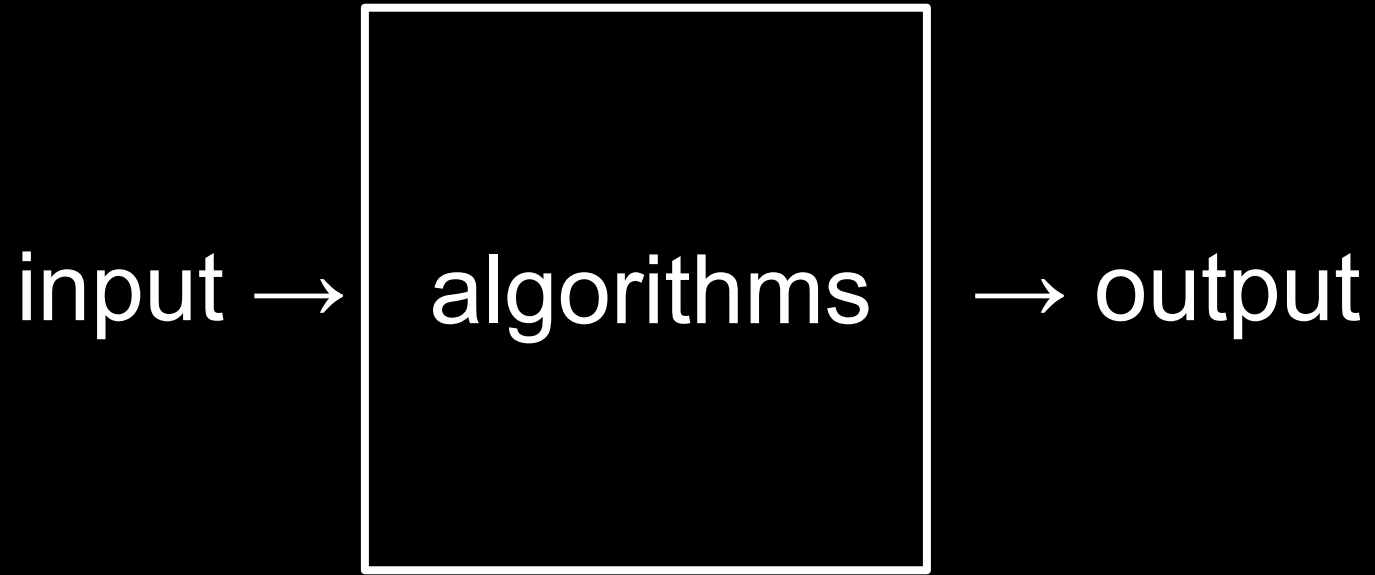


A purple Scratch 'say' block with a notch on the left and a bump on the right. It contains the text 'say' and 'hello, world' in a white rounded rectangle.

say

hello, world



hello, world

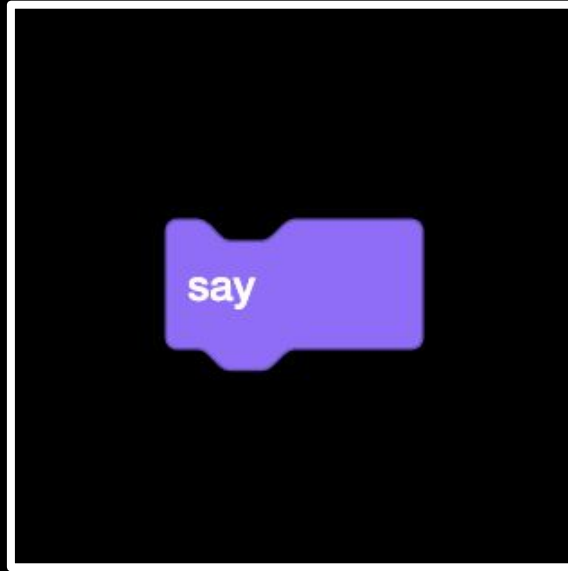


algorithms



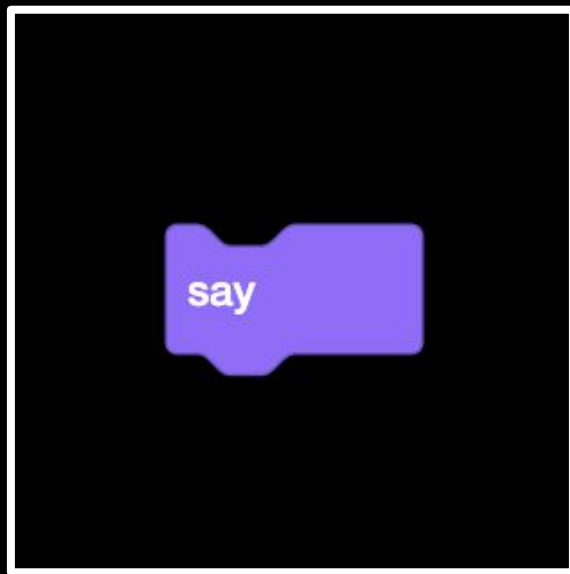
output

hello, world



→ output

hello, world

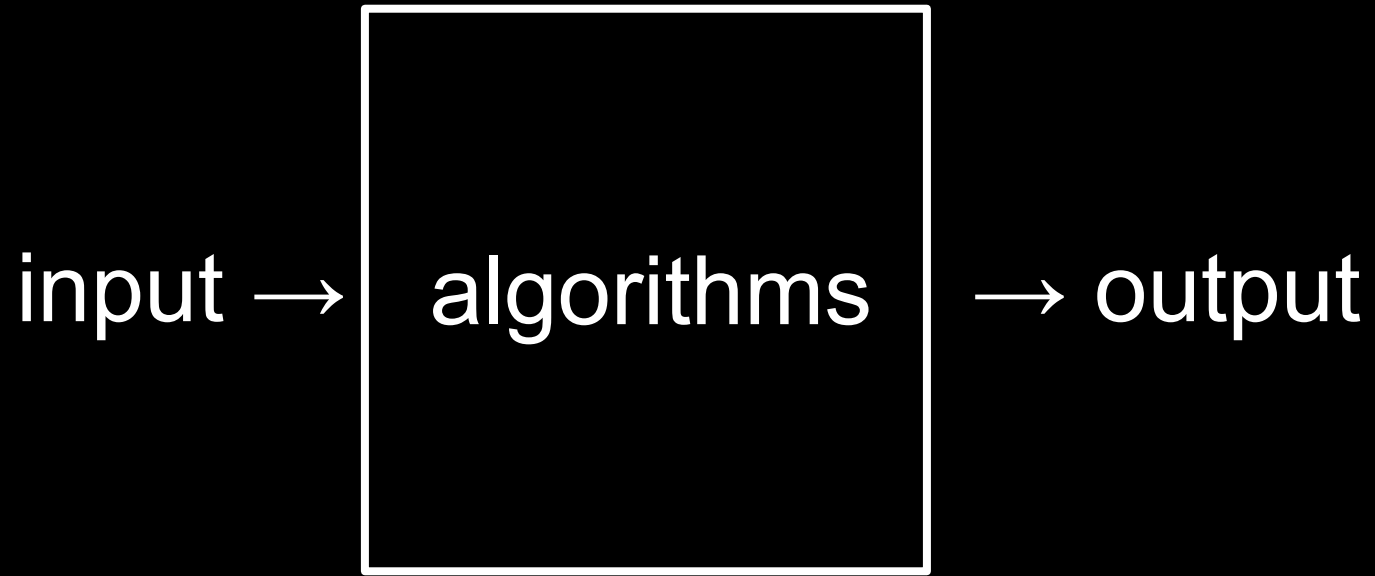


A blue Scratch 'ask and wait' block with a notch on the left and a bump on the right. It contains a white text input field with the text 'What's your name?'.

ask

What's your name?

and wait



What's your name?

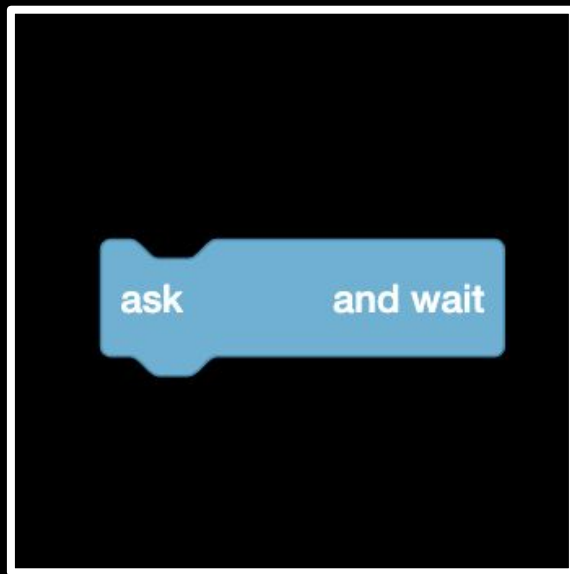


algorithms

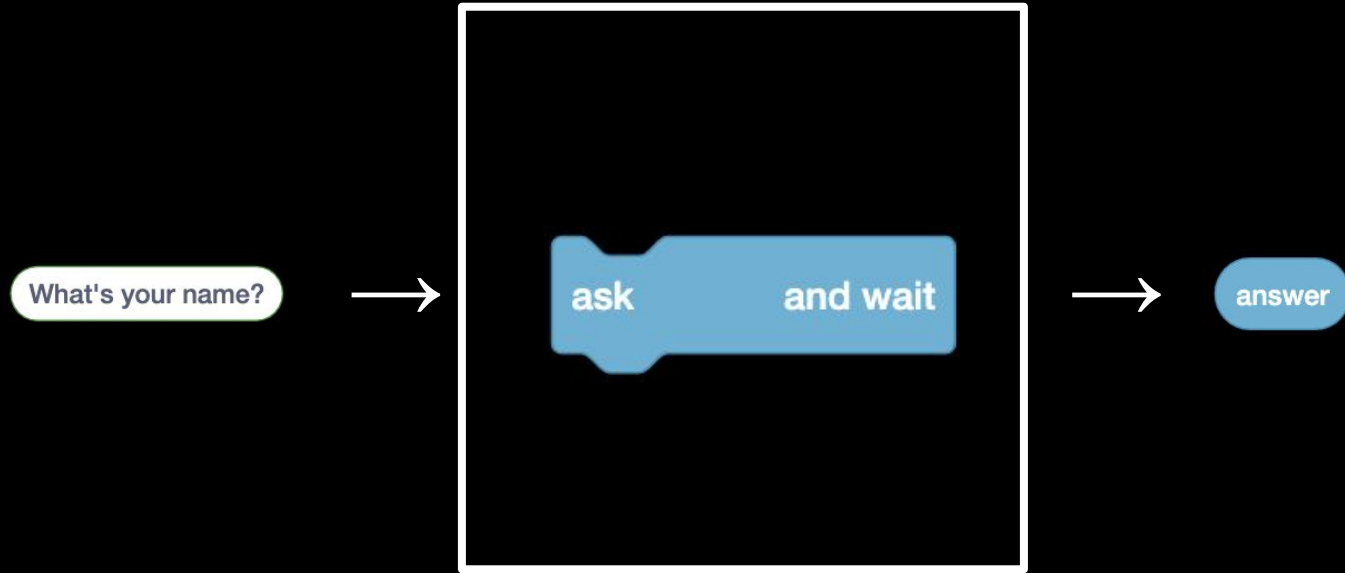


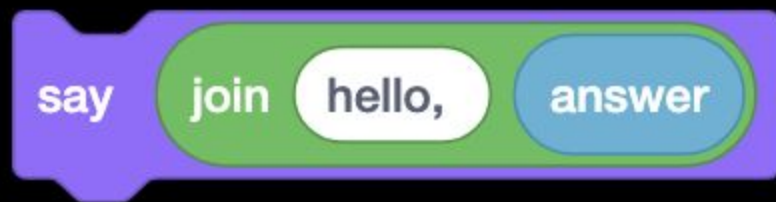
output

What's your name?



→ output



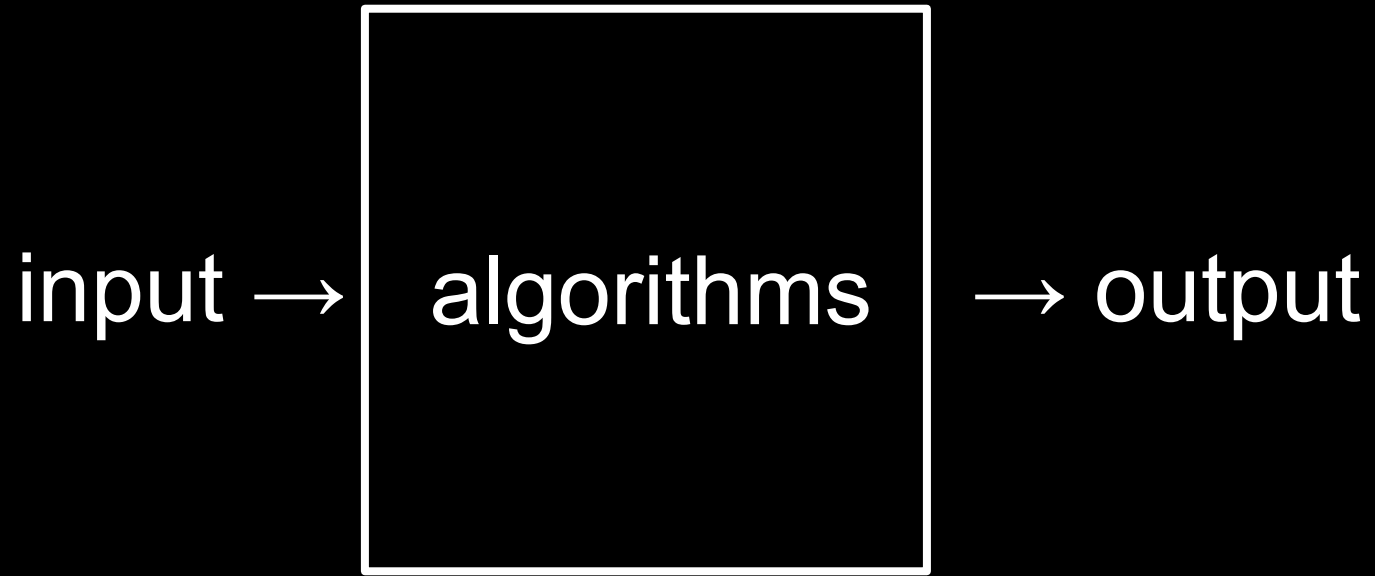


say

join

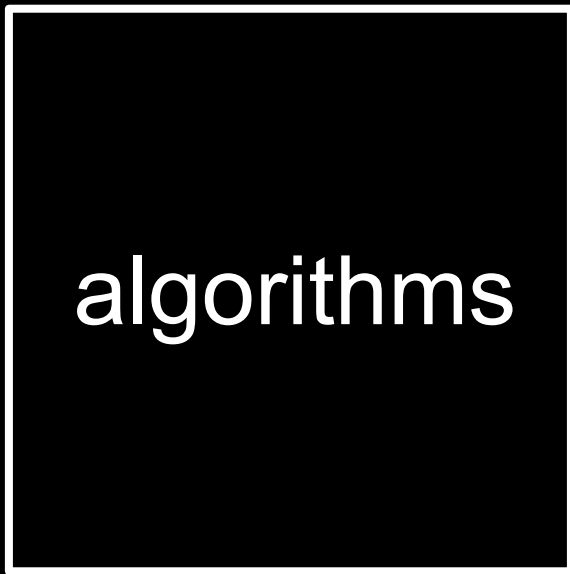
hello,

answer

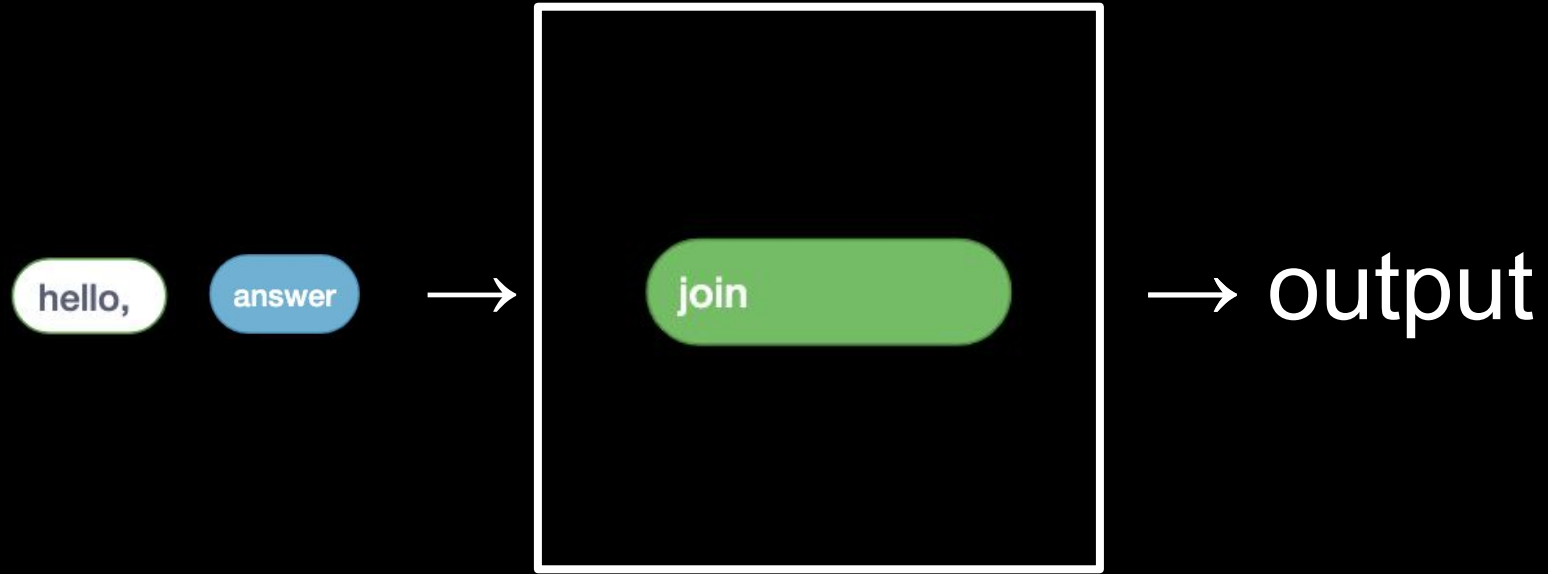


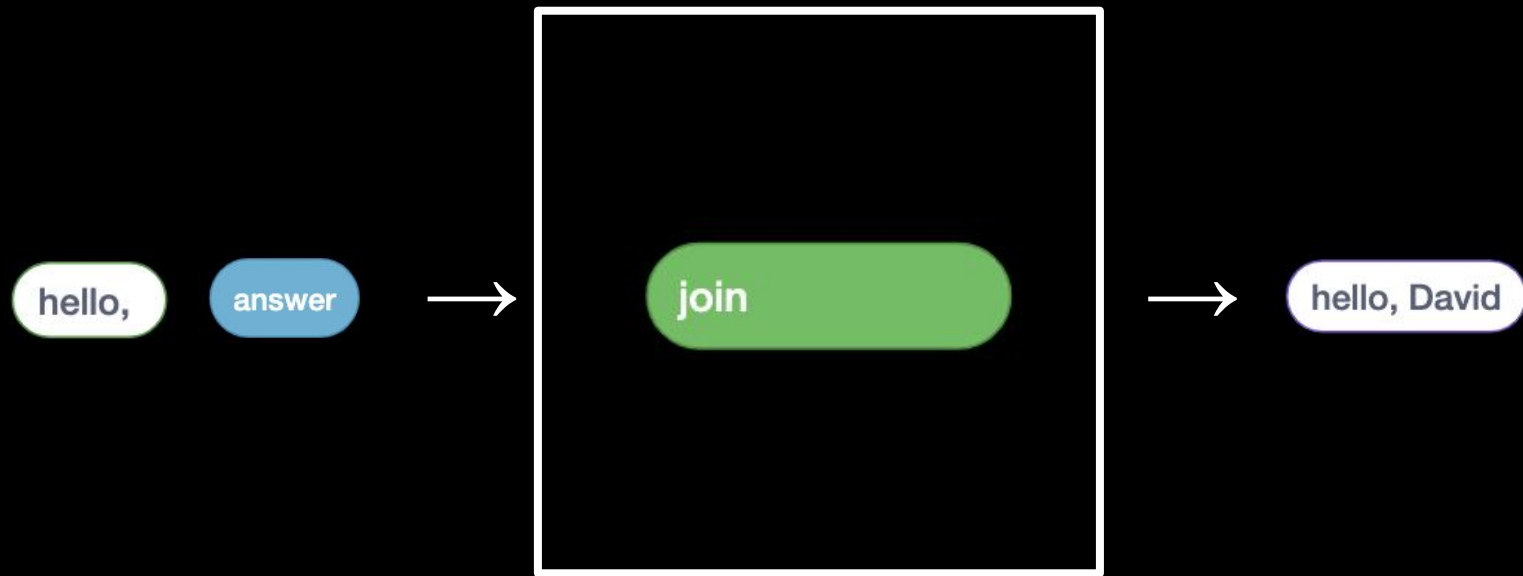
hello,

answer



output







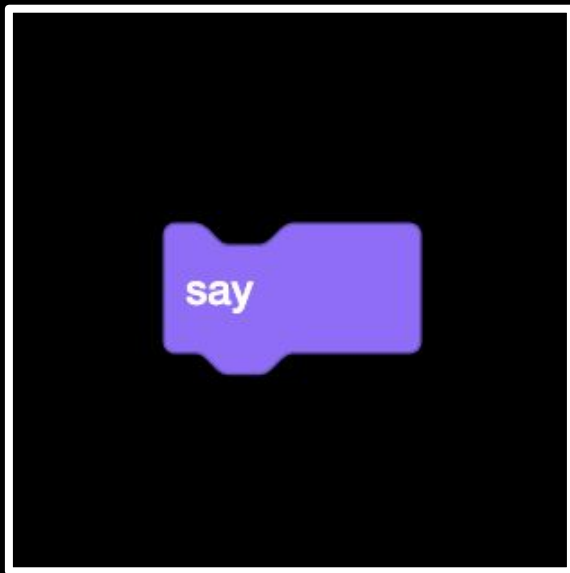
hello, David



hello, David



hello, David





hello, David



say



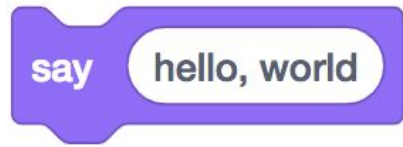
hello, David

```
make hello
```

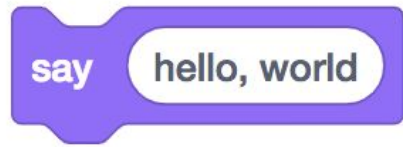
```
./hello
```

functions, arguments





```
print ( )
```



```
printf(      )
```




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



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

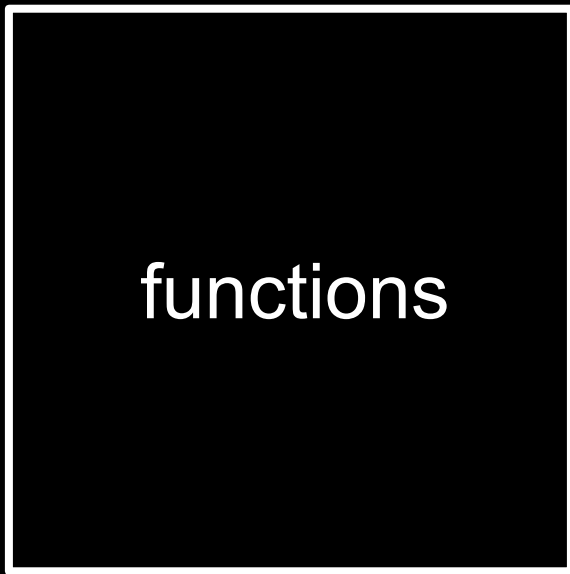


```
printf("hello, world");
```



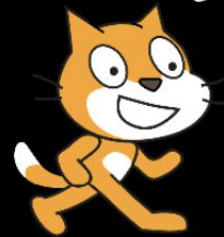
functions

arguments →



functions

side effects



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? ")
```

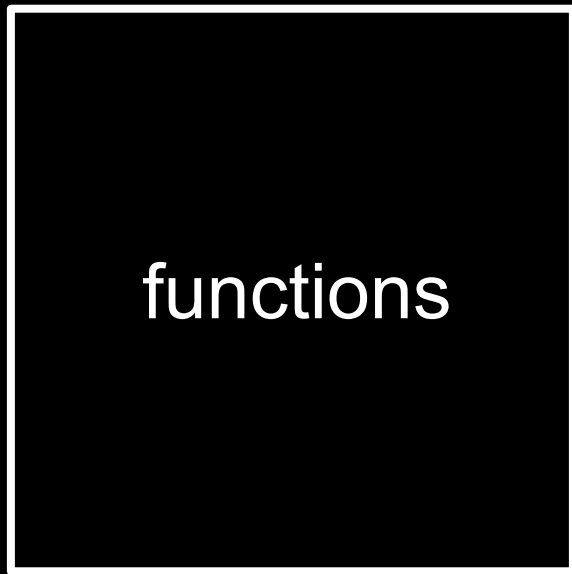


```
string answer = get_string("What's your name? ");
```



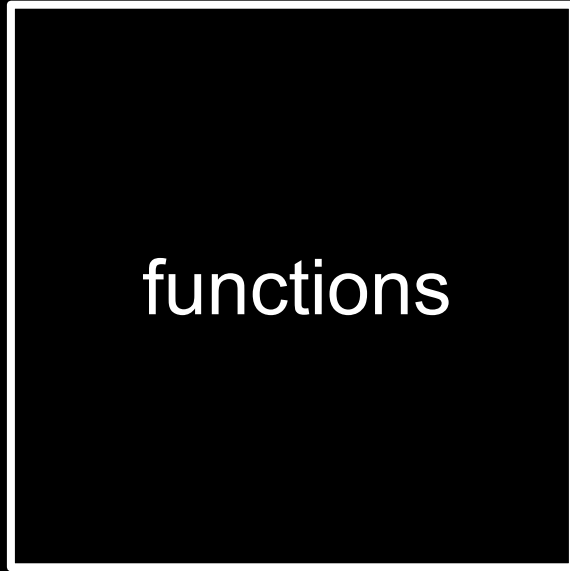
functions

arguments →



functions

arguments →



functions

→ return value





```
printf( );
```



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



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

main





```
int main(void)
{
}
}
```

header files



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



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

```
int main(void)
```

```
{
```

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

```
}
```

help50

style50

check50

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





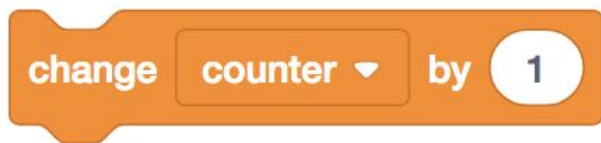
```
counter = 0
```

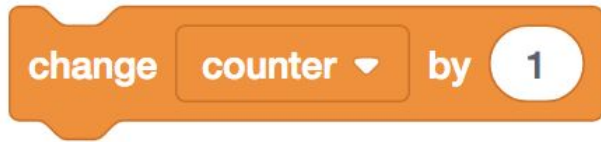


```
int counter = 0
```

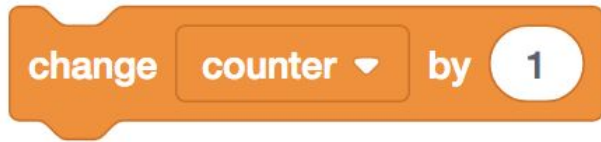


```
int counter = 0;
```

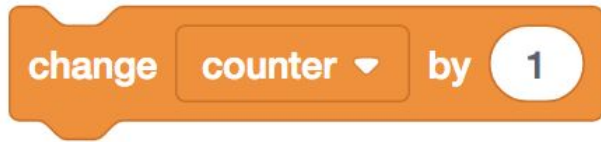




```
counter = counter + 1
```



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



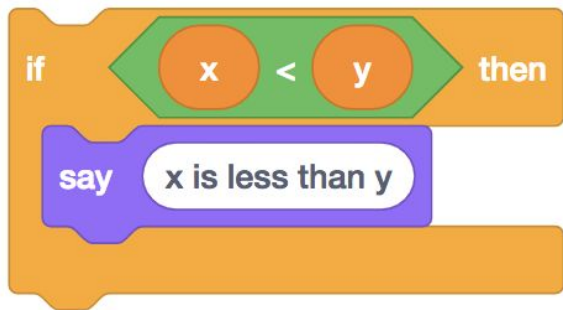
```
counter += 1;
```




```
counter++;
```

conditions





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

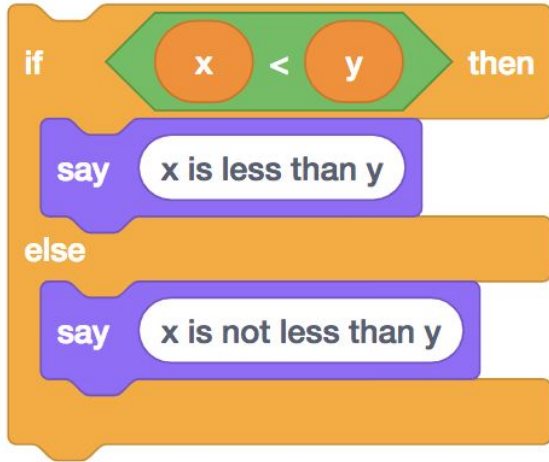


```
if (x < y)
{
    printf("x is less than y\n");
}
```

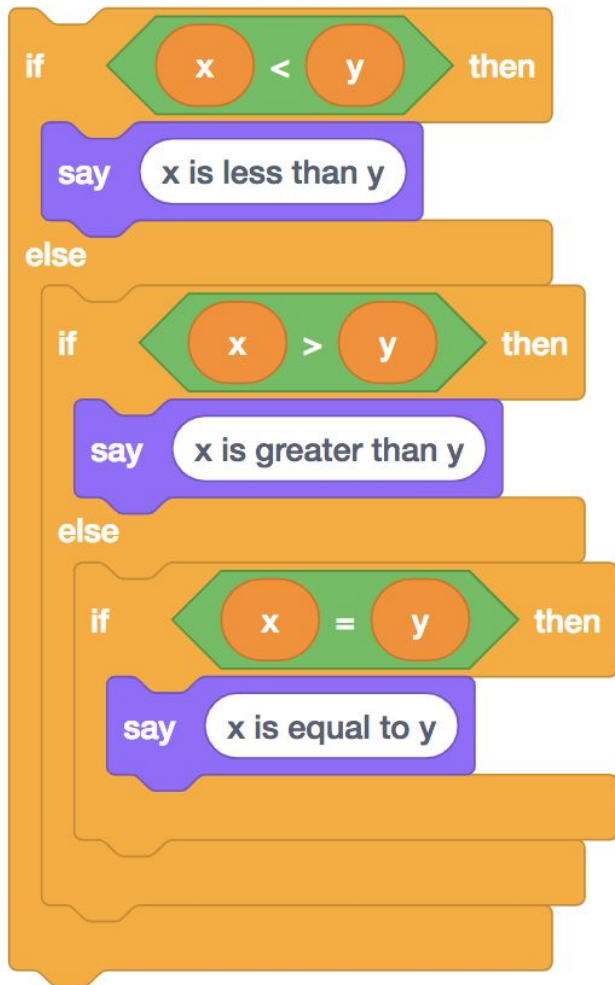


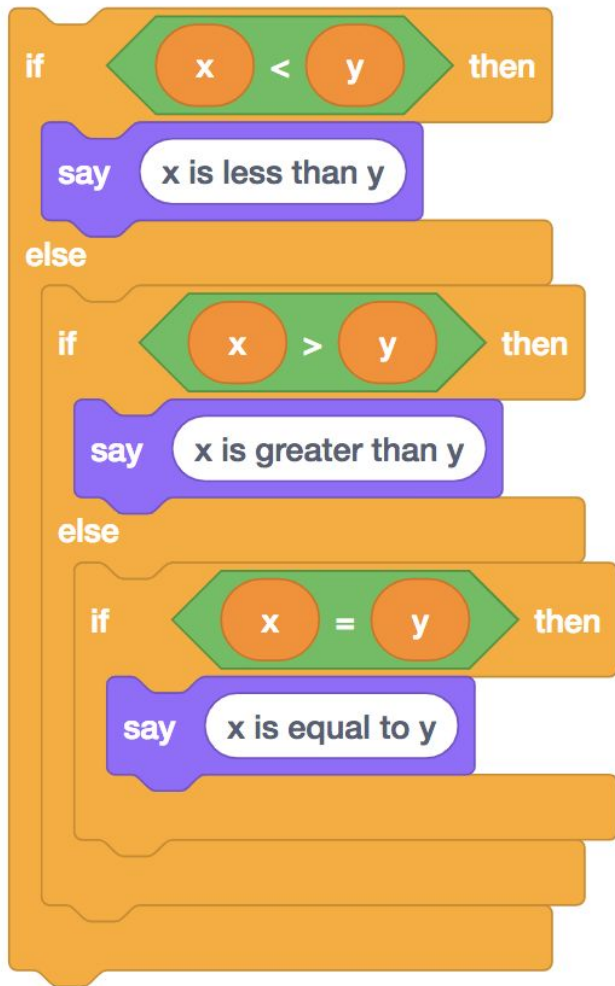


```
if (x < y)
{
}
else
{
}
```

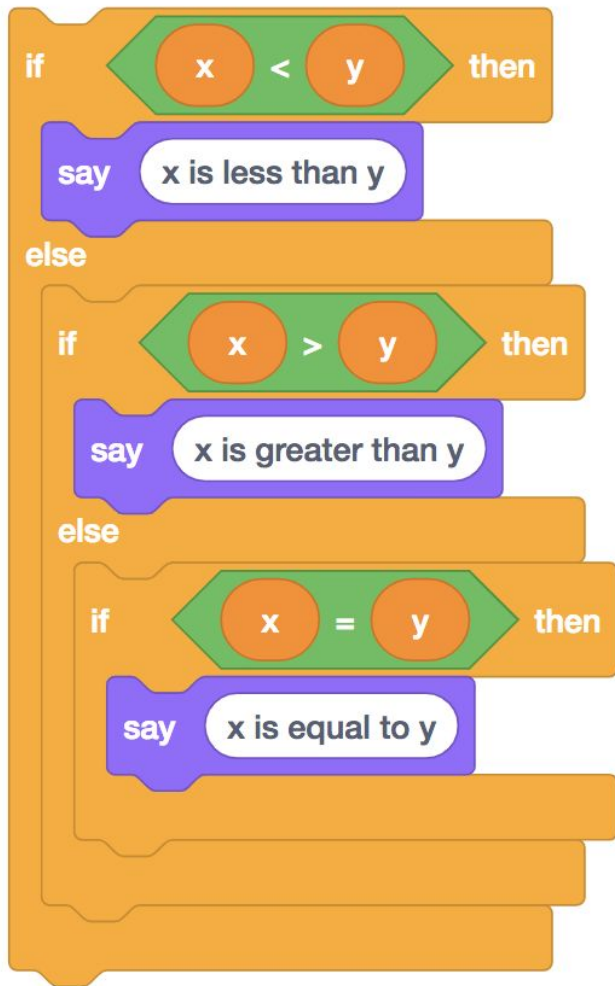


```
if (x < y)
{
    printf("x is less than y\n");
}
else
{
    printf("x is not less than y\n");
}
```

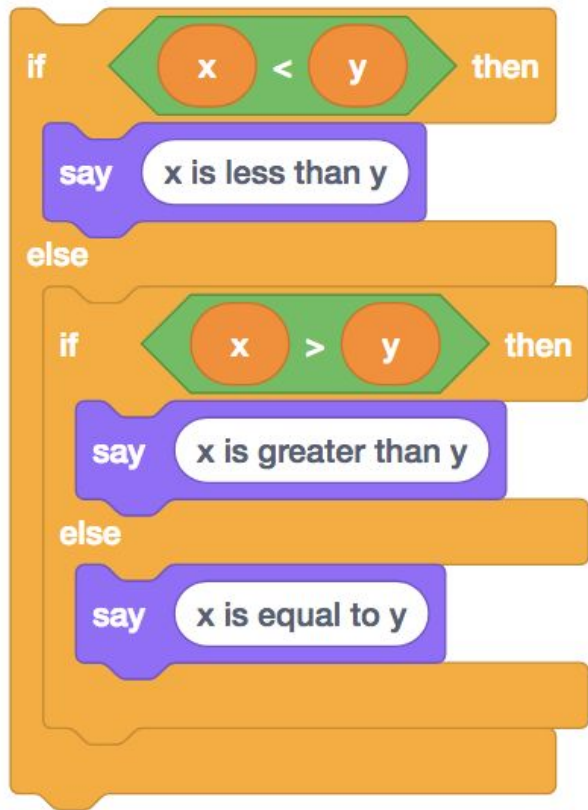





```
if (x < y)
{
}
else if (x > y)
{
}
else if (x == y)
{
}
```

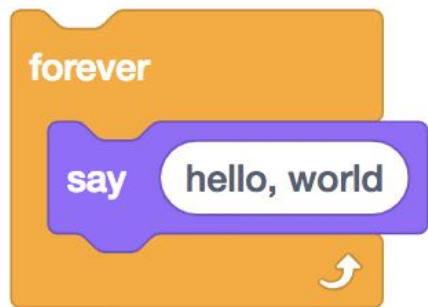


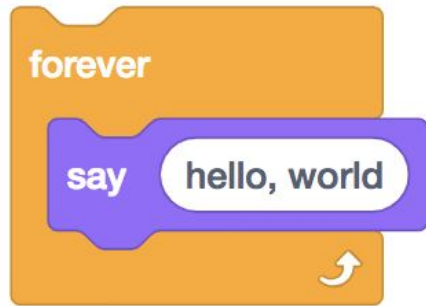
```
if (x < y)
{
    printf("x is less than y\n");
}
else if (x > y)
{
    printf("x is greater than y\n");
}
else if (x == y)
{
    printf("x is equal to y\n");
}
```



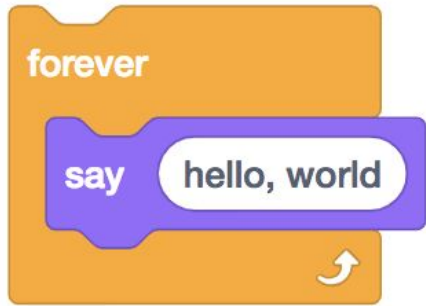
```
if (x < y)
{
    printf("x is less than y\n");
}
else if (x > y)
{
    printf("x is greater than y\n");
}
else
{
    printf("x is equal to y\n");
}
```

loops

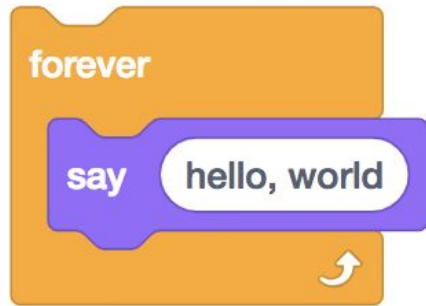




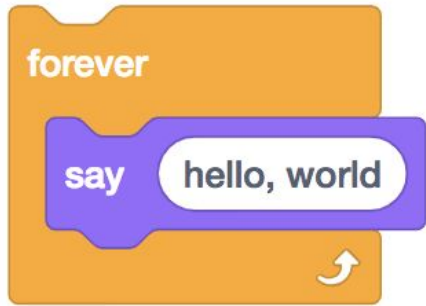
```
while  
{  
  
}
```



```
while
{
    printf("hello, world\n");
}
```

```
while ( )  
{  
    printf("hello, world\n");  
}
```



```
while (true)
{
    printf("hello, world\n");
}
```





```
int counter = 0;
```



```
int i = 0;
```



```
int i = 0;  
while (    )  
{  
  
}
```



```
int i = 0;  
while (i < 50)  
{  
  
}
```



```
int i = 0;  
while (i < 50)  
{  
    printf("hello, world\n");  
}
```




```
int i = 0;
while (i < 50)
{
    printf("hello, world\n");
    i = i + 1;
}
```



```
int i = 0;
while (i < 50)
{
    printf("hello, world\n");
    i += 1;
}
```



```
int i = 0;  
while (i < 50)  
{  
    printf("hello, world\n");  
    i++;  
}
```



```
int i = 1;
while (i <= 50)
{
    printf("hello, world\n");
    i++;
}
```



```
int i = 50;
while (i > 0)
{
    printf("hello, world\n");
    i--;
}
```





```
for  
{  
  
}  

```



```
for  
{  
    printf("hello, world\n");  
}
```




```
for (                                     )  
{  
    printf("hello, world\n");  
}
```



```
for (int counter = 0;           )  
{  
    printf("hello, world\n");  
}
```



```
for (int i = 0;           )  
{  
    printf("hello, world\n");  
}
```



```
for (int i = 0; i < 50;      )  
{  
    printf("hello, world\n");  
}
```



```
for (int i = 0; i < 50; i = i + 1)
{
    printf("hello, world\n");
}
```



```
for (int i = 0; i < 50; i += 1)
{
    printf("hello, world\n");
}
```



```
for (int i = 0; i < 50; i++)  
{  
    printf("hello, world\n");  
}
```