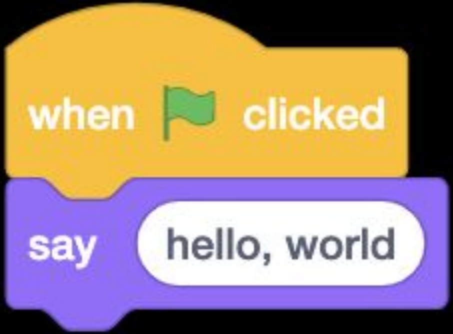
This is CS50



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

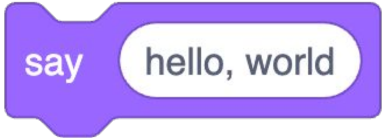
```c
#include <stdio.h>

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

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

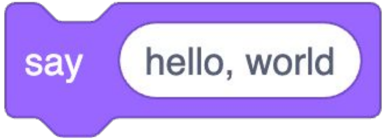
```
make hello

./hello
```

```
clang -o hello hello.c -lcs50

./hello
```

```
python hello.py
```

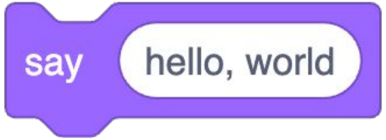


```
say (hello, world)
```



say hello, world

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



```
say hello, world
```

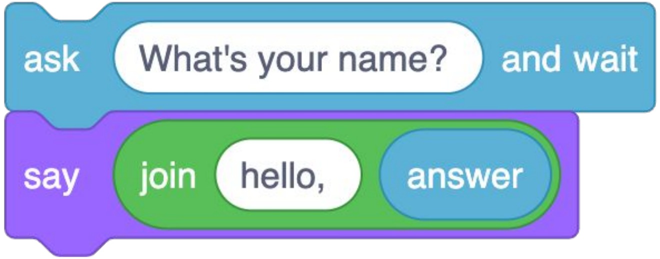
```
print("hello, world")
```

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

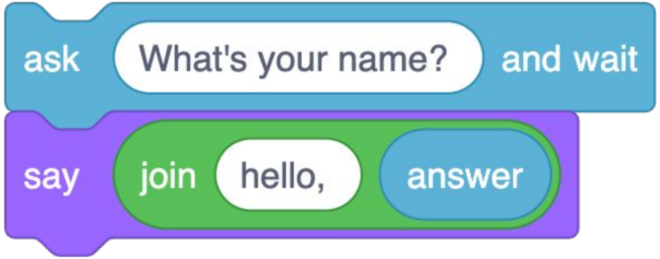
```
import cs50
```

```
from cs50 import get_string
```

functions

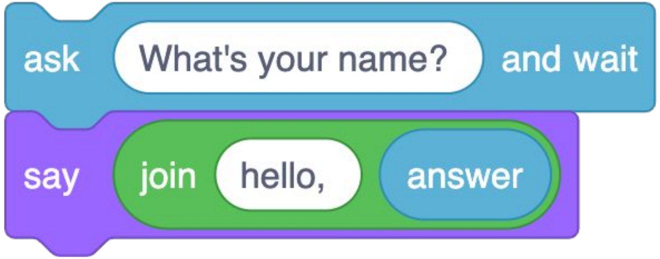


```
ask (What's your name?) and wait
say (join (hello,) (answer))
```



ask What's your name? and wait
say join hello, answer

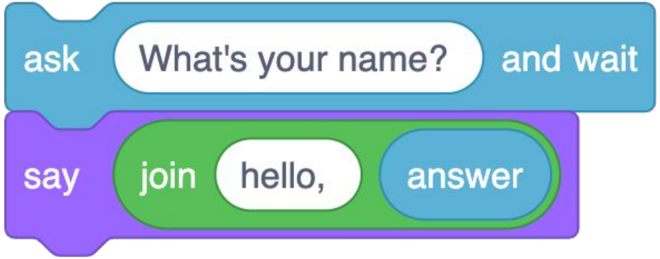
```
string answer = get_string("What's your name? ");
printf("hello, %s\n", answer);
```



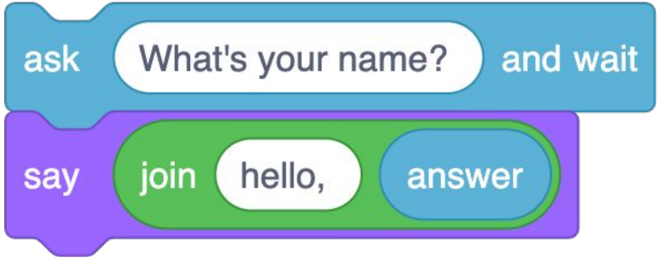
ask What's your name? and wait

say join hello, answer

```
answer = get_string("What's your name? ")
print("hello, " + answer)
```


ask What's your name? and wait
say join hello, answer

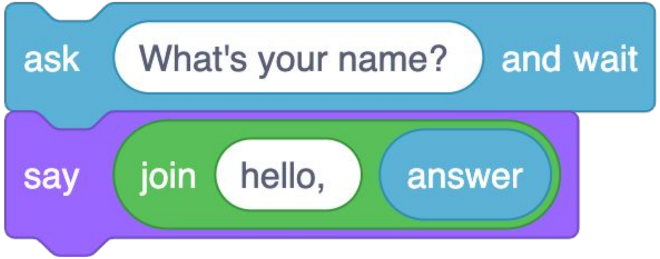
```
answer = get_string("What's your name? ")
print("hello,", answer)
```



ask What's your name? and wait

say join hello, answer

```
answer = get_string("What's your name? ")
print(f"hello, {answer}")
```



```
answer = input("What's your name? ")
print(f"hello, {answer}")
```

types

```
bool

char

double

float

int

long

string

...
```

```
bool



float

int


str

...
```

```
range

list

tuple

dict

set

...
```

`range`	sequence of numbers
`list`	sequence of mutable values
`tuple`	sequence of immutable values
`dict`	collection of key-value pairs
`set`	collection of unique values
`...`	

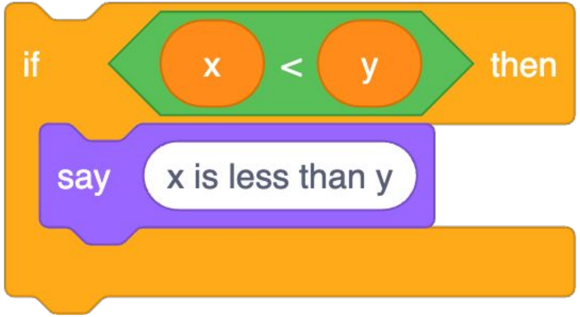
```
get_char

get_double

get_float

get_int

get_long

get_string

...
```

```
get_float

get_int



get_string
```

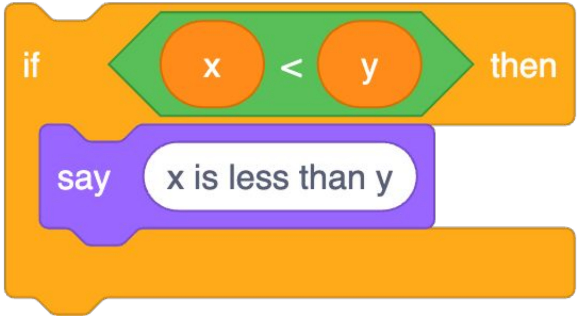
```
import cs50
```

```
from cs50 import get_float
from cs50 import get_int
from cs50 import get_string
from cs50 import get_float, get_int, get_string
```

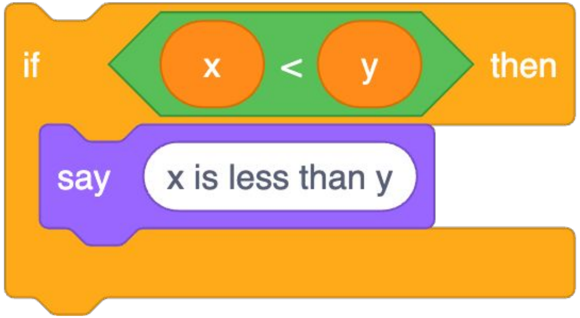
conditionals



```
if <(x) < (y)> then
    say [x is less than y]
```

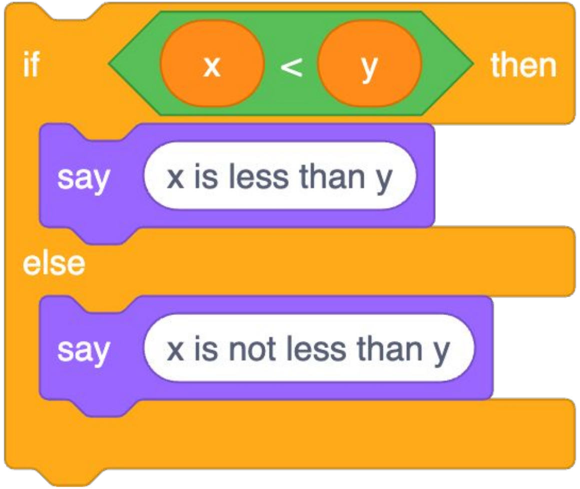


```
if (x < y)
{
    printf("x is less than y\n");
}
```

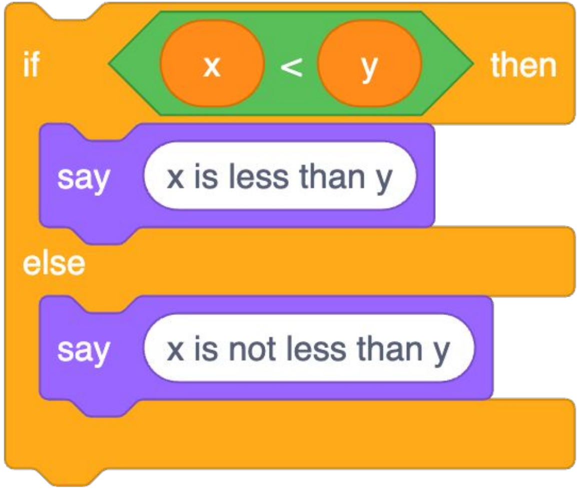


```
if x < y then
    say x is less than y
```

```
if x < y:
    print("x is less than y")
```

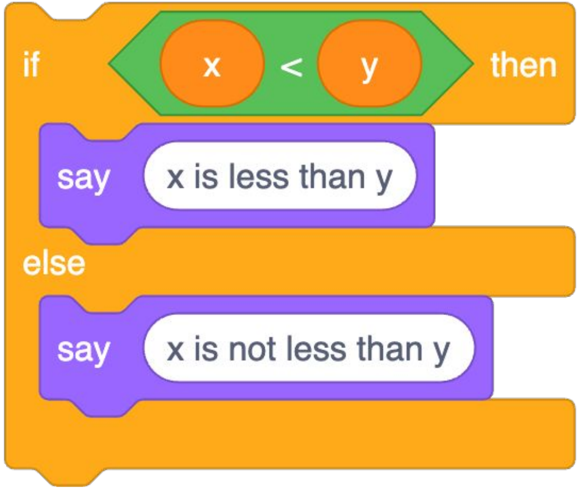


```
if <(x) < (y)> then
    say [x is less than y]
else
    say [x is not less than y]
```



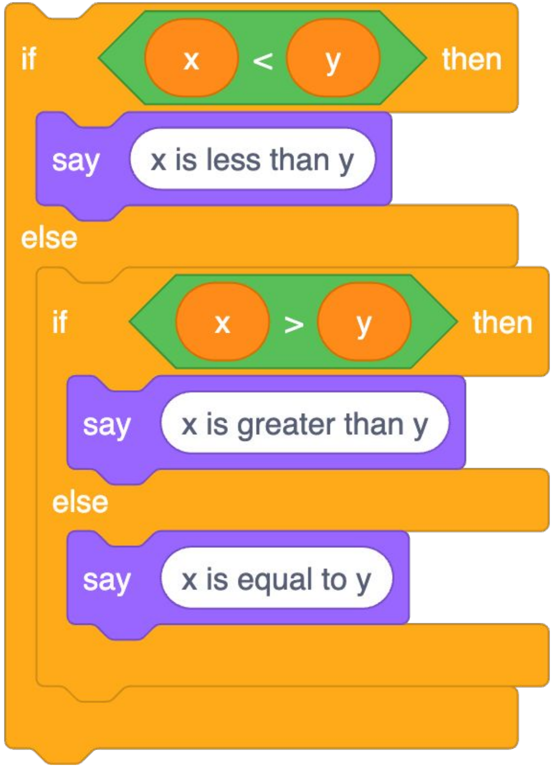
```
if x < y then
    say x is less than y
else
    say x is not less than y
```

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

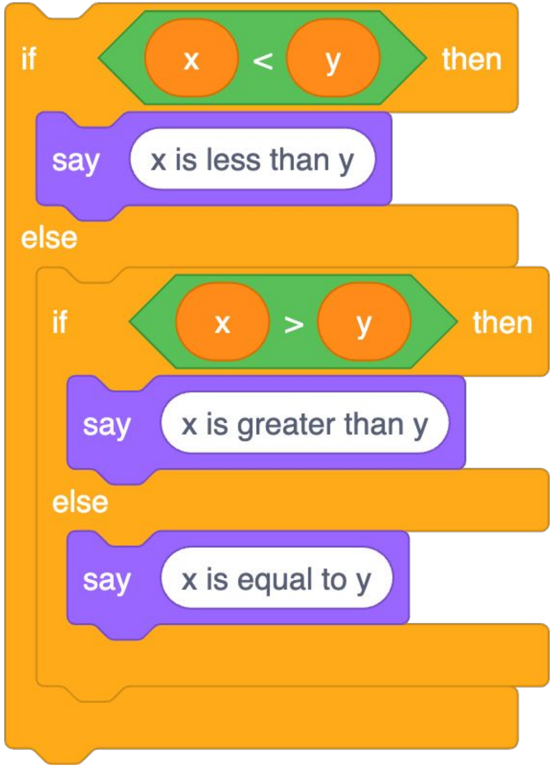


```
if x < y then
    say x is less than y
else
    say x is not less than y
```

```
if x < y:
    print("x is less than y")
else:
    print("x is not less than y")
```

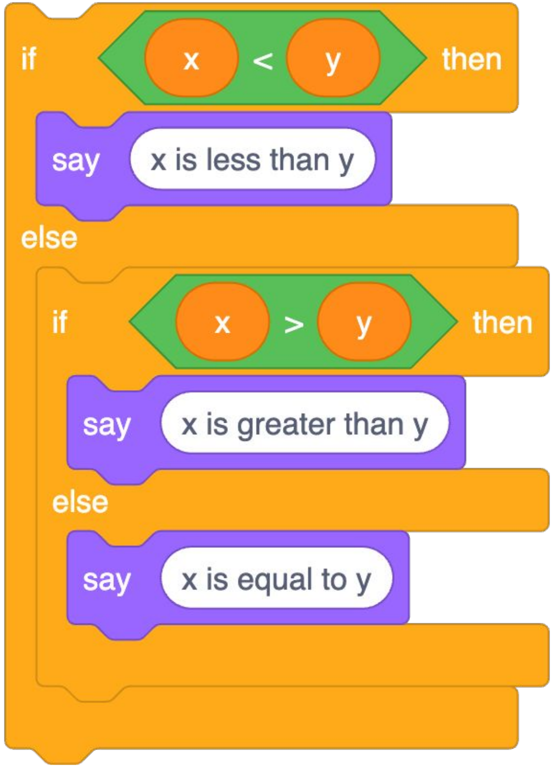


```
if x < y then
    say x is less than y
else
    if x > y then
        say x is greater than y
    else
        say x is equal to y
```



```
if x < y 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
```

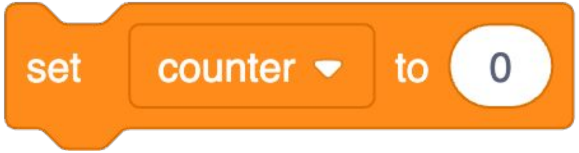
```
if (x < y)
{
    printf("x is less than y\n");
}
else if (x > y)
{
    printf("x is greater than y\n");
}
else
{
    printf("x is equal to y\n");
}
```



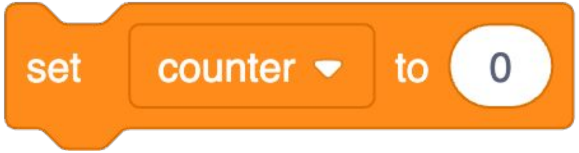
```
if x < y then
    say x is less than y
else
    if x > y then
        say x is greater than y
    else
        say x is equal to y
```

```
if x < y:
    print("x is less than y")
elif x > y:
    print("x is greater than y")
else:
    print("x is equal to y")
```

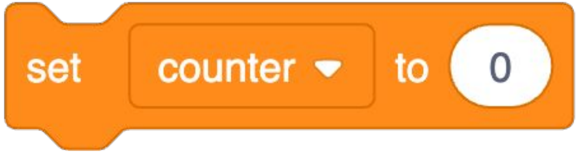
variables



```
set counter ▾ to 0
```

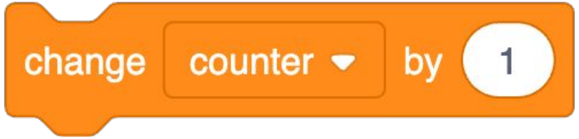


```
int counter = 0;
```

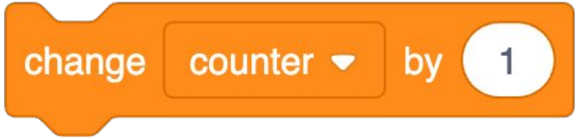


set counter ▾ to 0

```
counter = 0
```

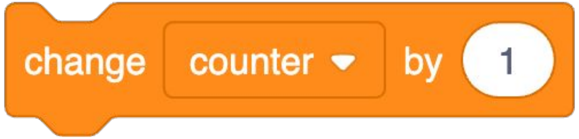


```
change counter ▾ by 1
```



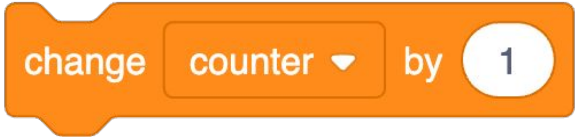
change counter ▾ by 1

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



change counter ▾ by 1

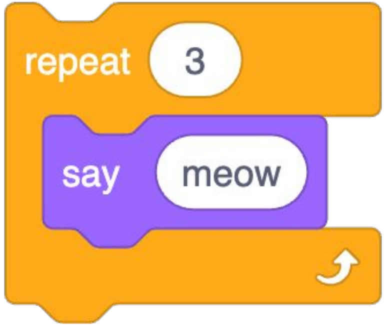
```
counter = counter + 1
```



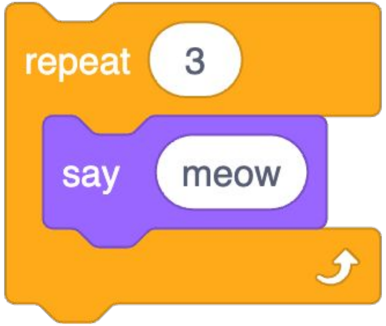
change counter by 1

```
counter += 1
```

loops

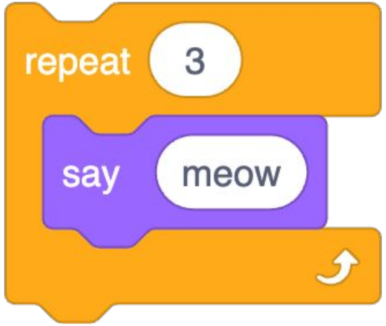


```
repeat 3
    say meow
```

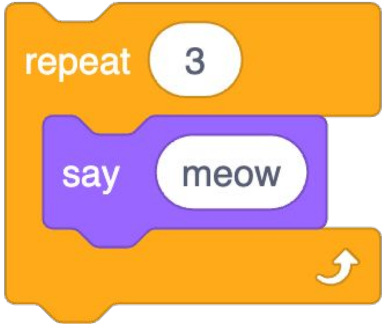


```
repeat 3
    say meow
```

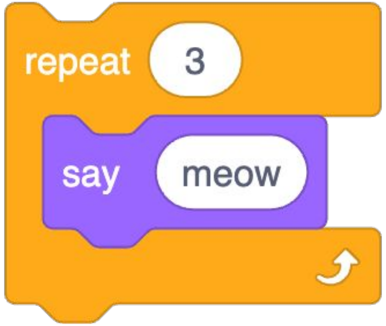
```
int i = 0;
while (i < 3)
{
    printf("meow\n");
    i++;
}
```



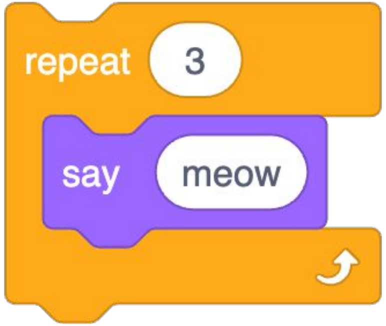
```
i = 0
while i < 3:
    print("meow")
    i += 1
```



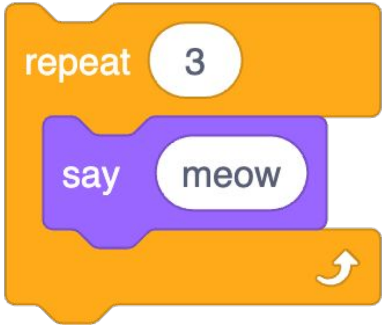
```
repeat 3
    say meow
```



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

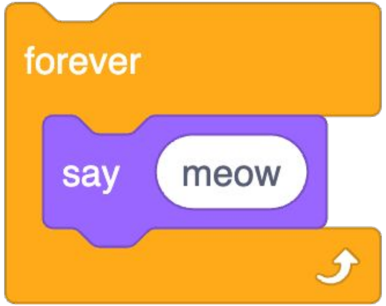


```
for i in [0, 1, 2]:
    print("hello, world")
```

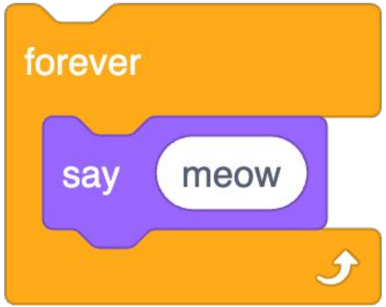


```
repeat 3
    say meow
```

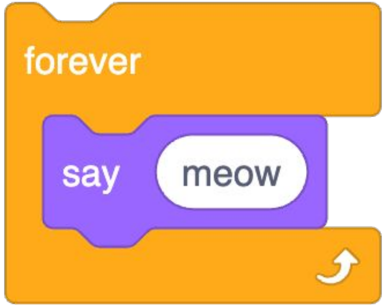
```
for i in range(3):
    print("hello, world")
```



```
forever
    say meow
```



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



```
while True:
    print("meow")
```

truncation

floating-point imprecision

integer overflow

~~integer overflow~~

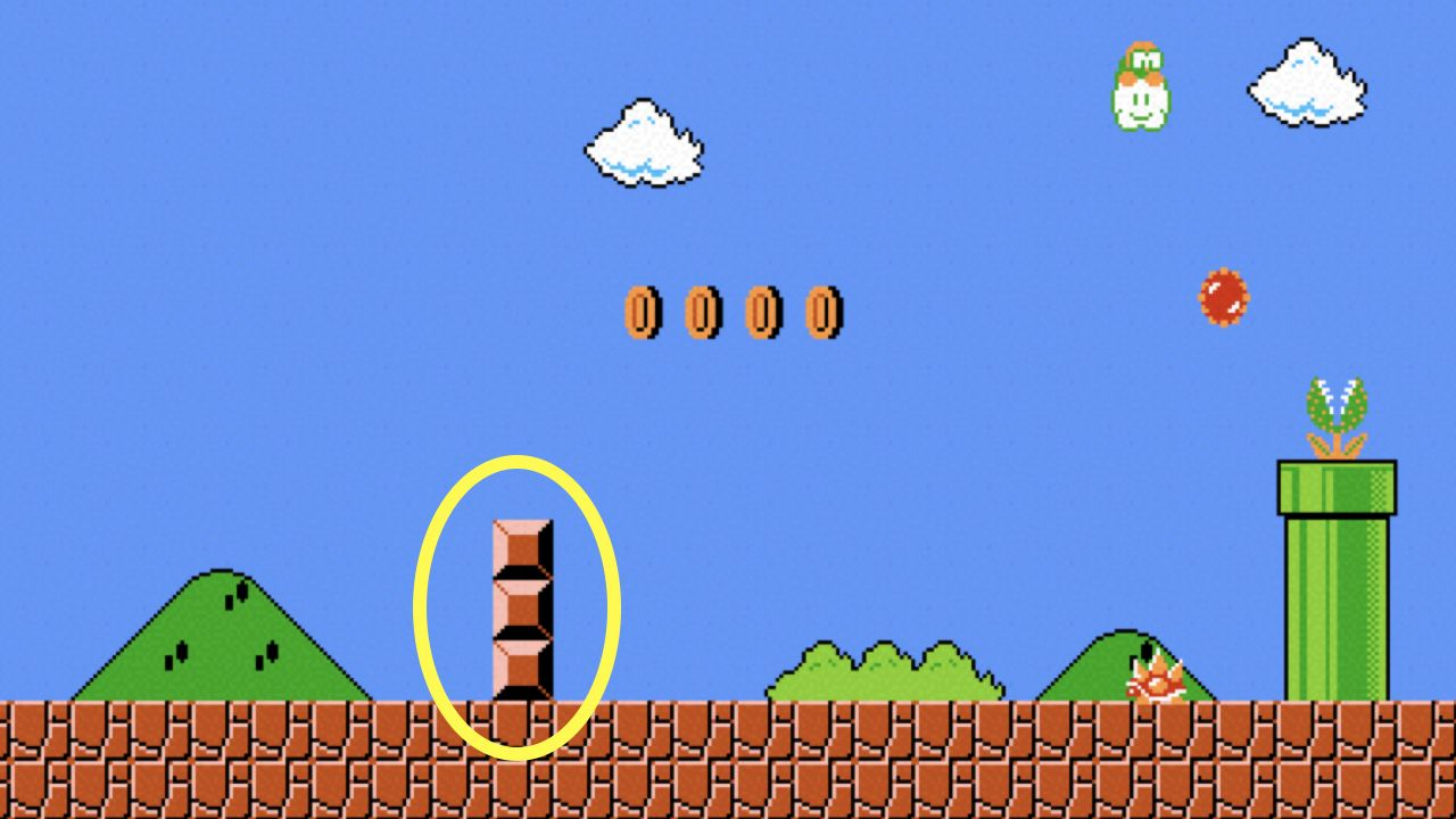
docs.python.org

docs.python.org/3/library/functions.html object-oriented programming

OOP

str

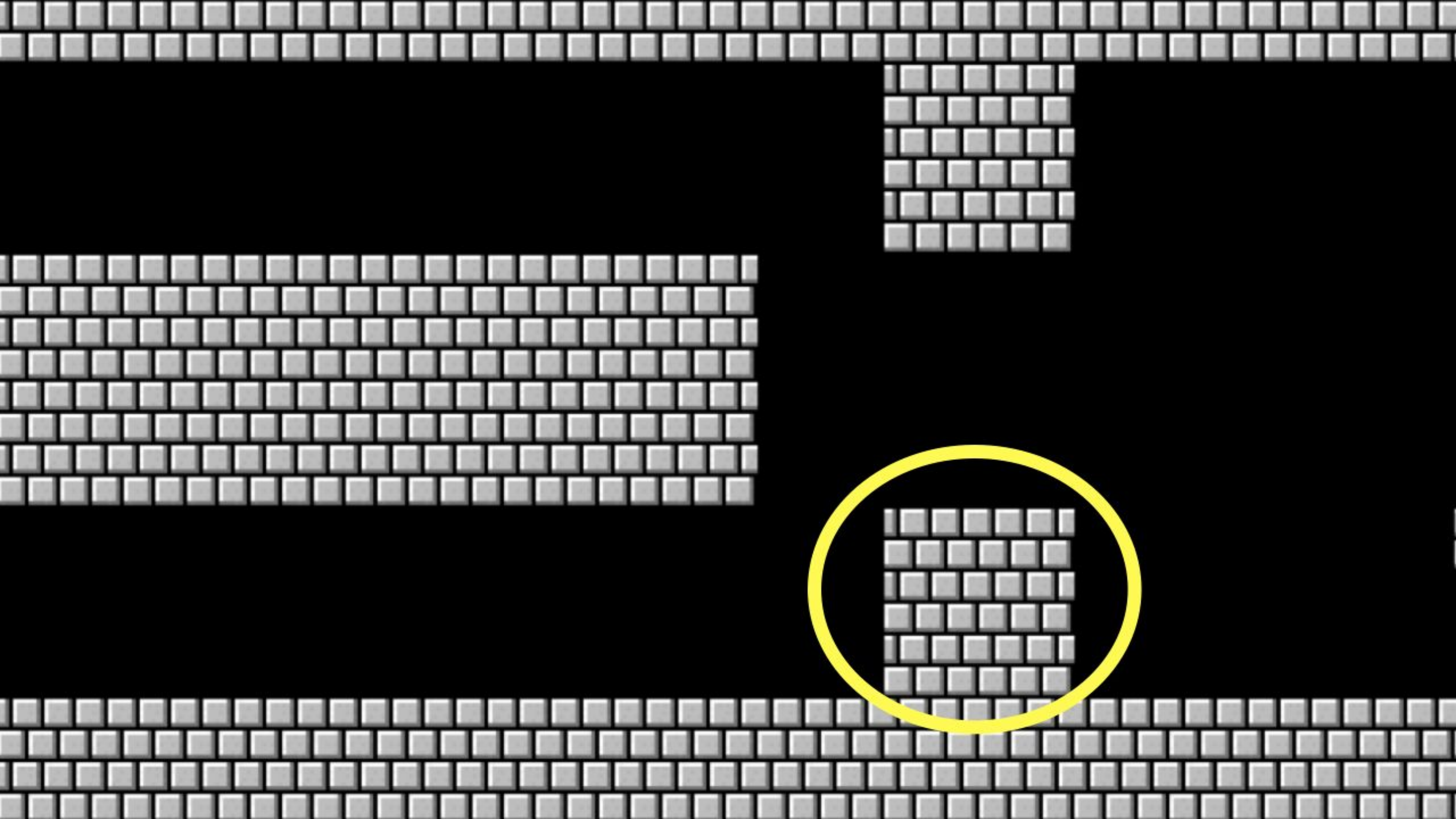
docs.python.org/3/library/stdtypes.html#string-methods



exceptions

```
print
```

docs.python.org/3/library/functions.html#print



`list`

docs.python.org/3/library/stdtypes.html#sequence-types-list-tuple-range len

docs.python.org/3/library/functions.html#len sys

docs.python.org/3/library/sys.html

CSV

docs.python.org/3/library/csv.html

key	value

This is CS50