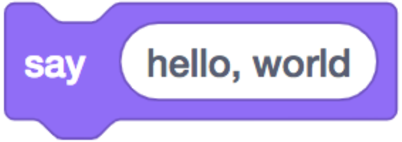
```
print (                    )
```



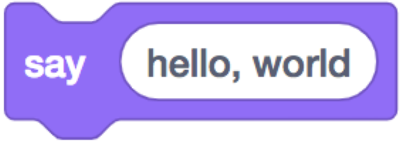
```
printf(                    )
```



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



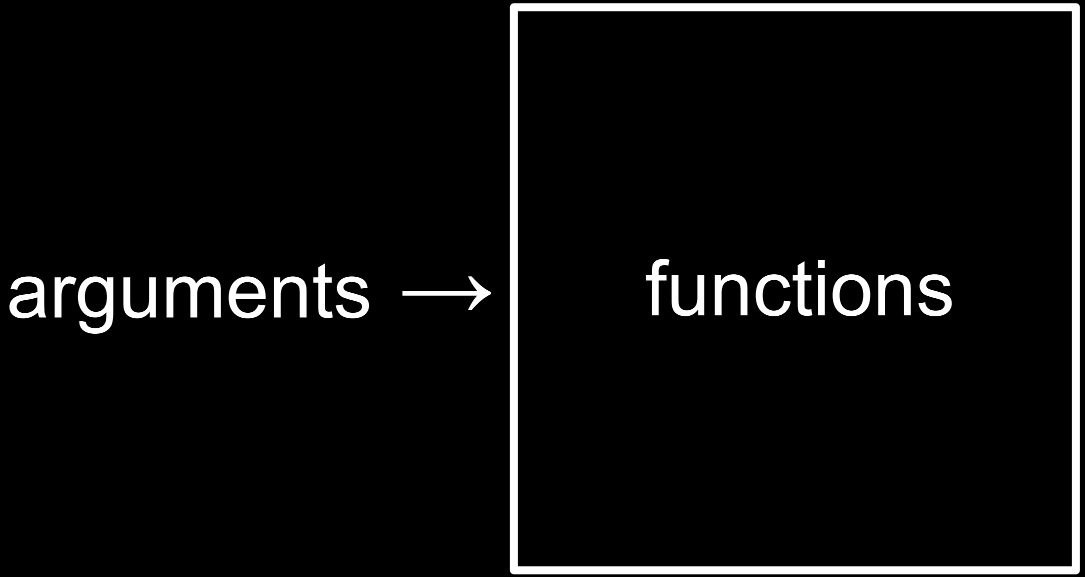
```
printf("hello, world")
```



say hello, world

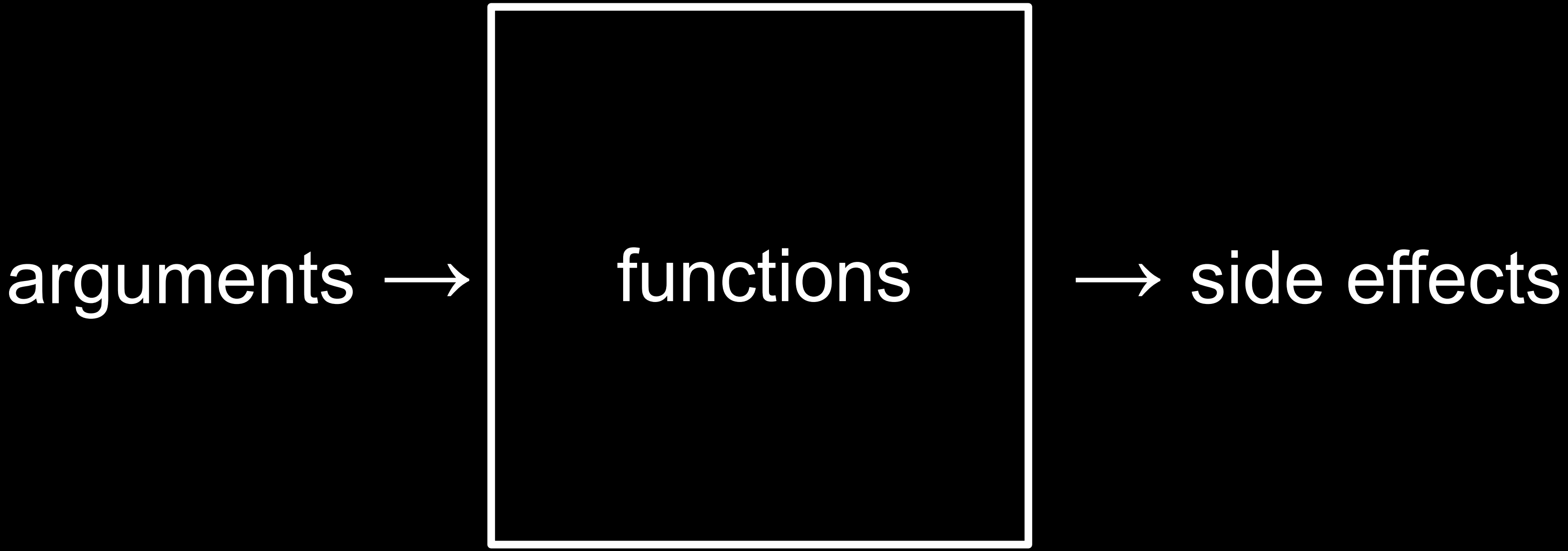
```
printf("hello, world");
```

functions

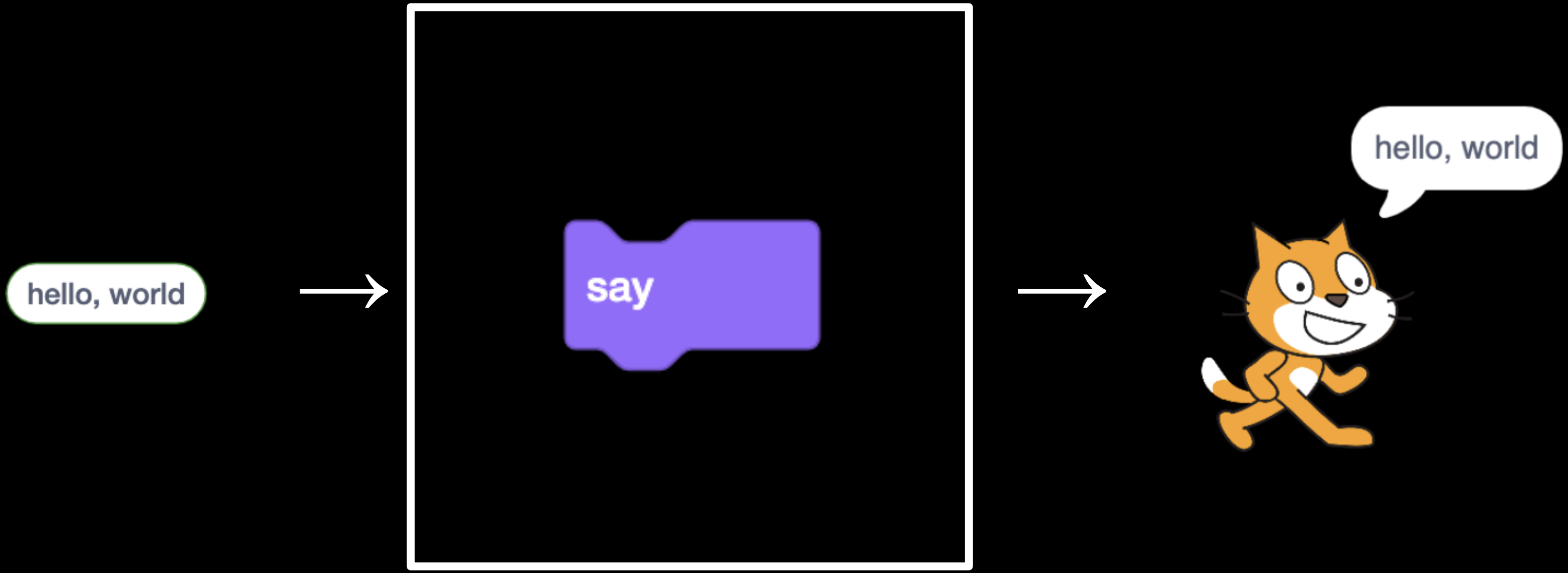


arguments →

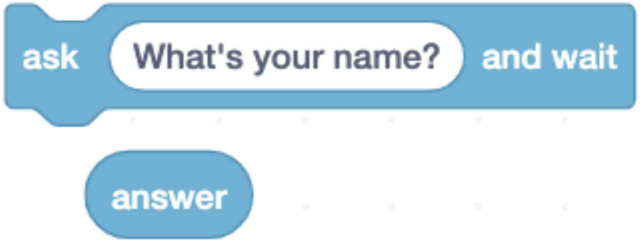
functions



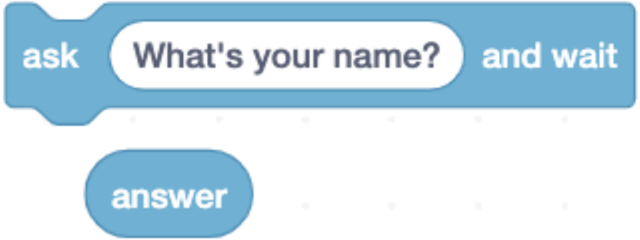
arguments → functions → side effects


hello, world
say
hello, world

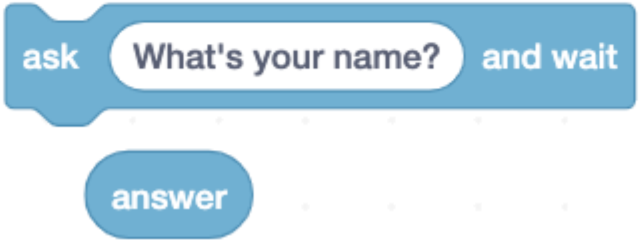
return values, variables



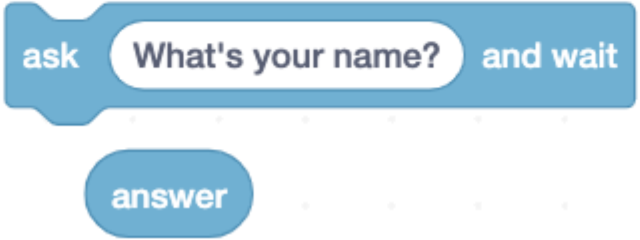
```
ask [What's your name?] and wait

(answer)
```



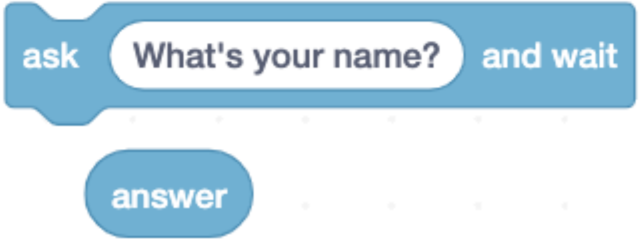
```
                    get_string(
)
```



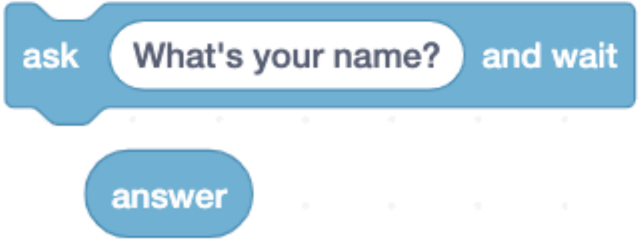
```
get_string("What's your name?")
```



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

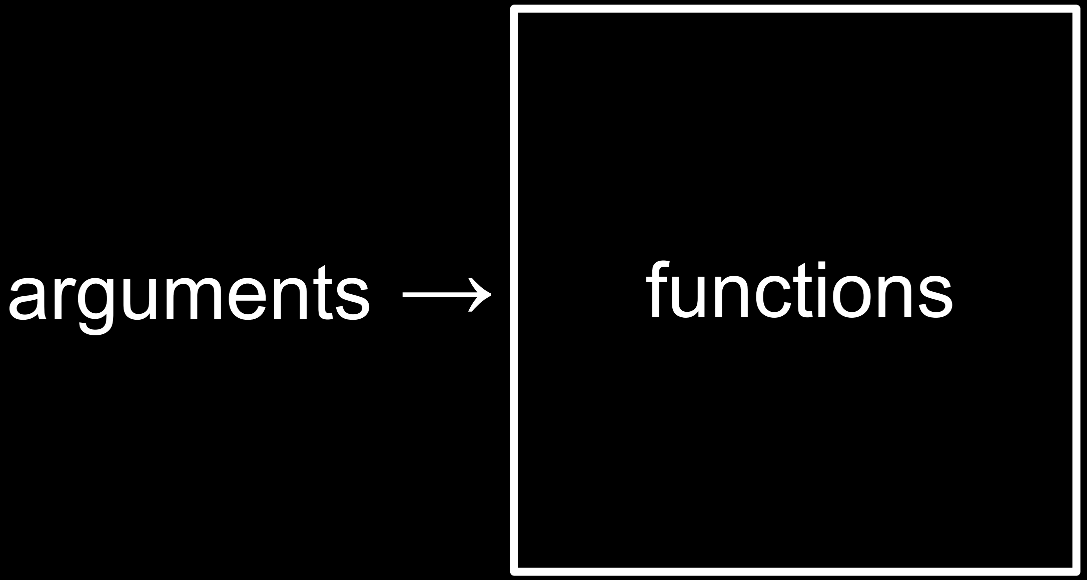


```
string answer = get_string("What's your name?
")
```



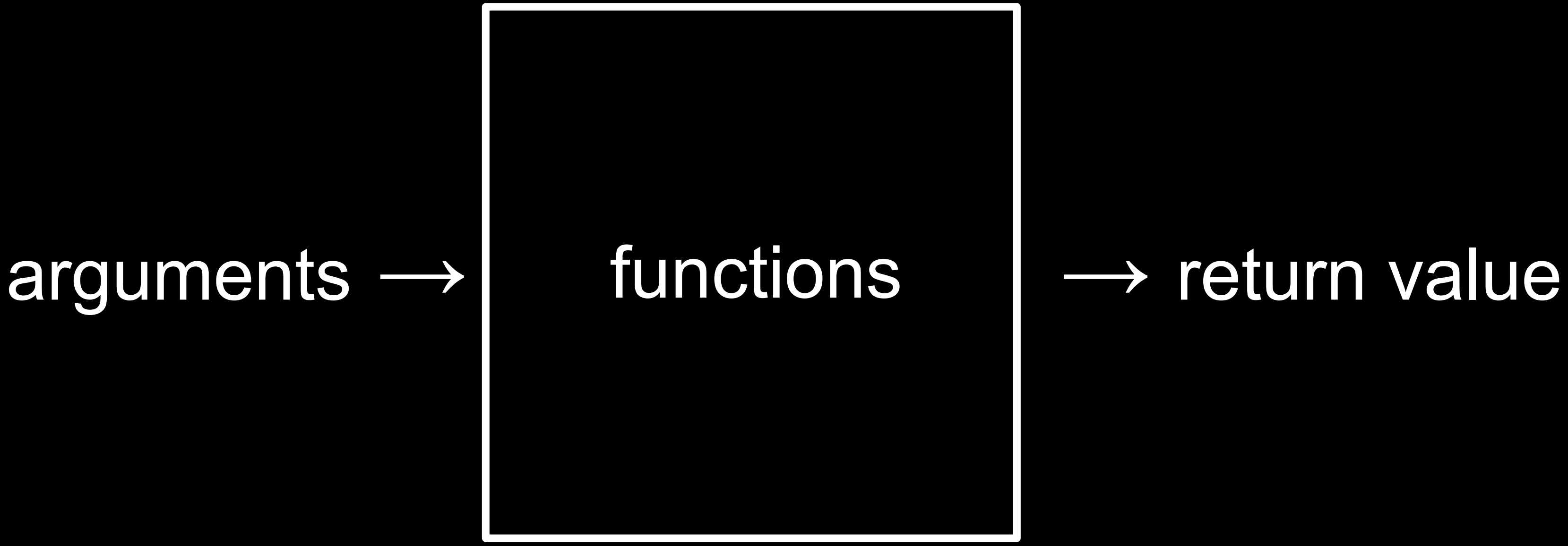
```
string answer = get_string("What's your name?
");
```

functions

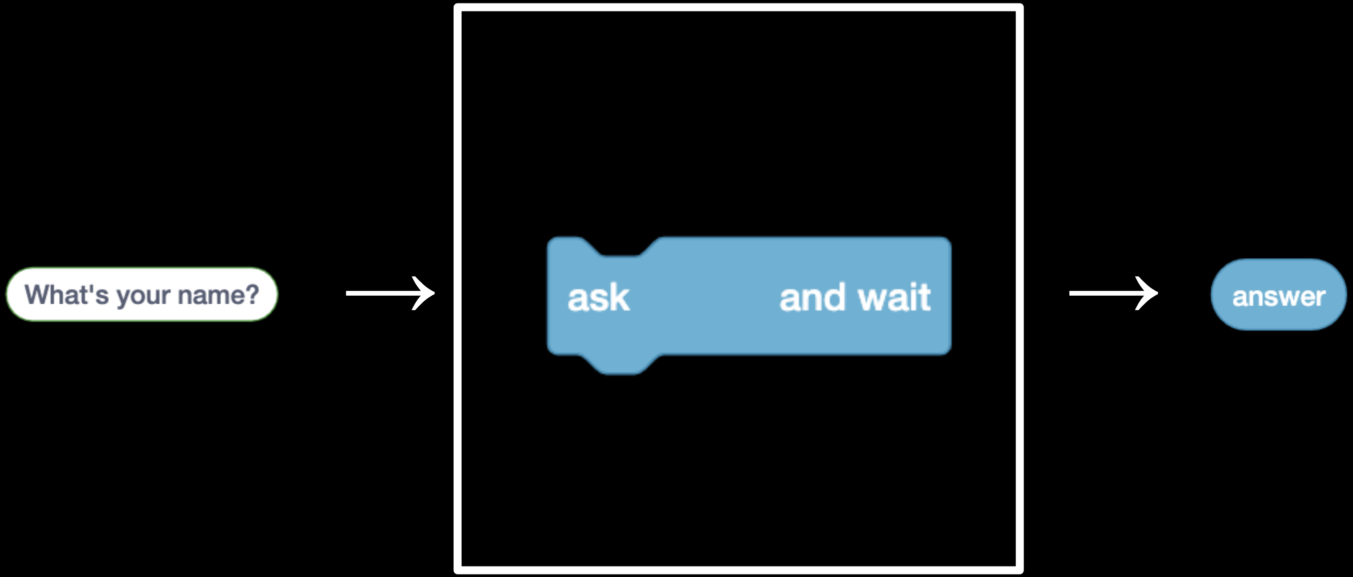


arguments →

functions



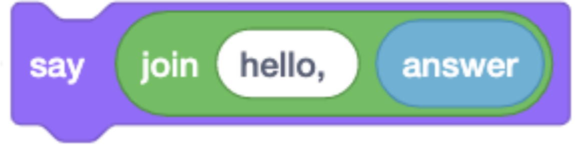
arguments → functions → return value



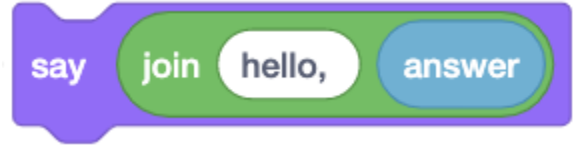
What's your name?

ask and wait

answer

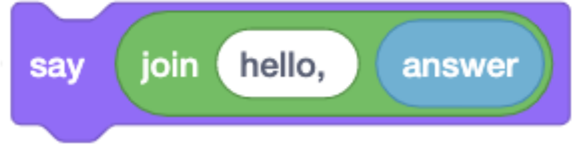


```
say join hello, answer
```



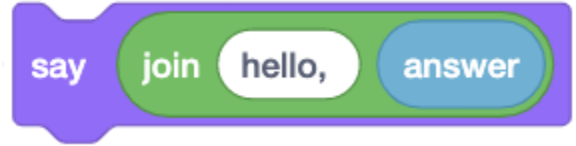
say join hello, answer

```
printf(                        );
```



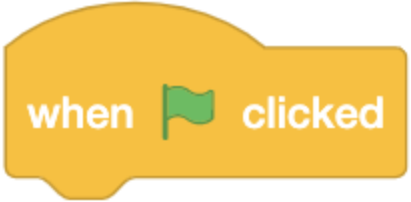
say join hello, answer

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

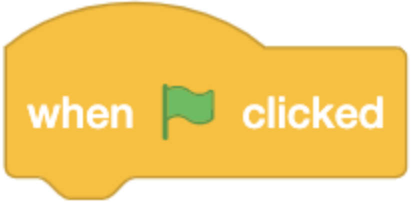


```
printf("hello, %s", answer);
```

main

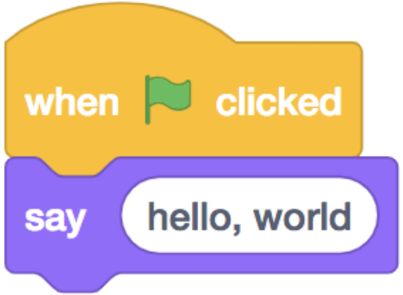


when ⚑ clicked



```
int main(void)
{

}
```

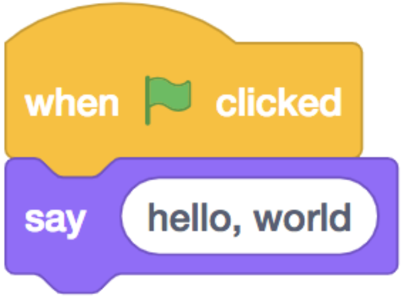
header files



when clicked

say hello, world

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



```
when 🏁 clicked
say hello, world
```

```
#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

+

−

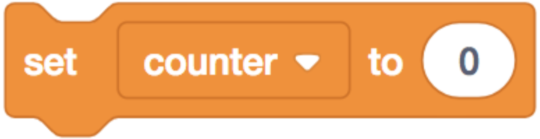
*

/

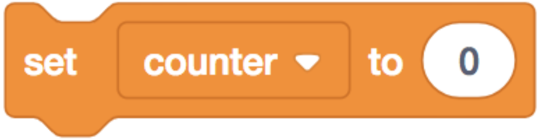
%

+	addition
-	subtraction
*	multiplication
/	division
%	remainder

variables, syntactic sugar

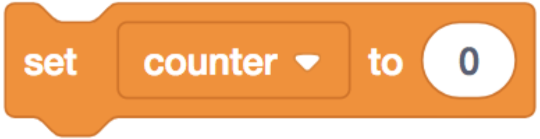


```
set counter ▾ to 0
```



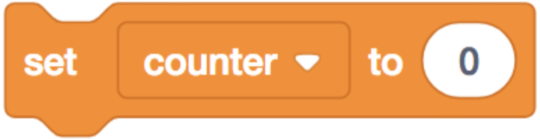
set counter ▾ to 0

```
counter = 0
```



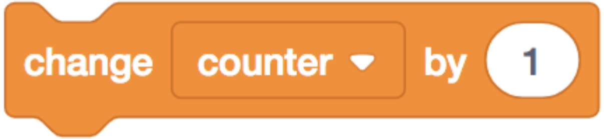
set counter ▾ to 0

```
int counter = 0
```

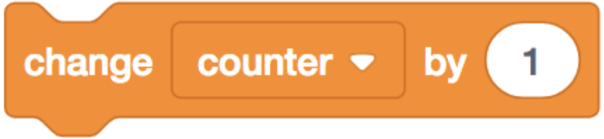


set counter ▾ to 0

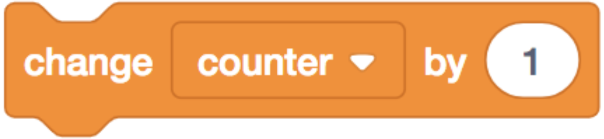
```
int counter = 0;
```



change counter ▾ by 1

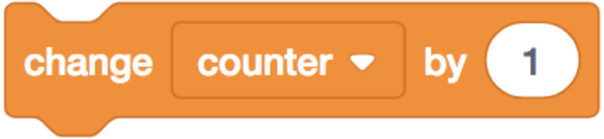


```
counter = counter + 1
```



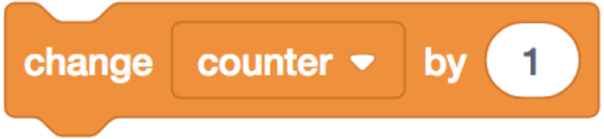
change counter ▾ by 1

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



change counter ▾ by 1

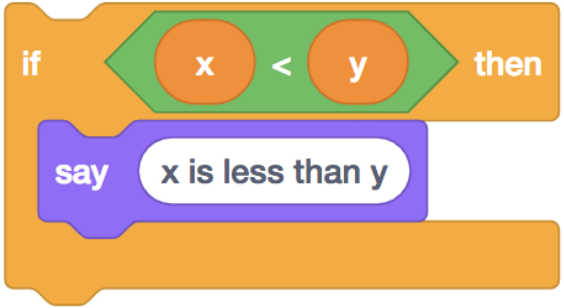
```
counter += 1;
```



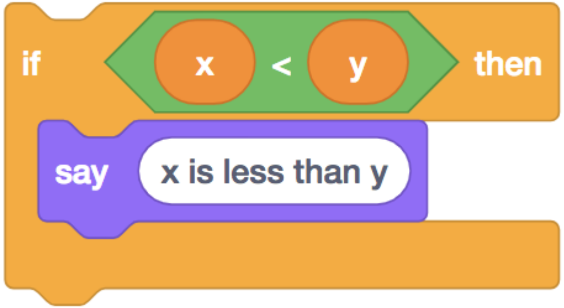
change counter ▾ by 1

```
counter++;
```

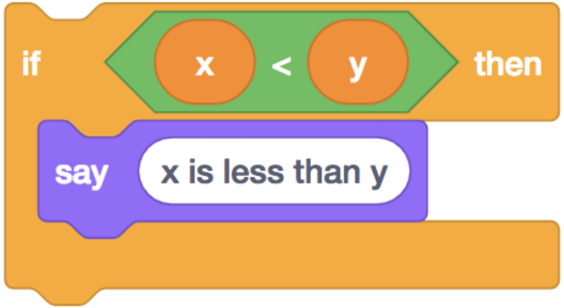
conditionals



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

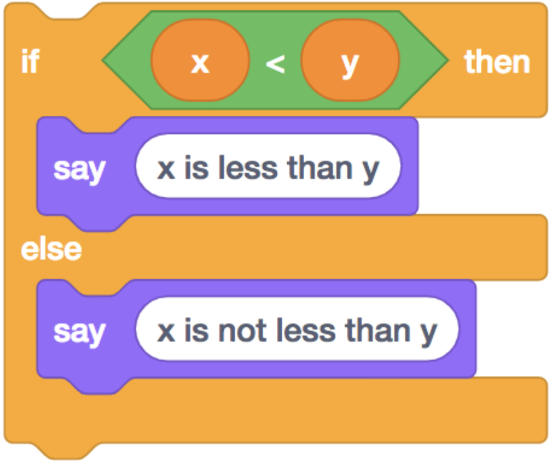


```
if (x < y)
{



}
```

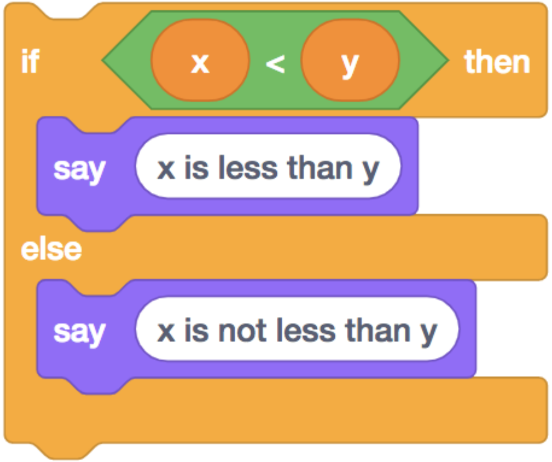


```
if x < y then
    say x is less than y
```

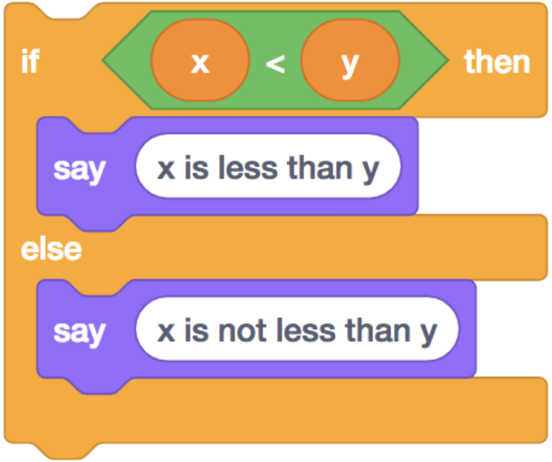
```
if (x < y)
{
    printf("x is less than
y\n");
}
```



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

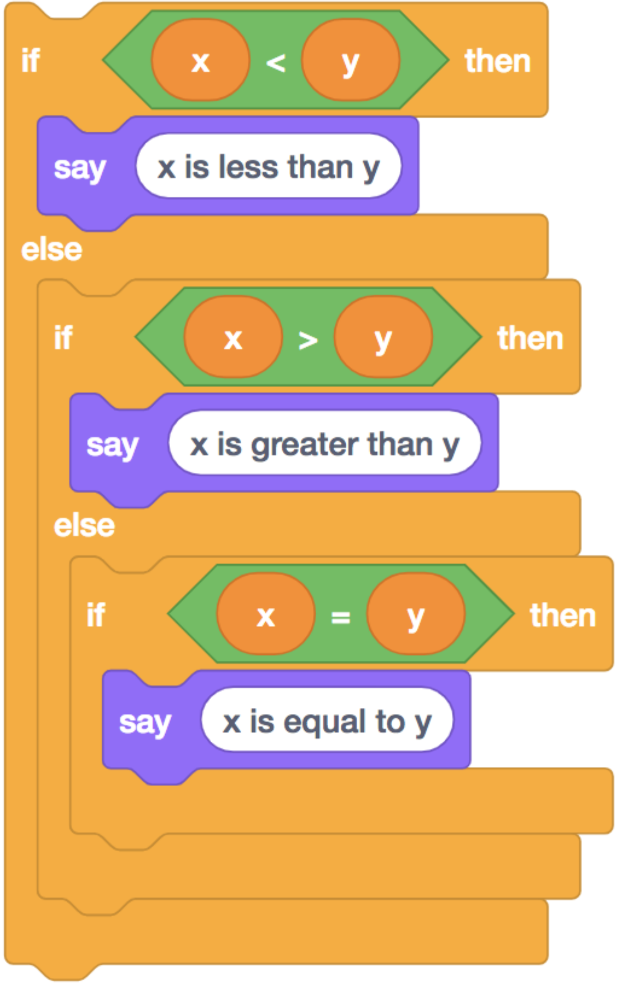


```
if (x < y)
{

}
else
{


}
```

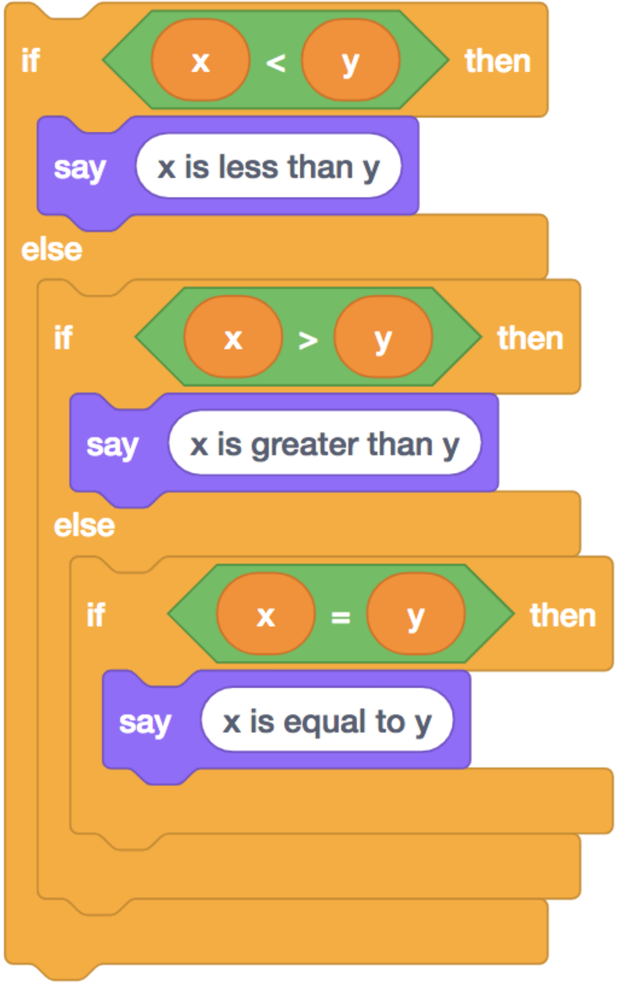


```
if x < y then
    say x is less than y
else
    say x is not less than y
```

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

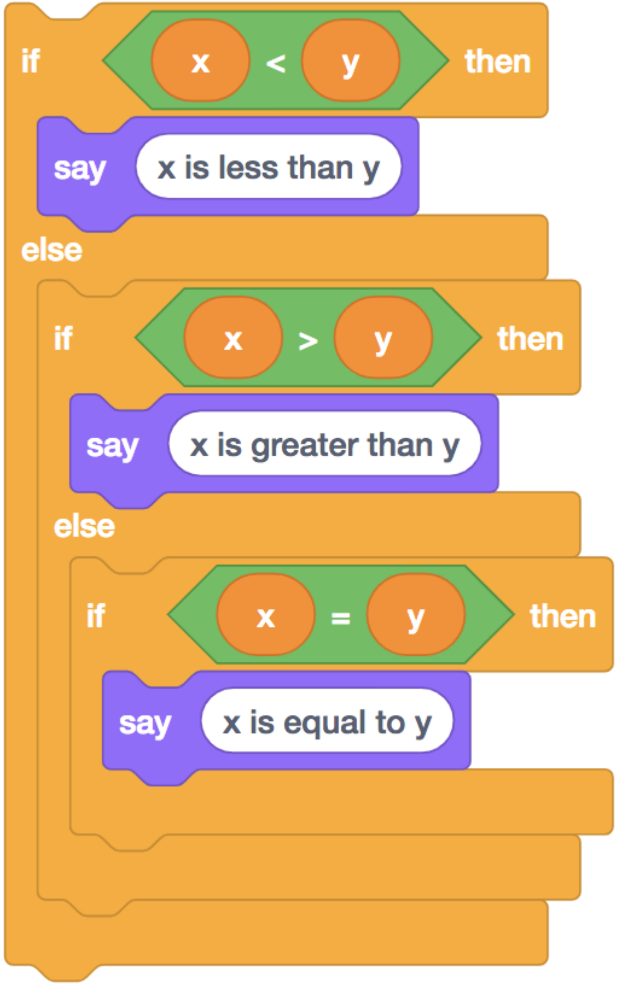


```
if x < y then
    say x is less than y
else
    if x > y then
        say x is greater than y
    else
        if x = y then
            say x is equal to y
```



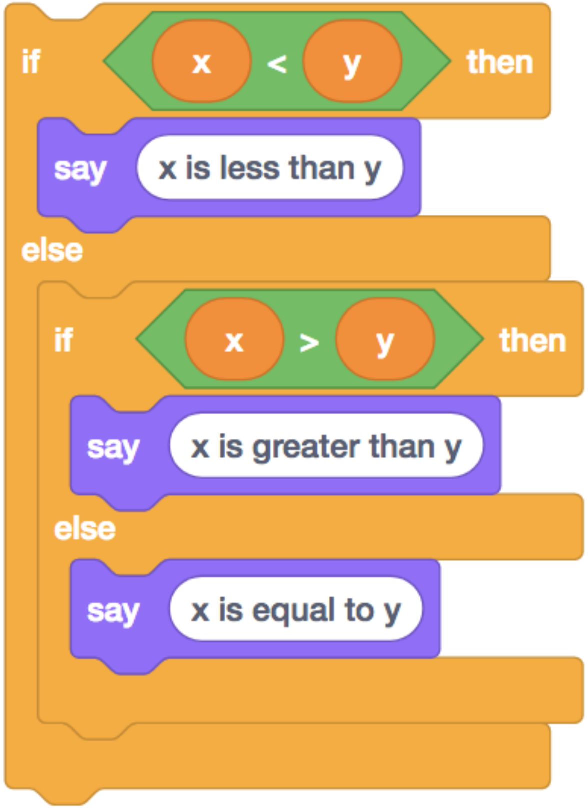
```
if x < y then
  say x is less than y
else
  if x > y then
    say x is greater than y
  else
    if x = y then
      say x is equal to y
```

```
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
```

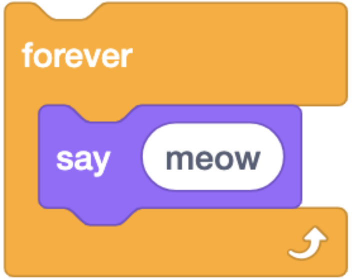
```
if (x < y)
{
    printf("x is less than y\n");
}
else if (x > y)
{
    printf("x is greater than
y\n");
}
else if (x == y)
{
    printf("x is equal to y\n");
}
```



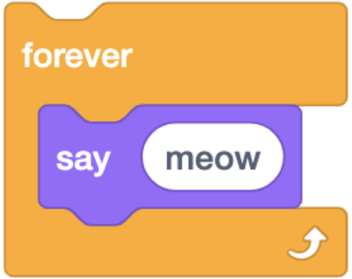
```
if x < y then
    say x is less than y
else
    if x > y then
        say x is greater than y
    else
        say x is equal to y
```

```
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");
}
```

loops



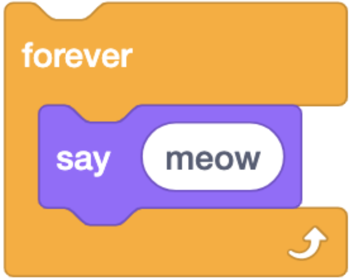
```
forever
    say meow
```



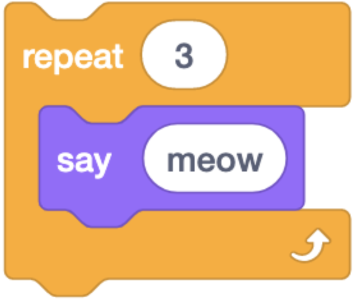
forever

say meow

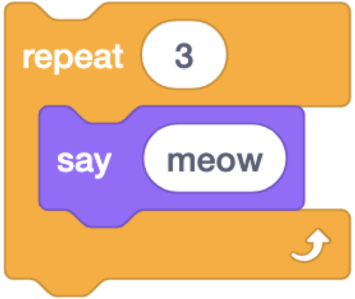
```
while (true)
{

}
```



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

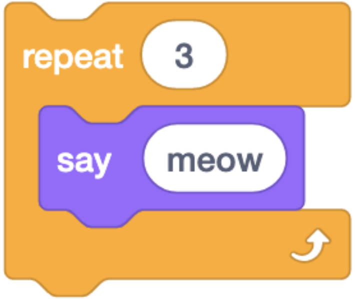


```
repeat 3
    say meow
```

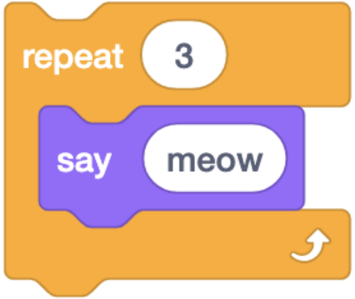


```
repeat 3
  say meow
```

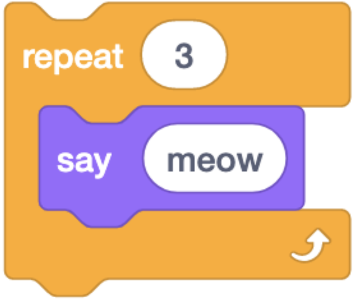
```
int counter = 0;
while (counter < 3)
{


}
```



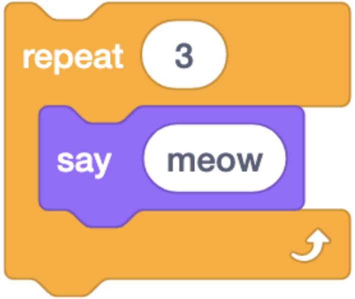
```
int counter = 0;
while (counter < 3)
{
    printf("meow\n");

}
```



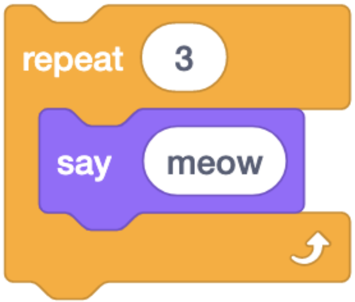
```
int counter = 0;
while (counter < 3)
{
    printf("meow\n");
    counter = counter + 1;
}
```



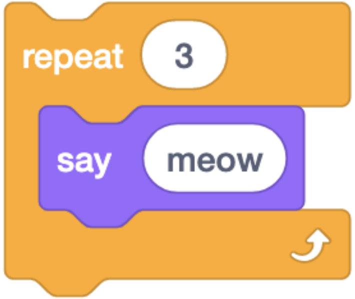
```
int i = 0;
while (i < 3)
{
    printf("meow\n");
    i = i + 1;
}
```



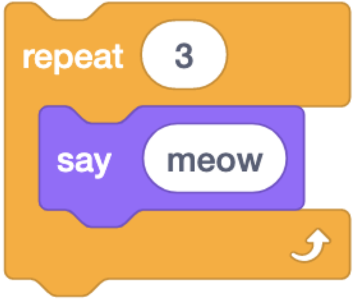
```
int i = 0;
while (i < 3)
{
    printf("meow\n");
    i += 1;
}
```



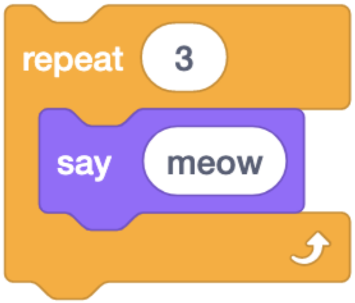
```
int i = 0;
while (i < 3)
{
    printf("meow\n");
    i++;
}
```



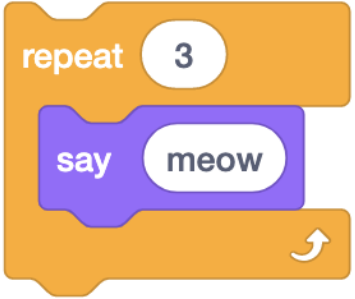
```
int i = 0;
while (i < 3)
{
    printf("meow\n");
    i++;
}
```



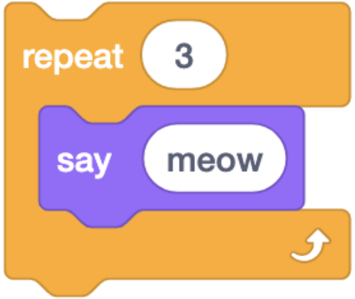
```
int i = 0;
while (i < 3)
{
    printf("meow\n");
    i++;
}
```



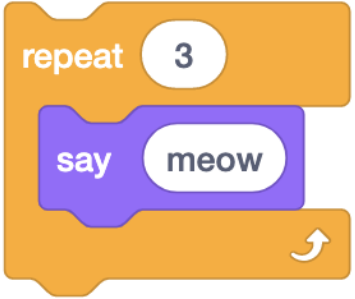
```
int i = 0;
while (i < 3)
{
    printf("meow\n");
    i++;
}
```



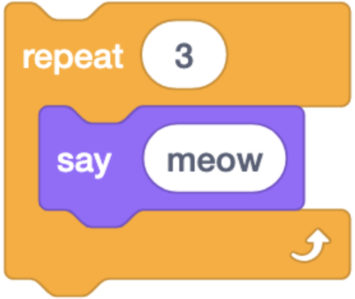
```
int i = 0;
while (i < 3)
{
    printf("meow\n");
    i++;
}
```



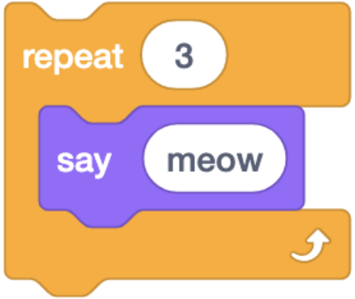
```
int i = 0;
while (i < 3)
{
    printf("meow\n");
    i++;
}
```



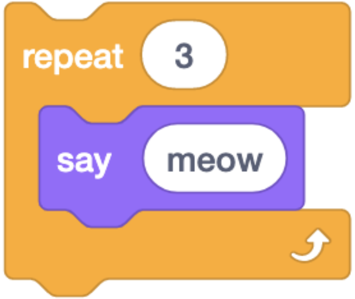
```
int i = 0;
while (i < 3)
{
    printf("meow\n");
    i++;
}
```



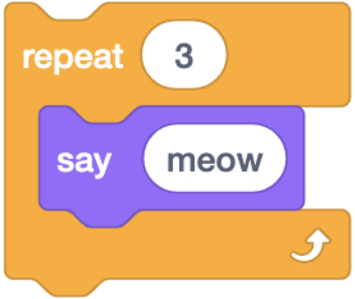
```
int i = 0;
while (i < 3)
{
    printf("meow\n");
    i++;
}
```



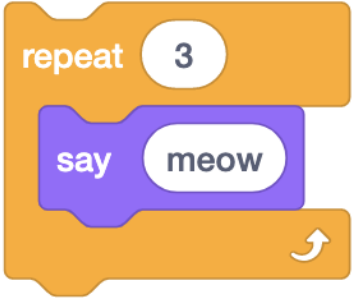
```
int i = 1;
while (i <= 3)
{
    printf("meow\n");
    i++;
}
```



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



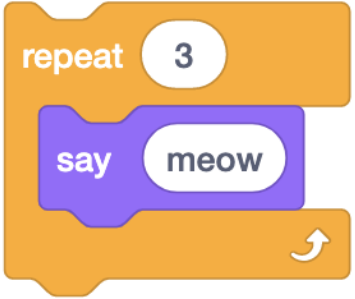
```
repeat 3
    say meow
```



repeat 3

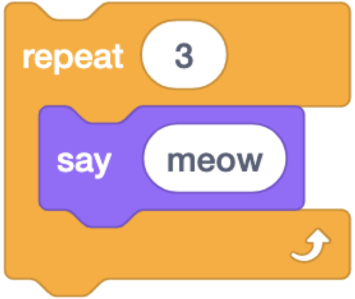
say meow

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

}
```

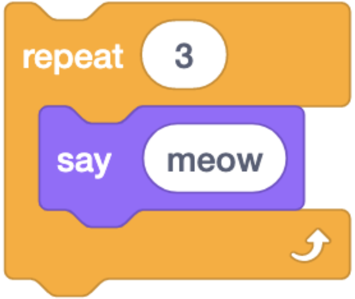


```
repeat 3
    say meow
```

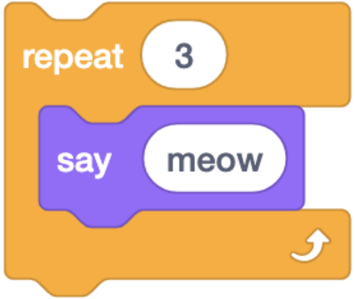
```
for (int i = 0; i < 3; i++)
{
    printf("meow\n");
}
```



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

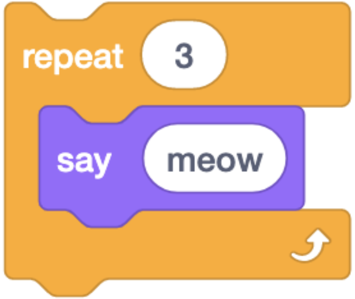


```
for (int i = 0; i < 3; i++)
{
    printf("meow\n");
}
```

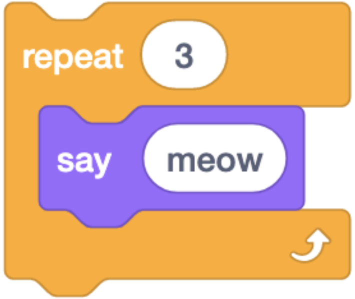


```
repeat 3
    say meow
```

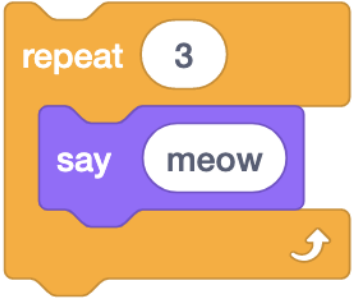
```
for (int i = 0; i < 3; i++)
{
    printf("meow\n");
}
```



```
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");
}
```