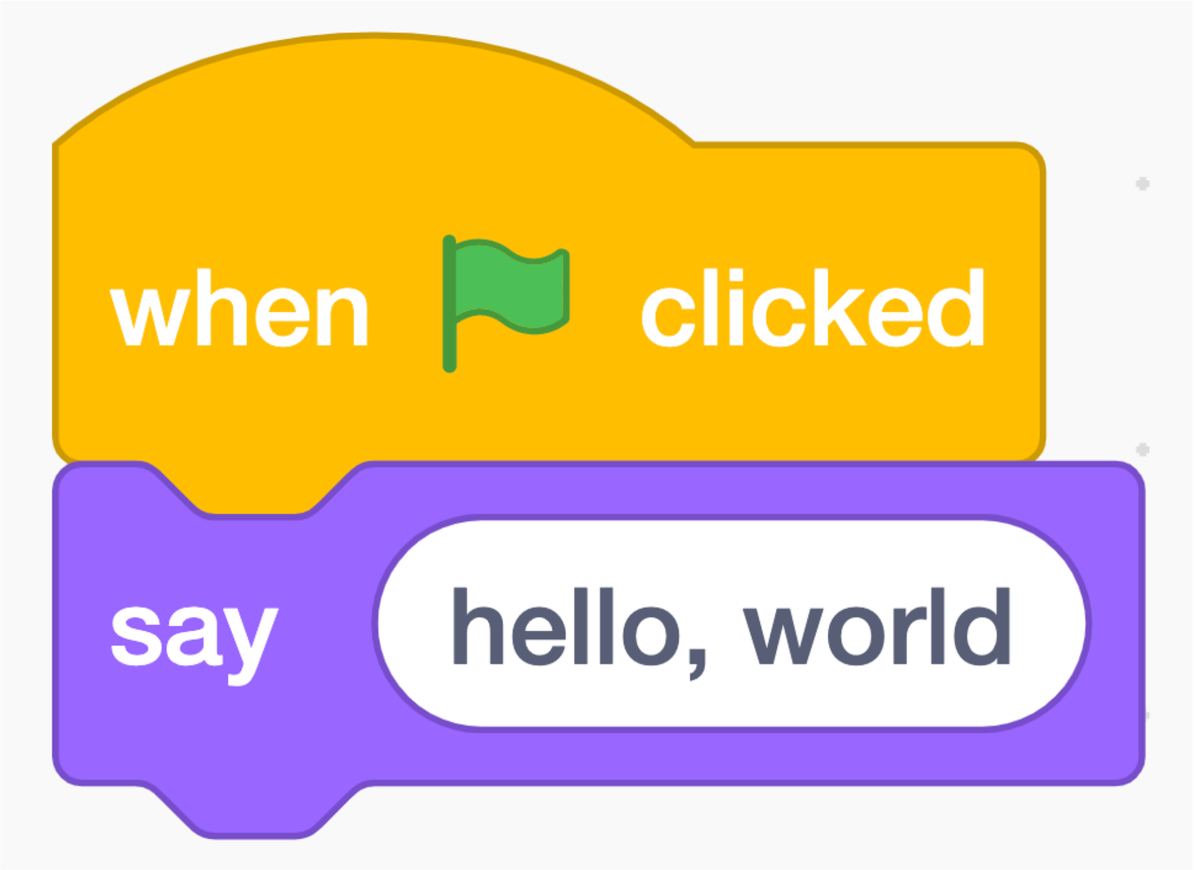
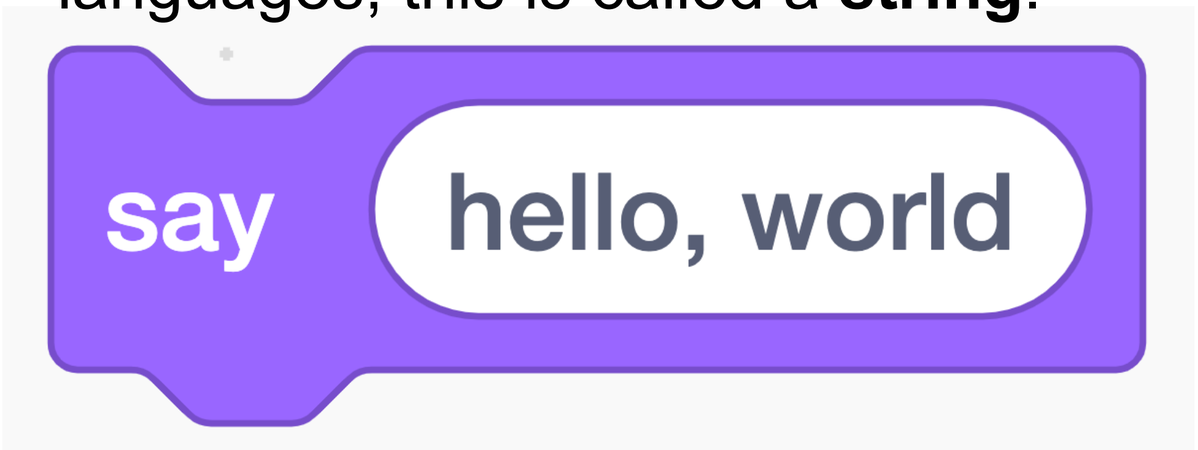
Python vs. Scratch




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

Say Block vs. print

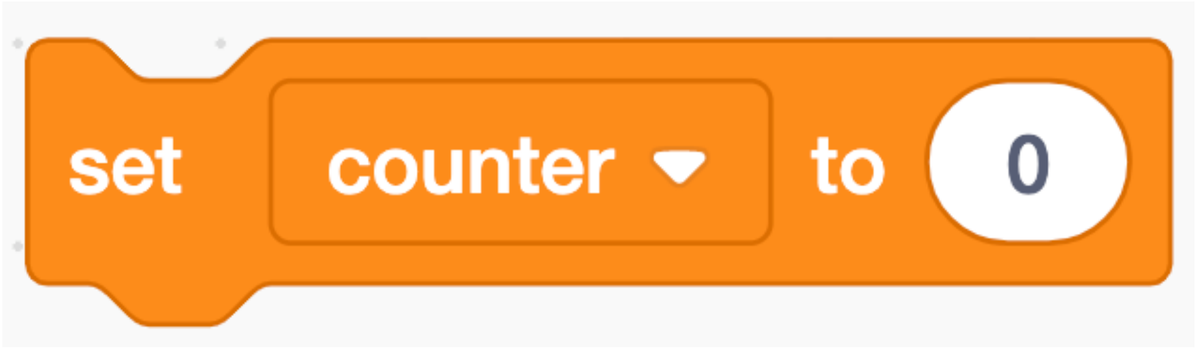
- Use double quotes to surround your text.
- In Python and some other general use languages, this is called a **string**.




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

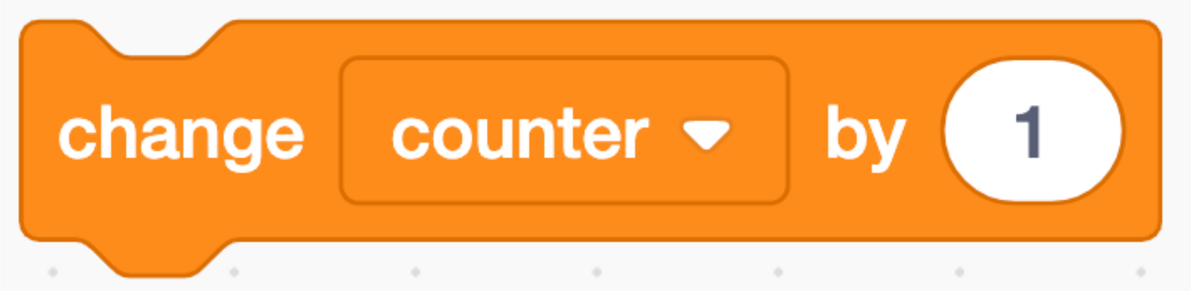
Variable Declaration

- The “set [counter] to (0)” block creates a variable and assigns 0 to it.
- = is not like an equation. It means “copy the value on the right, into the value on the left.”



counter = 0

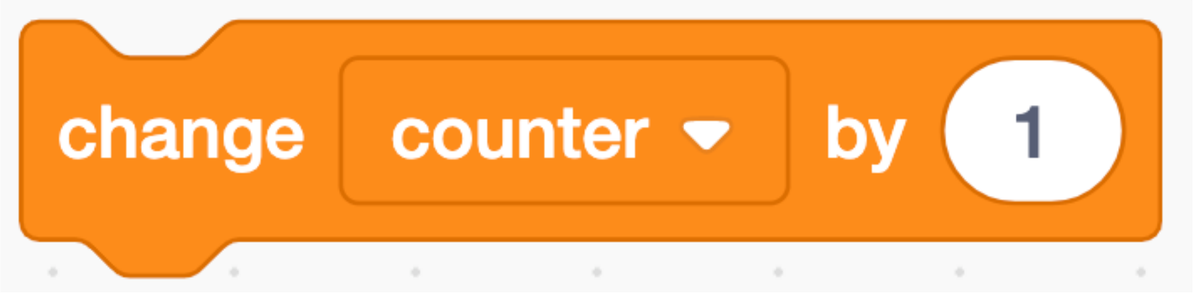
Change Values in Variables



counter = counter + 1

Change Values in Variables

- This has the same meaning as the last one:
  - **counter += 1** is the same as **counter = counter + 1**
  - This is a shortcut.



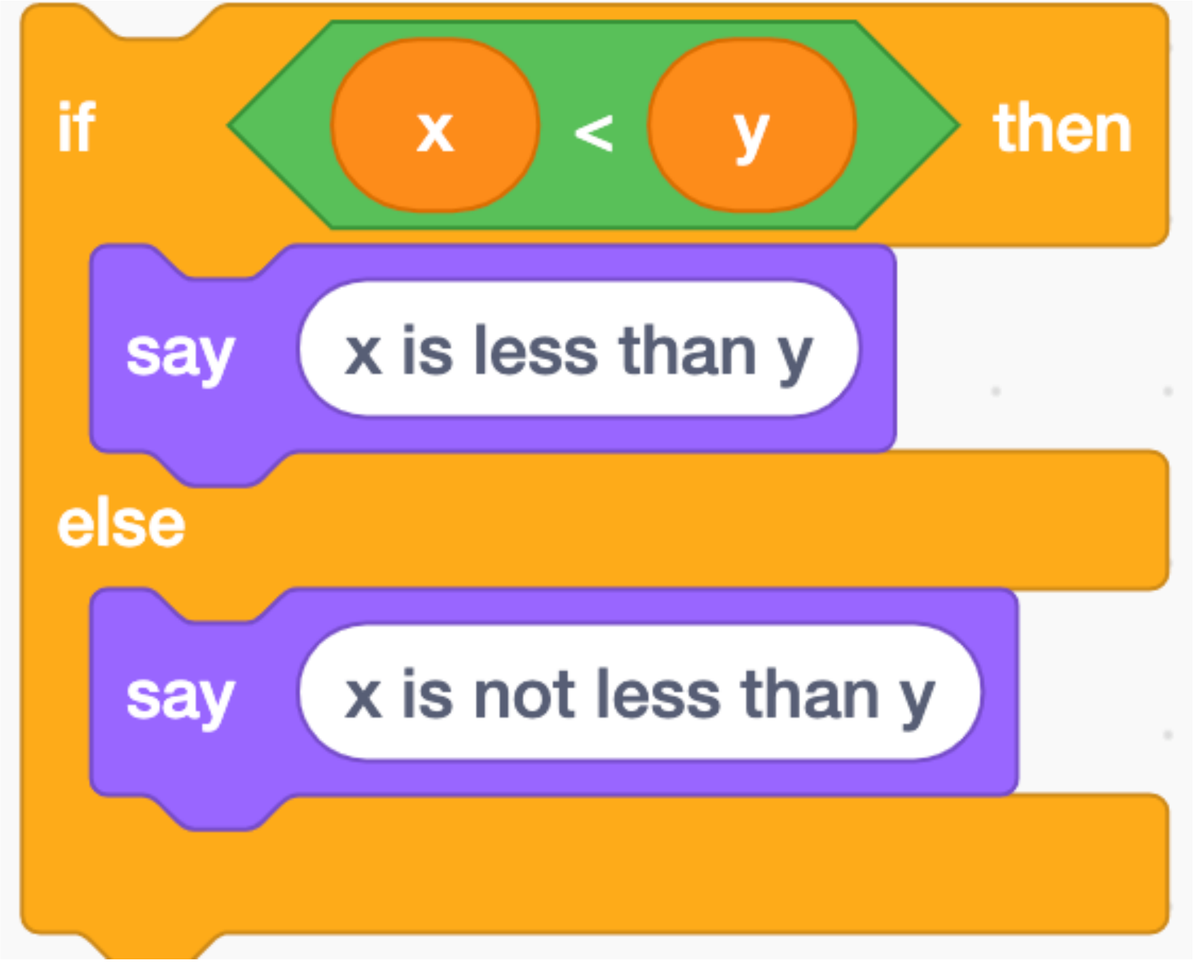
counter += 1

Conditions

- Notice that in Python, we use **:** (as well as indentation) to indicate how lines of code should be nested.


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

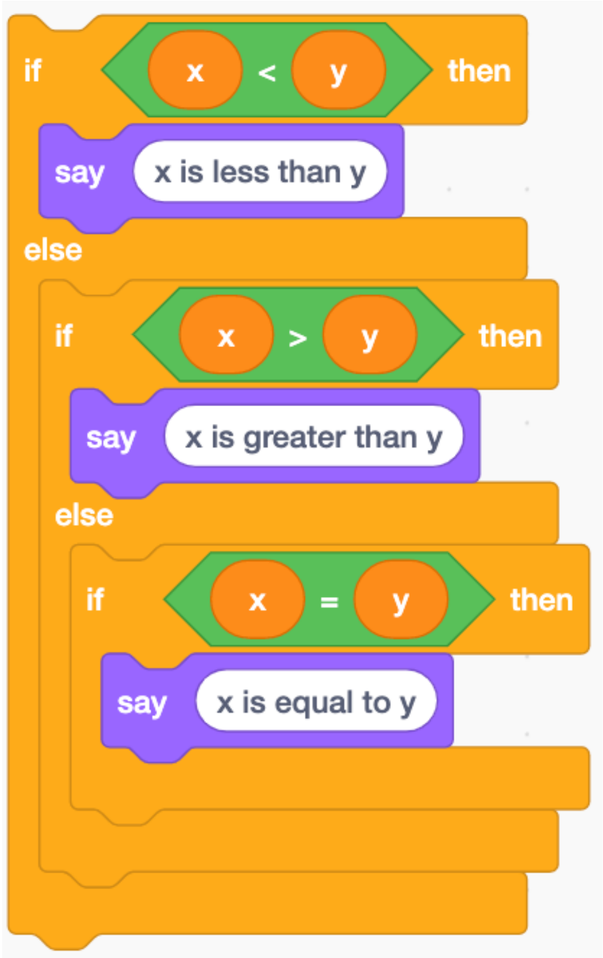
Conditions




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

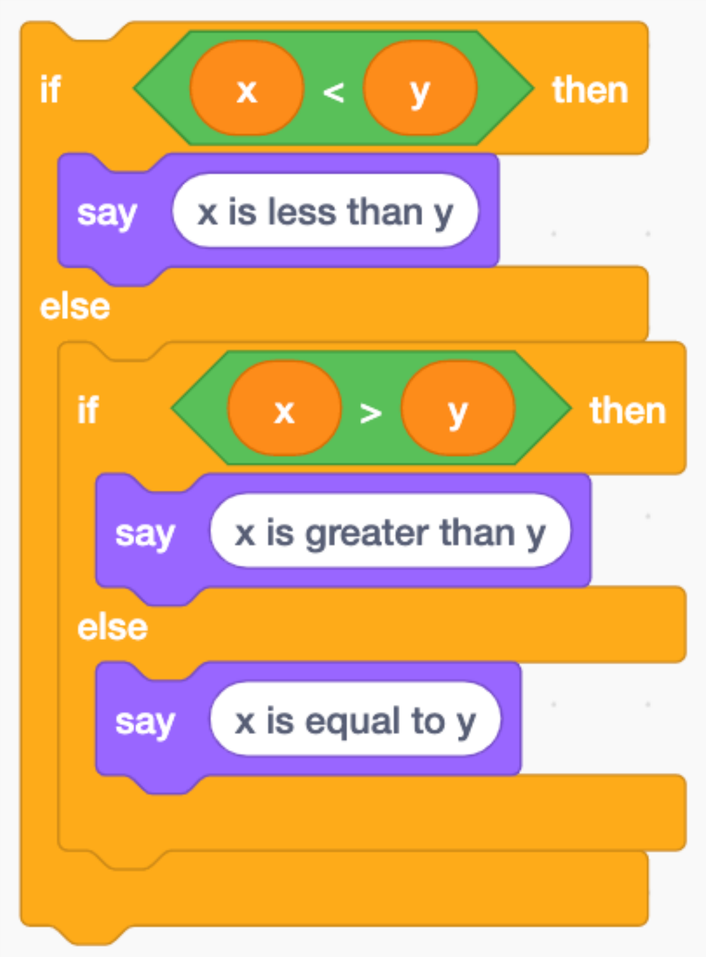
Conditions

- Use **==** to compare two values in Python.




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

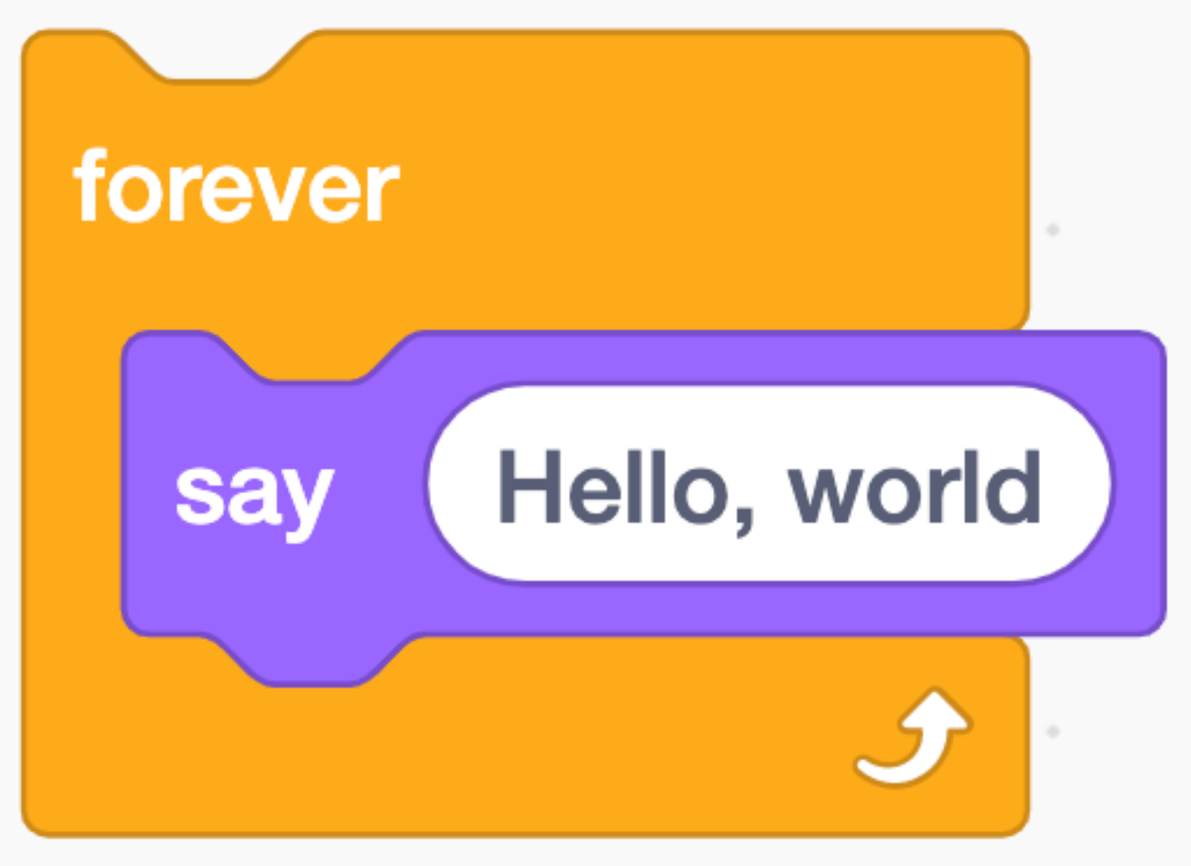
Conditions




```
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")
```

Loops

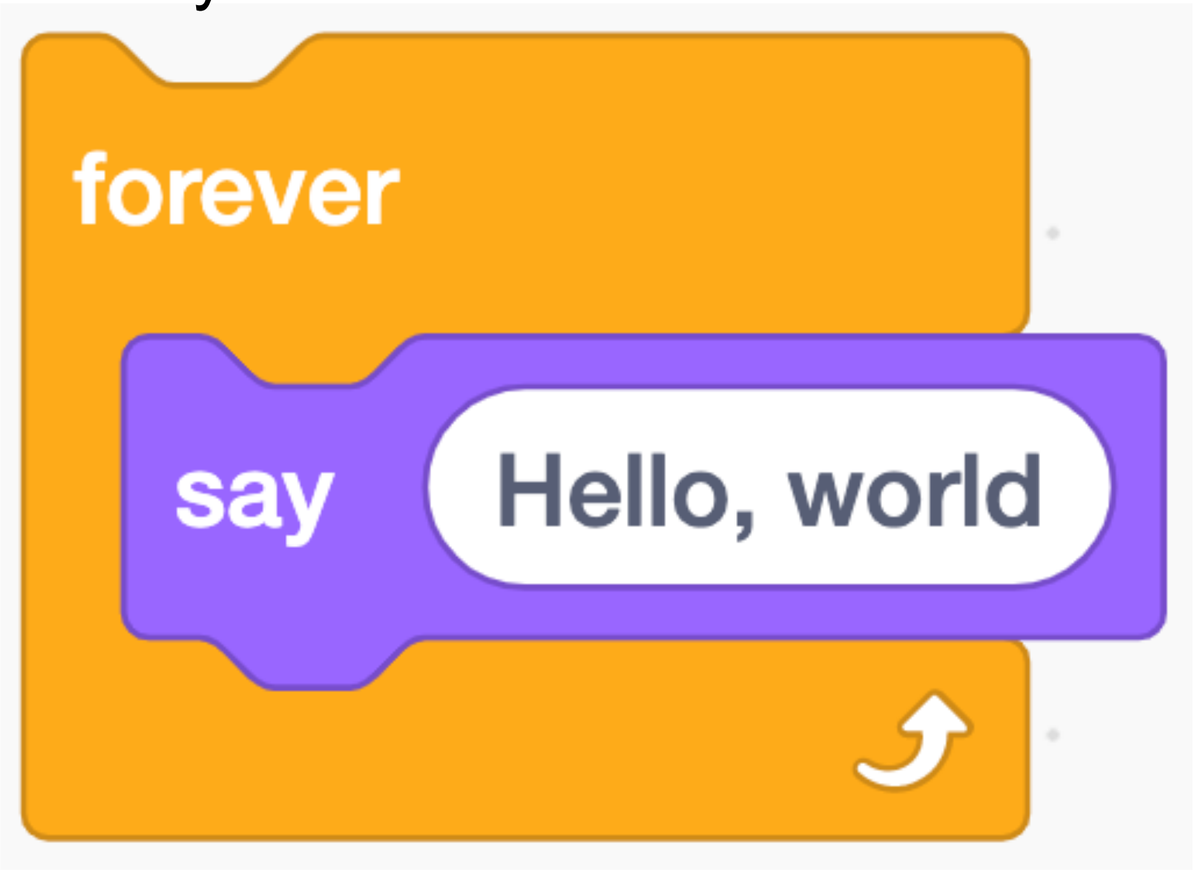
- Use the **while** keyword to create a loop in Python.



while

Loops

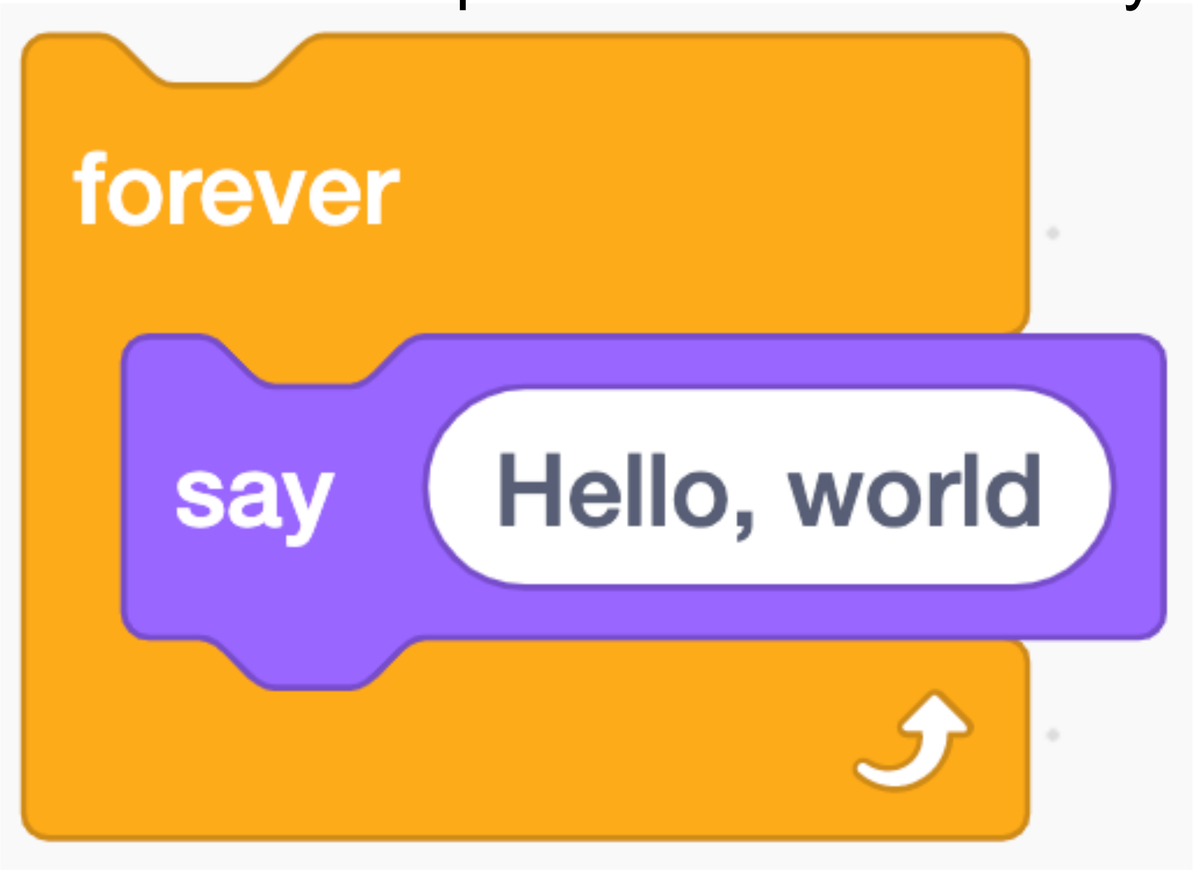
- The **while** keyword also requires **:** (colon).
- Every indented line after the colon will be looped.



while:

Loops

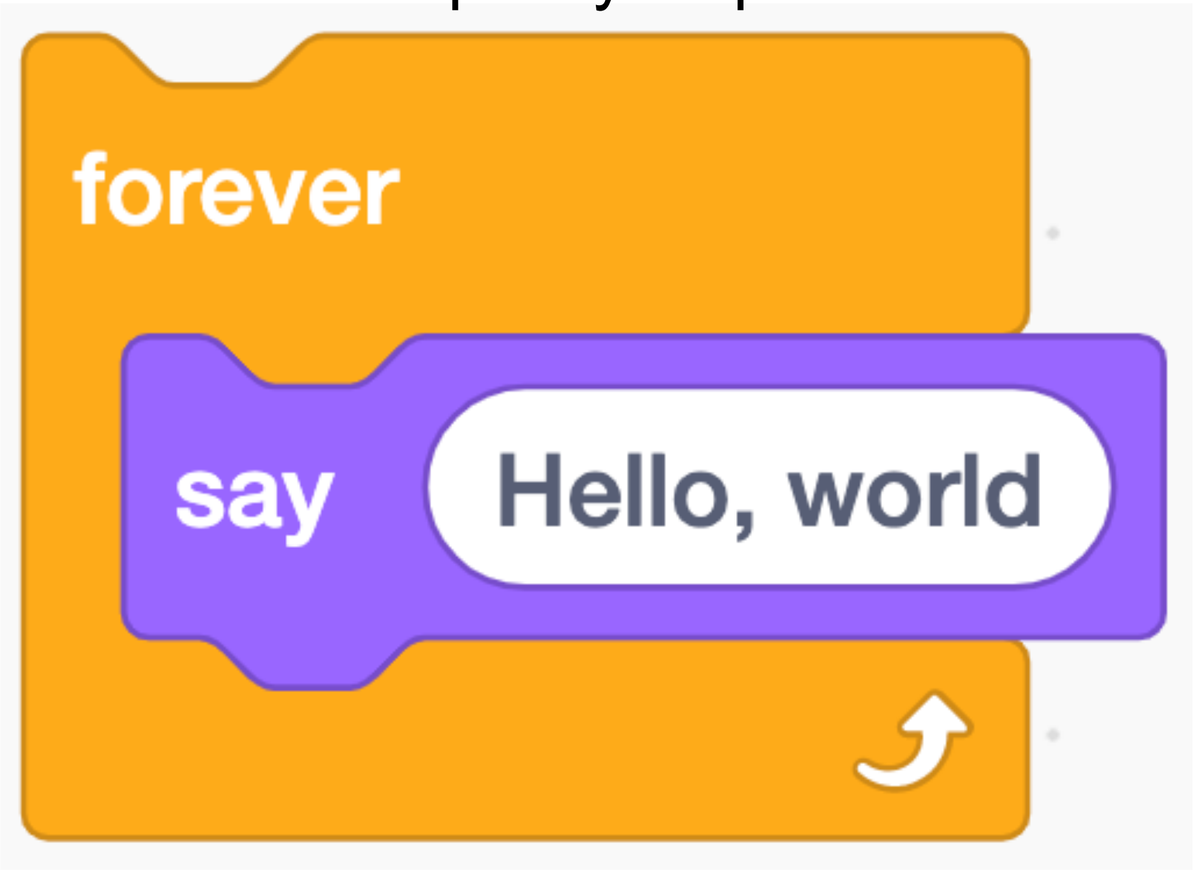
- The **while** keyword also requires a condition to tell the computer when to stop.
- Use a white space after the while keyword to set the condition.




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

Loops

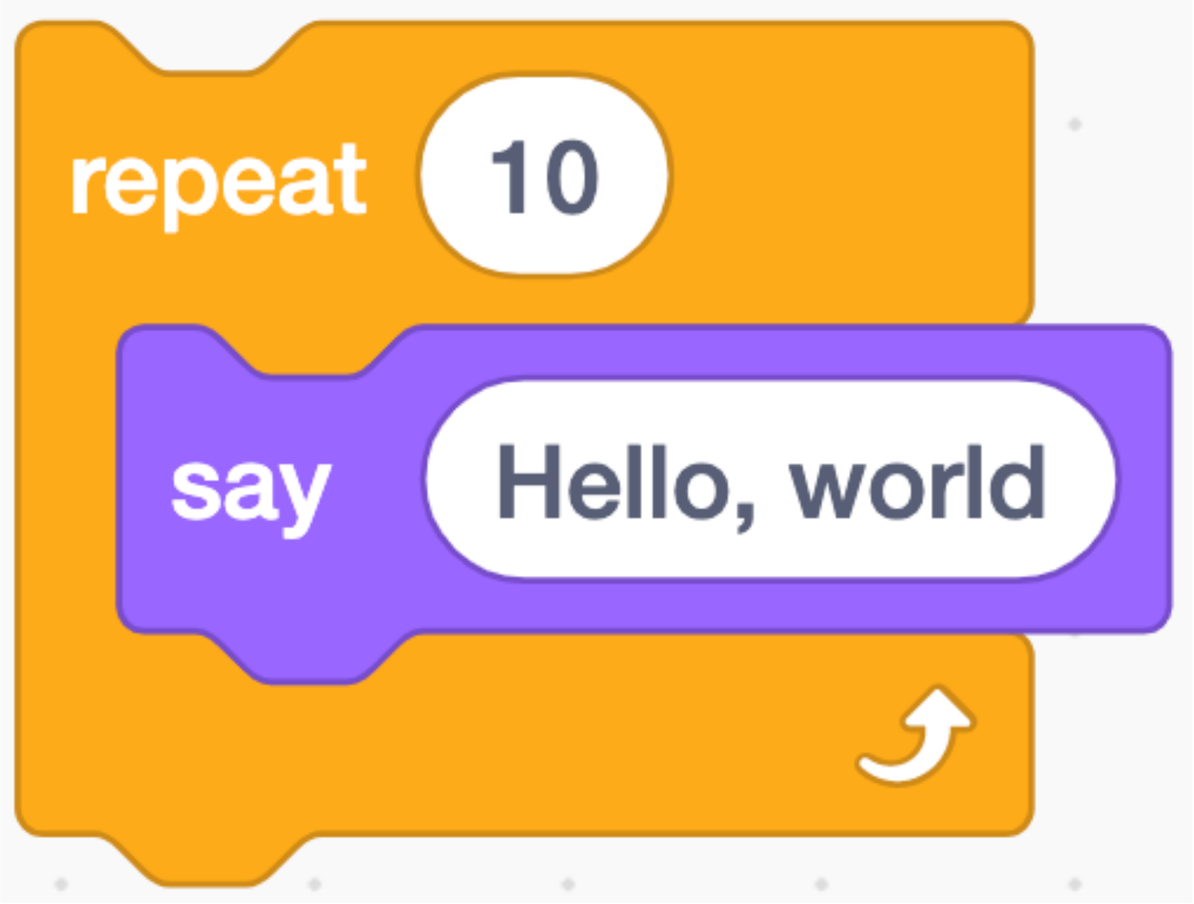
- Use **True** as the Boolean expression to ensure that our loop will run forever.
- The while loop only stops when the boolean expression evaluates to **False**.




```
while True:
  print("hello, world")
```

Loops

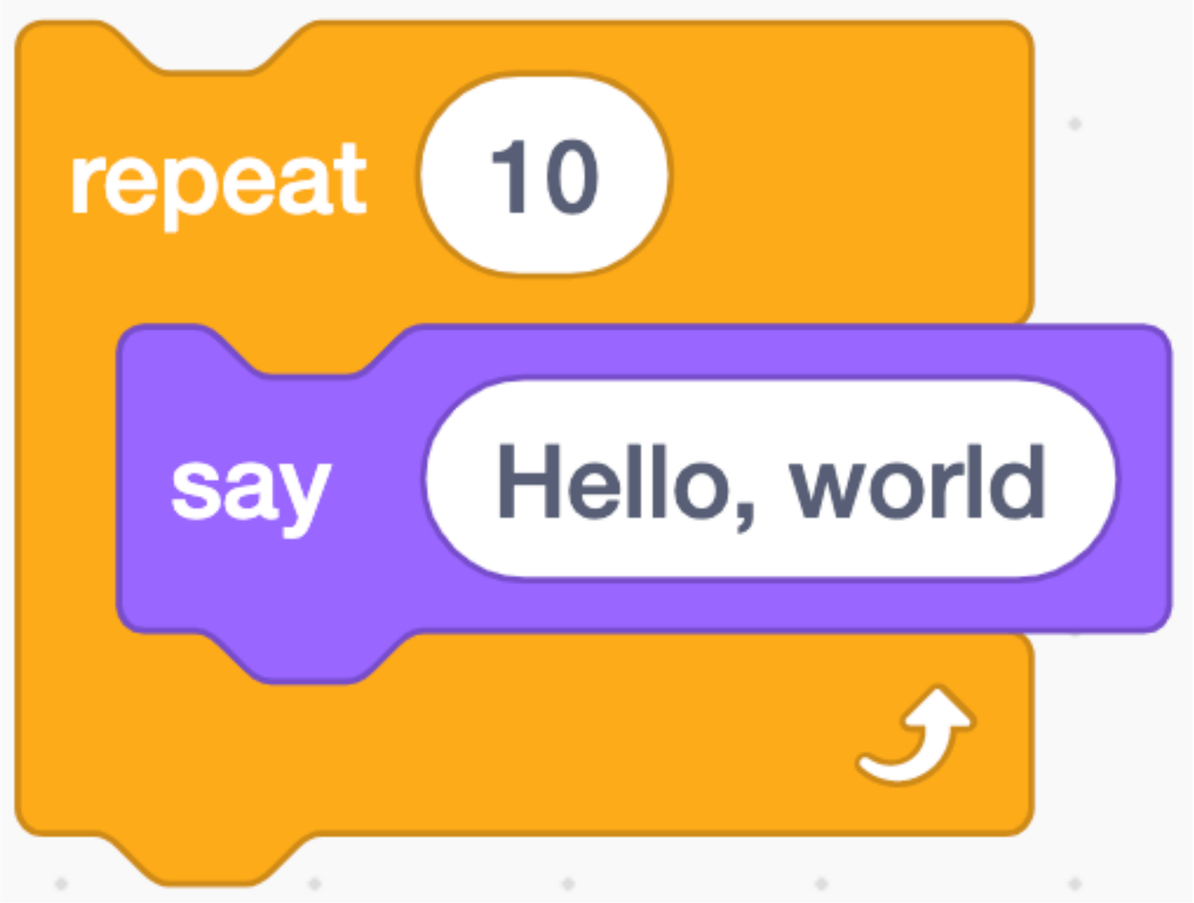
- Use the **for** keyword to create a loop that repeats 10 times.



for

Loops

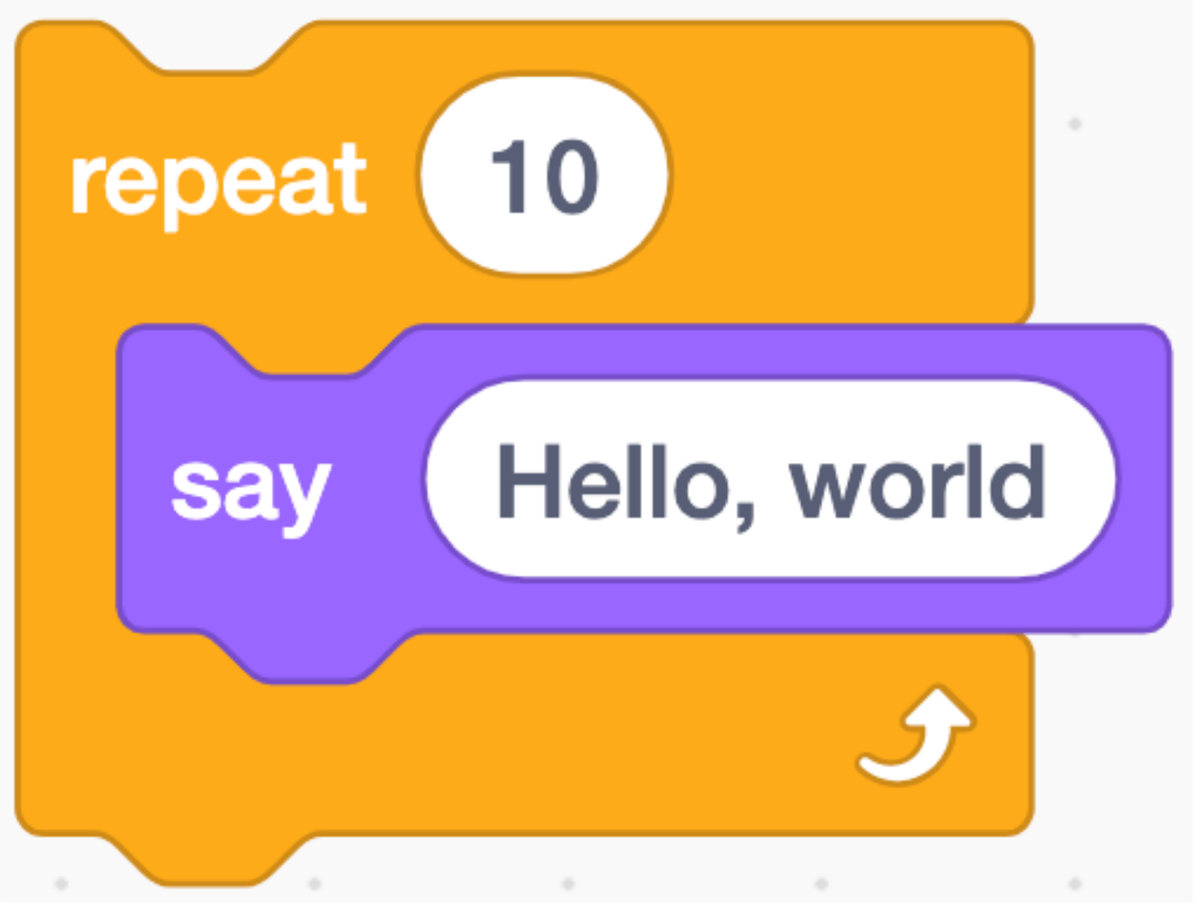
- The **for** keyword also requires **:** (colon).
- Every indented line after the colon will be looped.



for:

Loops

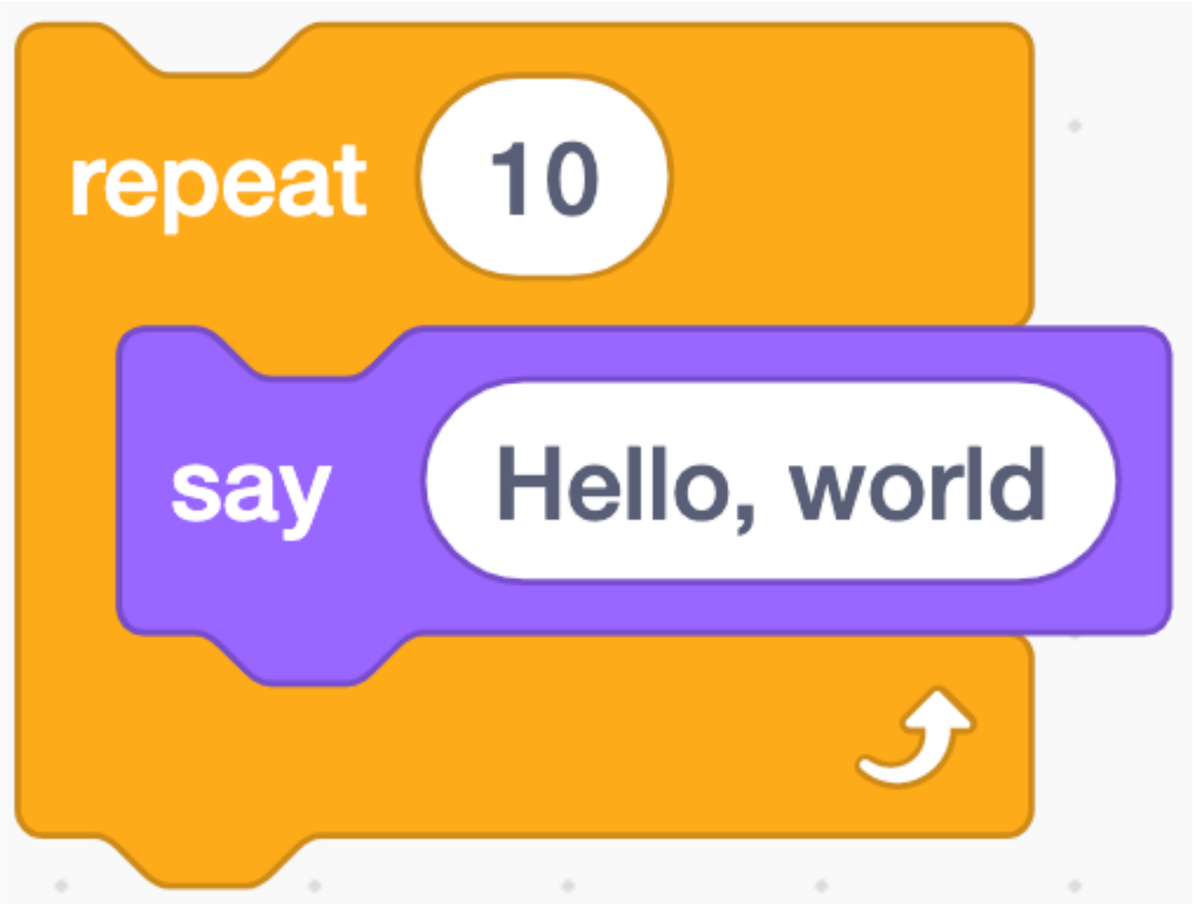
- The **for** keyword also requires a condition to tell the computer when to stop.
- Use the whitespace after the for keyword to set the condition.




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

Loops

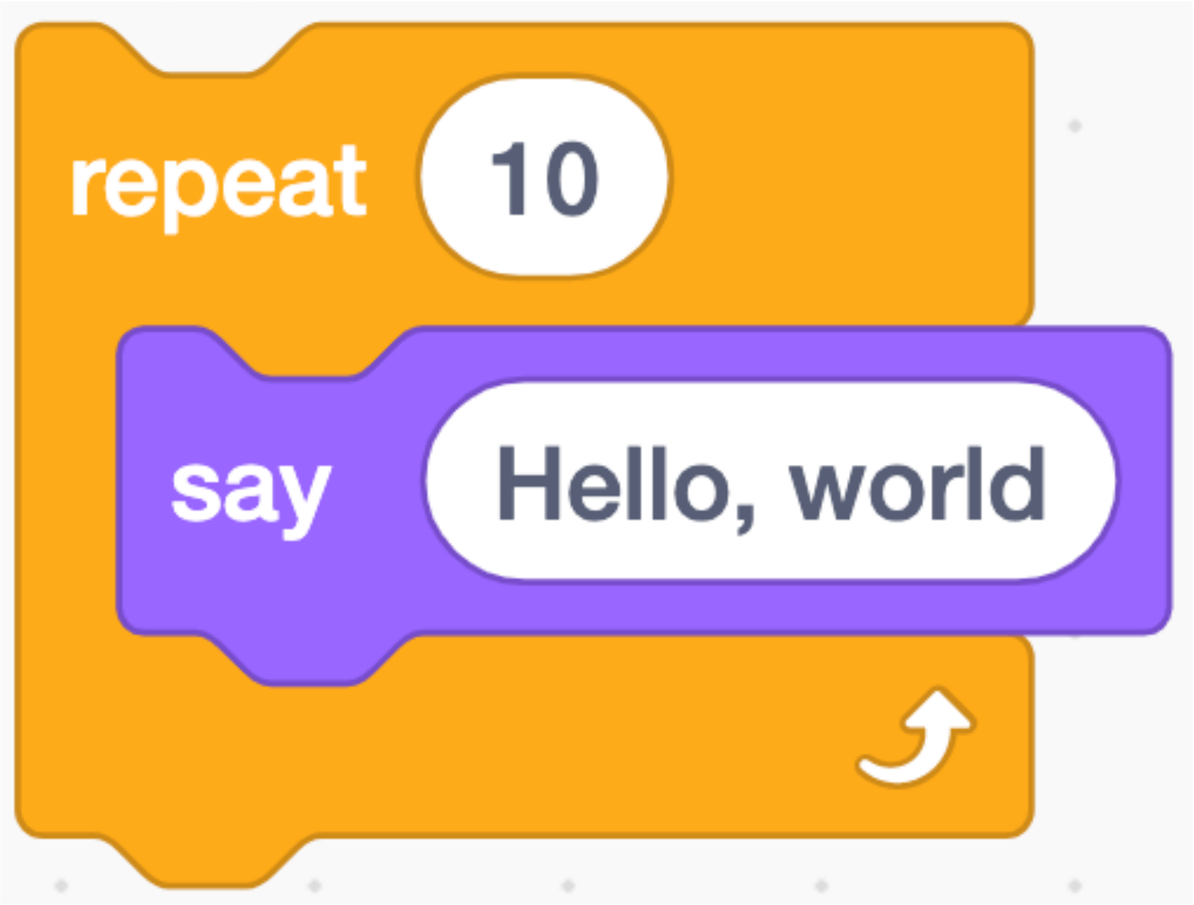
- The **for** keyword needs an expression after the keyword.
- The expression begins with declaring an iterating variable (in this case, **index**)




```
for index:
  print("hello, world")
```

Loops

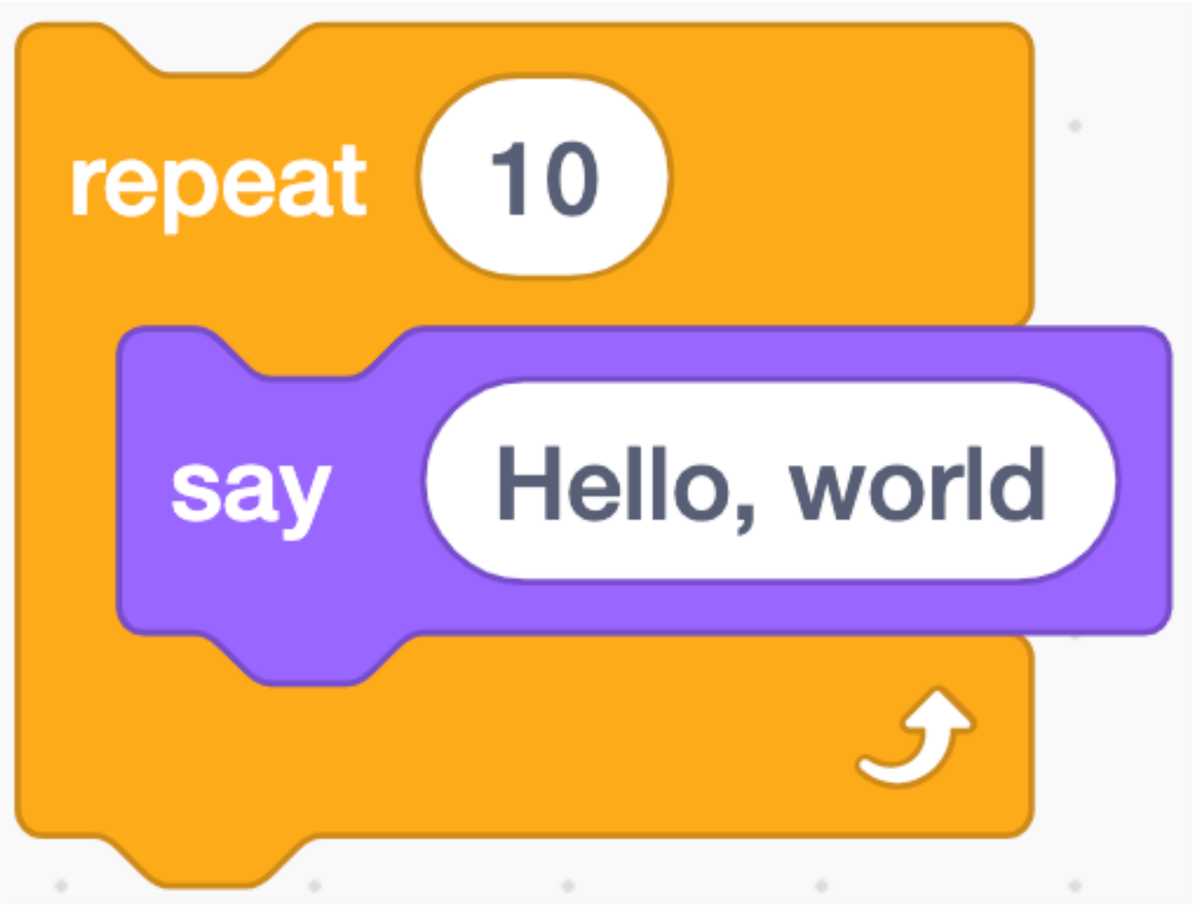
- We then tell the for-loop when it should end using the **range()** keyword.
- range() tells the for-loop the maximum number index should increment to.




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

Loops

- By putting number 10 in the parenthesis, the for-loop knows when to stop (after 10 times).




```
for index in range(10):
    print("hello, world")
```