Introduction to Programming
Lesson 2

Outline

- How it works
- Interactive mode vs Scripting mode
- `print()` function
- `#comments`
- Interactive input/output
- `if/elif/else`
- `while/for`

Lesson Code

goo.gl/l3RFLb

Boolean Practice

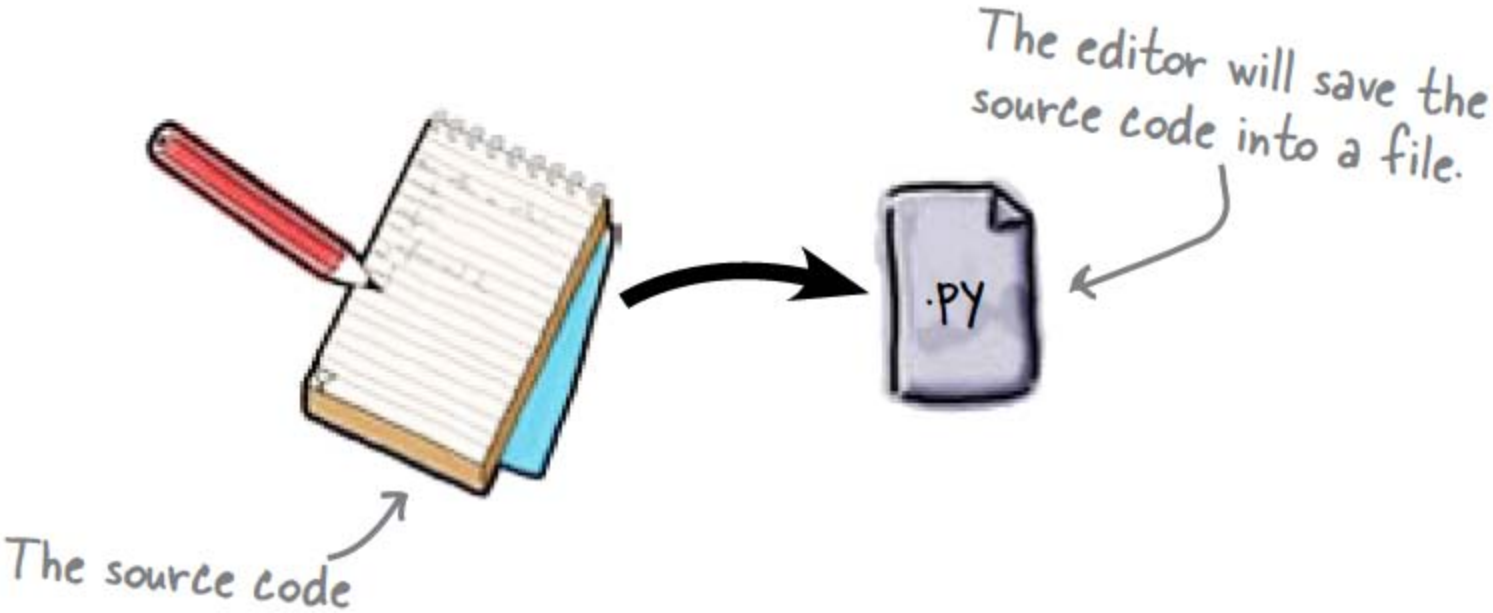
```
1.   True and True
2.   False and True
3.   1 == 1 and 2 == 1
4.   "test" == "test"
5.   1 == 1 or 2 != 1
6.   True and 1 == 1
7.   False and 0 != 0
8.   True or 1 == 1
9.   "test" == "testing"
10.  1 != 0 and 2 == 1
```

```
>>> True and True
True
>>> 1 == 1 and 2 == 2
True
```

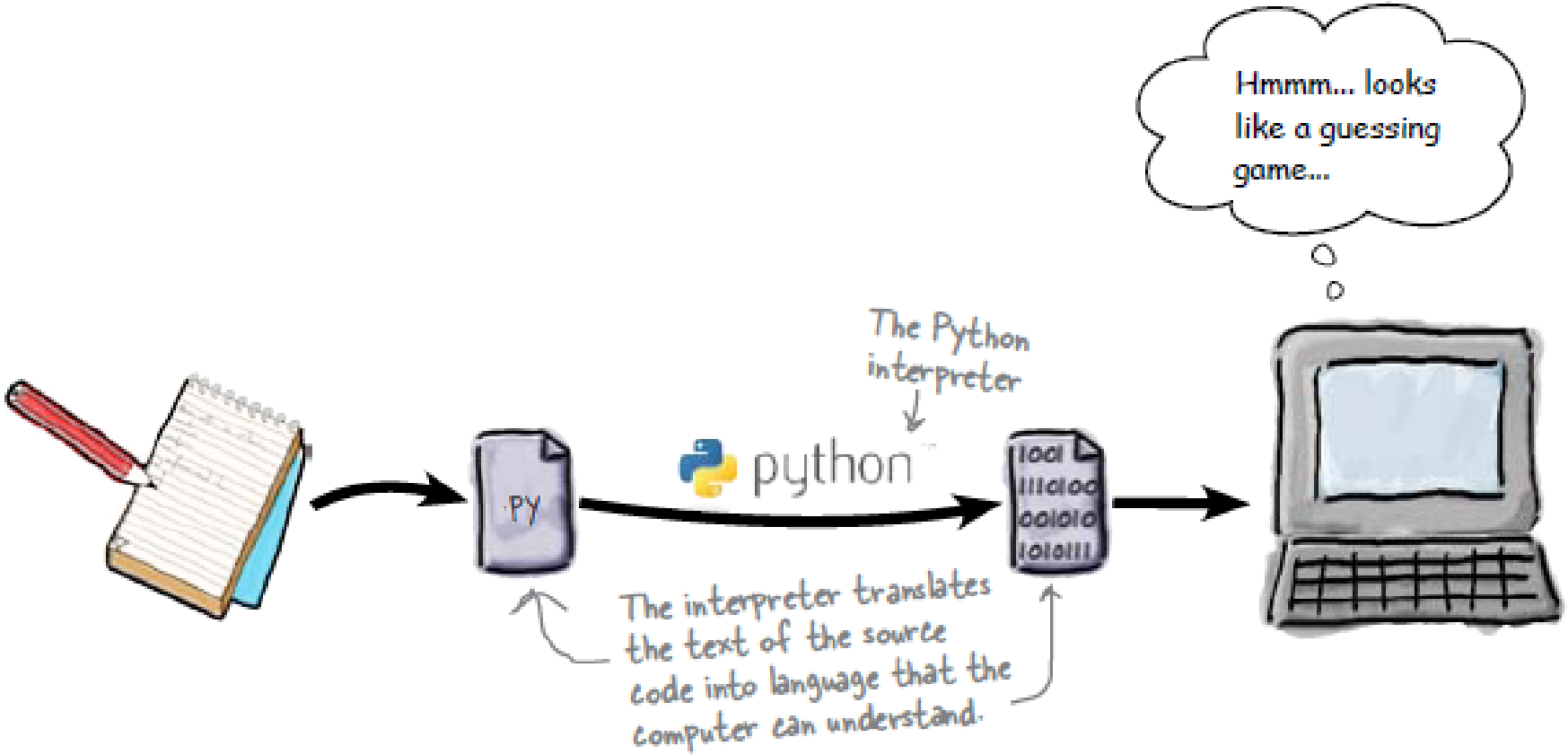
Boolean Practice

```
11.  "test" != "testing"
12.  "test" == 1
13.  not (True and False)
14.  not (1 == 1 and 0 != 1)
15.  not (10 == 1 or 1000 == 1000)
16.  not (1 != 10 or 3 == 4)
17.  not ("testing" == "testing" and "Zed" == "Cool Guy")
18.  1 == 1 and not ("testing" == 1 or 1 == 0)
19.  "chunky" == "bacon" and not (3 == 4 or 3 == 3)
20.  3 == 3 and not ("testing" == "testing" or "Python" == "Fun")
```

How it works


The editor will save the source code into a file.
.PY
The source code

How it works


Hmmm... looks like a guessing game...
The Python interpreter
python
.py
1001
1110100
0010101
1010111
The interpreter translates the text of the source code into language that the computer can understand.

script_1.py

```python
print("Hello World!")
print("Hello Again")
print("I am in a script")
```

Ln : 3 Col : 24 Sel : 0 | 0 Dos\Windows UTF-8 w/o BOM INS

Script: A program stored in a file.

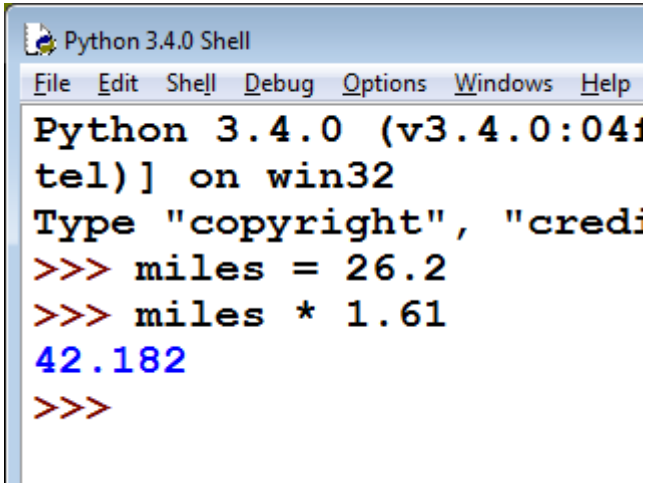
C:\Windows\system32\cmd.exe

```
C:\Users\Sparta\Desktop\lesson2>python script_1.py
Hello World!
Hello Again!
I am in a script!

C:\Users\Sparta\Desktop\lesson2>
```

Interactive vs Scripting mode

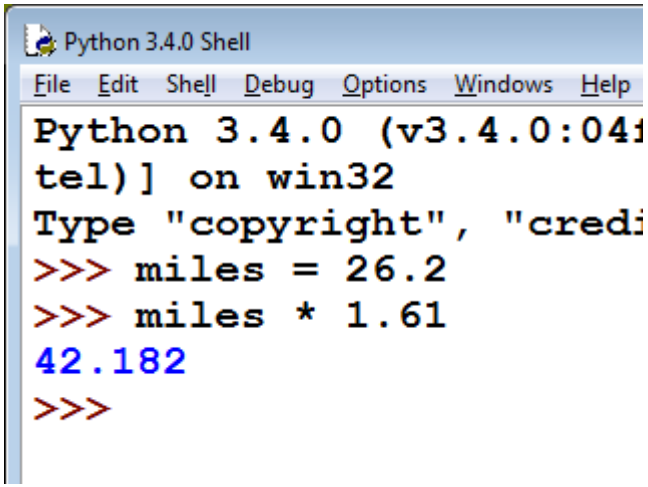
Interactive vs Scripting mode

- Interactive mode

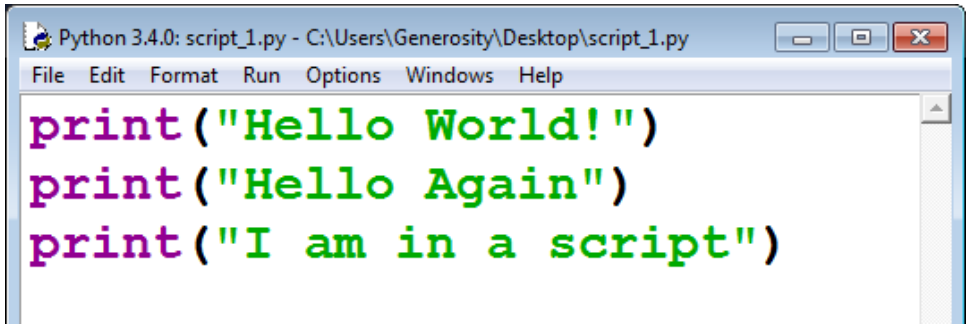


Interactive vs Scripting mode

- Interactive mode



- Scripting mode



Exercise 1

- Ստեղծեք ex1.py
- Գրեք file-ի մեջ

```
x = 5
y = 6
z = x + y
```

- Run it
- Տպեք z-ի արժեքը

Exercise 1

- Ստեղծեք ex1.py
- Գրեք file-ի մեջ

```
x = 5
y = 6
z = x + y
```

- Run it
- Տպեք z-ի արժեքը

```
x = 5
y = 6
z = x + y
print(z)
```

ex1.py

#Մեկնաբանություն
#No Comments ☺

#Մեկնաբանություն
#No Comments ☺

#

#Մեկնաբանություն
#No Comments ☺

```
#
# x = 5
```

#Մեկնաբանություն
#No Comments ☺


```
#
# x = 5
# z = x
```

#Մեկնաբանություն
#No Comments ☺


```
#
# x = 5
# z = x
```

```
# A comment, you can read later.
# Anything after the # is ignored by python.
print ("This is Sparta") # and the comment after is ignored

# comment out a piece of code:
# print("Don't run.")
print("Run")
```

comments.py

Program-ի հնարը

```
line1 = 'Hello Python developer...'
line2 = 'Welcome to the world of Python!'
print(line1)
print(line2)
```

hello.py

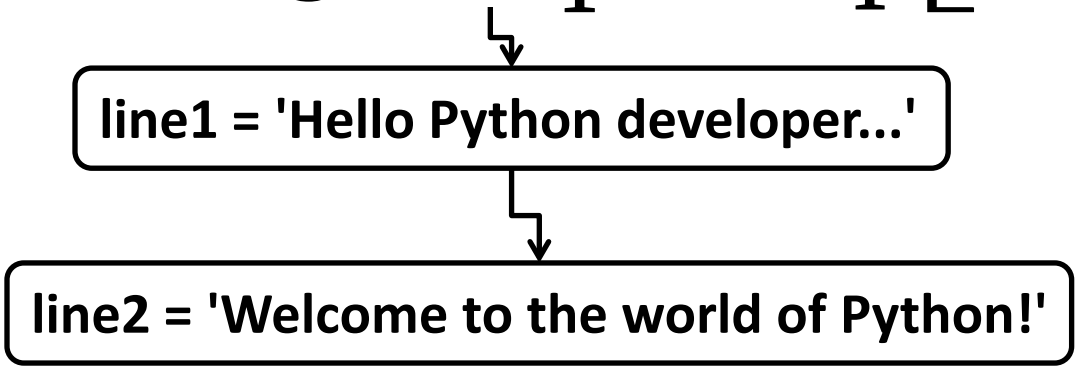
Program-h hnupp

line1 = 'Hello Python developer...'

```
line1 = 'Hello Python developer...'
line2 = 'Welcome to the world of Python!'
print(line1)
print(line2)
```

hello.py

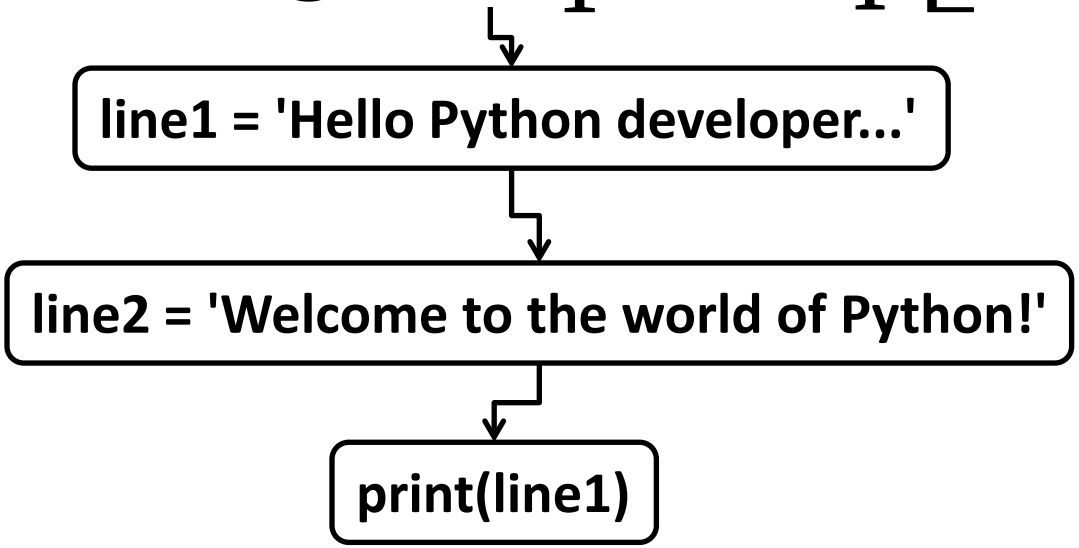
Program-ի հոսքը




```
line1 = 'Hello Python developer...'
line2 = 'Welcome to the world of Python!'
print(line1)
print(line2)
```

hello.py

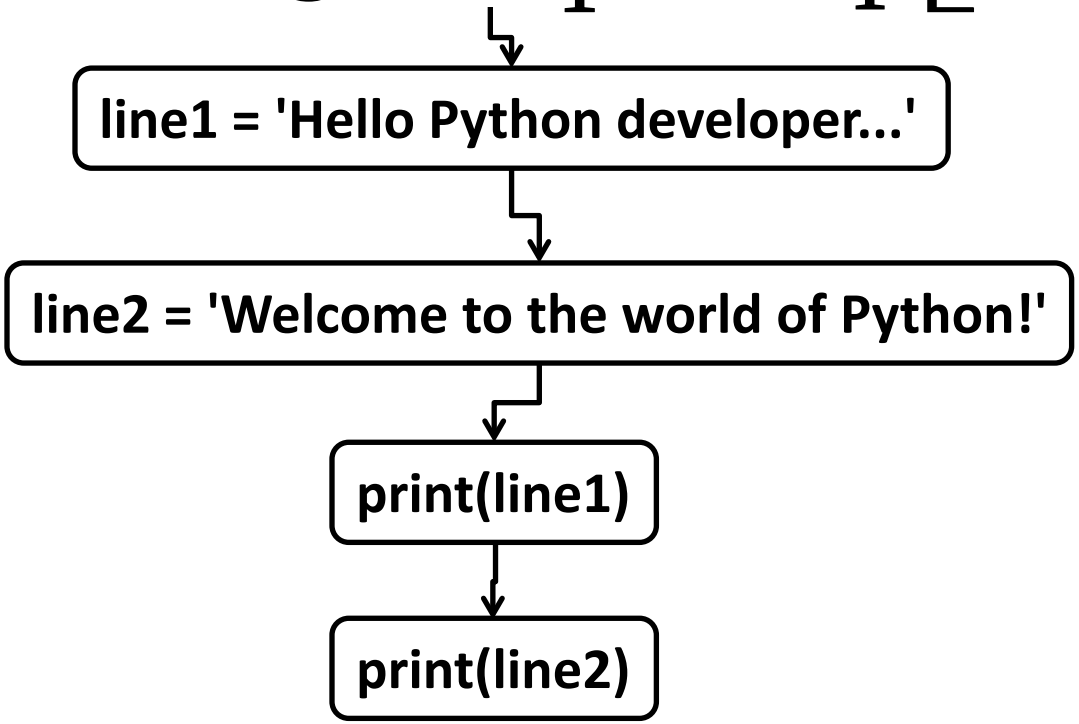
Program-ի հոսքը




```
line1 = 'Hello Python developer...'
line2 = 'Welcome to the world of Python!'
print(line1)
print(line2)
```

hello.py

Program-h hnupp




```
line1 = 'Hello Python developer...'
line2 = 'Welcome to the world of Python!'
print(line1)
print(line2)
```

hello.py

Exercise 2: Type and run

1. Ստեղծեք ex2.py file
2. file-ի մեջ գրեք

```
print ("Is it greater?", 5 > - 2)
print ("Is it greater or equal?", 5 >= - 2)
print ("Is it less or equal?", 5 <= - 2)
```

ex2.py

input() function

input() function

```
>>> person = input('Enter your name: ')
Enter your name: Kiazh
```

input() function

```
>>> person = input('Enter your name: ')
Enter your name: Kiazh
>>> print('Hello', person)
Hello Kiazh
```

input() function

```
>>> person = input('Enter your name: ')
Enter your name: Kiazh
>>> print('Hello', person)
Hello Kiazh
>>> person
'Kiazh'
>>> type(person)
<class 'str'>
```

input() function

```
>>> person = input('Enter your name: ')
Enter your name: Kiazh
>>> print('Hello', person)
Hello Kiazh
>>> person
'Kiazh'
>>> type(person)
<class 'str'>
```

OPTIONAL

```
>>> person = input()
```

input() function

```
>>> person = input('Enter your name: ')
Enter your name: Kiazh
>>> print('Hello', person)
Hello Kiazh
>>> person
'Kiazh'
>>> type(person)
<class 'str'>
```

OPTIONAL

```
>>> person = input()
```

Միշտ **string**

Exercise 3

1. Ստեղծիր ex3.py file
2. file-ի մեջ գրի

```
name = input('Enter your first name: ')
text = 'Hello’ + name
print(text)
print('Welcome to the world of Python!')
```

ex3.py

3. Run/Execute

eval() function

input()-ը միշտ վերադարձնում է string.

Մեզ պետք է int, float

eval() function

input()-ը միշտ վերադարձնում է string.

Մեզ պետք է int, float

Solution 1: Use type conversion

eval() function

input()-ը միշտ վերադարձնում է string.

Մեզ պետք է int, float

Solution 1: Use type conversion

Solution 2: Use eval()

eval() function

input()-ը միշտ վերադարձնում է string.

Մեզ պետք է int, float

Solution 1: Use type conversion

Solution 2: Use eval()

```
>>> age = input('Enter
your age: ')
Enter your age: 18
>>> age
'18'
```

eval() function

input()-ը միշտ վերադարձնում է string.

Մեզ պետք է int, float

Solution 1: Use type conversion

Solution 2: Use eval()

```
>>> age = input('Enter
your age: ')
Enter your age: 18
>>> age
'18'
>>> int(age)
18
```

eval() function

input()-ը միշտ վերադարձնում է string.

Մեզ պետք է int, float

Solution 1: Use type conversion

Solution 2: Use eval()

eval()

1. վեցնում է string as input
2. վերլուծում է որպես Python expression

```
>>> age = input('Enter your age: ')
Enter your age: 18
>>> age
'18'
>>> int(age)
18
>>> eval('18')
18
>>> eval('age')
'18'
>>> eval('[2,3+5]')
[2, 8]
>>> eval('x')
Traceback (most recent call last):
```

Type Conversion functions

int(), float(), str(),…

Type Conversion functions

int(), float(), str(),…

```
>>> type(32)
<class 'int'>
>>> int('Hello')
ValueError…
>>> int(3.99)
3
>>> int(-2.3) #chops of the fraction, no rounding
-2
```

Type Conversion functions

int(), float(), str(),…

```
>>> type(32)
<class 'int'>
>>> int('Hello')
ValueError…
>>> int(3.99)
3
>>> int(-2.3) #chops of the fraction, no rounding
-2
```

```
>>> float(32)
32.0
>>> float(3.1415)
3.1415
>>> str(32)
'32'
>>> str(3.14)
'3.14'
```

		Built-in Functions		
abs()	dict()	help()	min()	setattr()
all()	dir()	hex()	next()	slice()
any()	divmod()	id()	object()	sorted()
ascii()	enumerate()	input()	oct()	staticmethod()
bin()	eval()	int()	open()	str()
bool()	exec()	isinstance()	ord()	sum()
bytearray()	filter()	issubclass()	pow()	super()
bytes()	float()	iter()	print()	tuple()
callable()	format()	len()	property()	type()
chr()	frozenset()	list()	range()	vars()
classmethod()	getattr()	locals()	repr()	zip()
compile()	globals()	map()	reversed()	__import__()
complex()	hasattr()	max()	round()	
delattr()	hash()	memoryview()	set()	

Python interpreter has a number of functions and types built into it that are always available

>>>dir(__builtins__)

Exercise 4A

Գրեք ծրագիր որը վերցնում է user-ից թիվ և վերադարձնում այդ թվի կրկնապատիկը

```
x = input('Enter number:')
y = 2*int(x)
print(y)
```

ex4.py

Exercise 4B

Գրեք ծրագիր որը

- Հարցնում է user-ի տարիքը
- Հետո տպում է +1 ավելացնելով

```
>>>
Enter your age: 17
You will be 18 next year!
```

Exercise 4B

Գրեք ծրագիր որը

- Հարցնում է user-ի տարիքը
- Հետո տպում է +1 ավելացնելով

```
>>>
Enter your age: 17
You will be 18 next year!
```

```
age = input('Enter your age: ')
age = int(age)
next = age + 1
line ='You will be ' + str(next) + ' next year!'
print(line)
```

ex4.py

if statement

```
if <boolean expression>:
    <indented code block>
<non-indented statement>
```

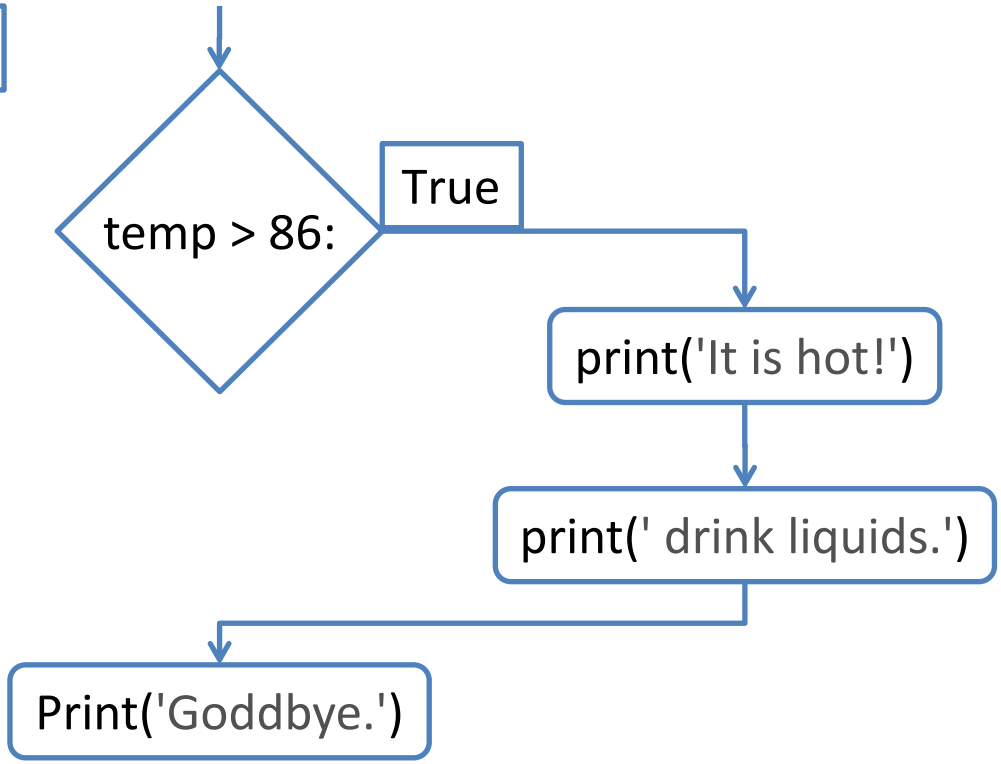
```
if temp > 86:
    print('It is hot!')
    print(' drink liquids.')
print('Goodbye.')
```

if statement

```
if <boolean expression>:
    <indented code block>
<non-indented statement>
```

```
if temp > 86:
    print('It is hot!')
    print(' drink liquids.')
print('Goodbye.')
```

The value of temp is 90.



if statement

```
if <boolean expression>:
    <indented code block>
<non-indented statement>
```

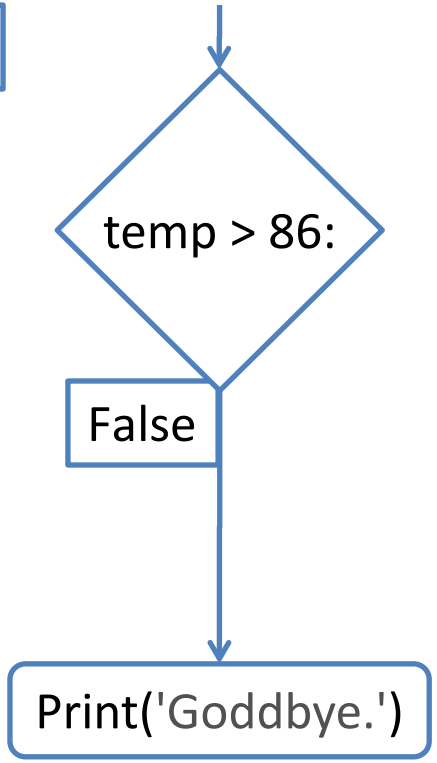
```
if temp > 86:
    print('It is hot!')
    print(' drink liquids.')
print('Goodbye.')
```

if statement

```
if <boolean expression>:
    <indented code block>
<non-indented statement>
```

```
if temp > 86:
    print('It is hot!')
    print(' drink liquids.')
print('Goodbye.')
```

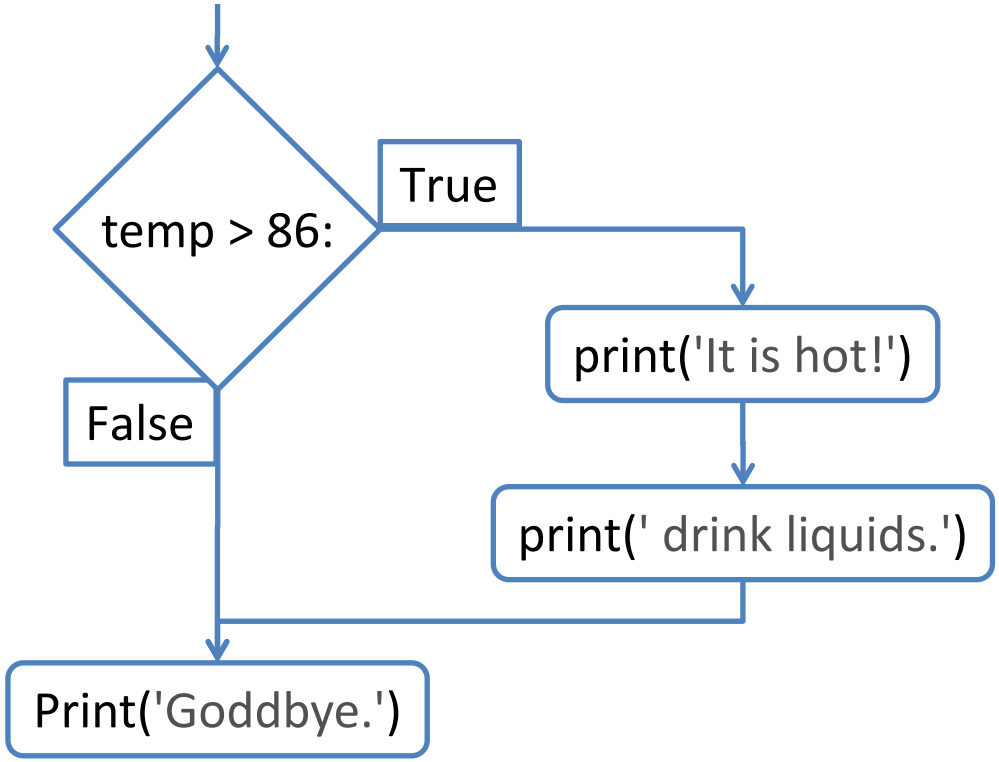
The value of temp is 50.



if statement

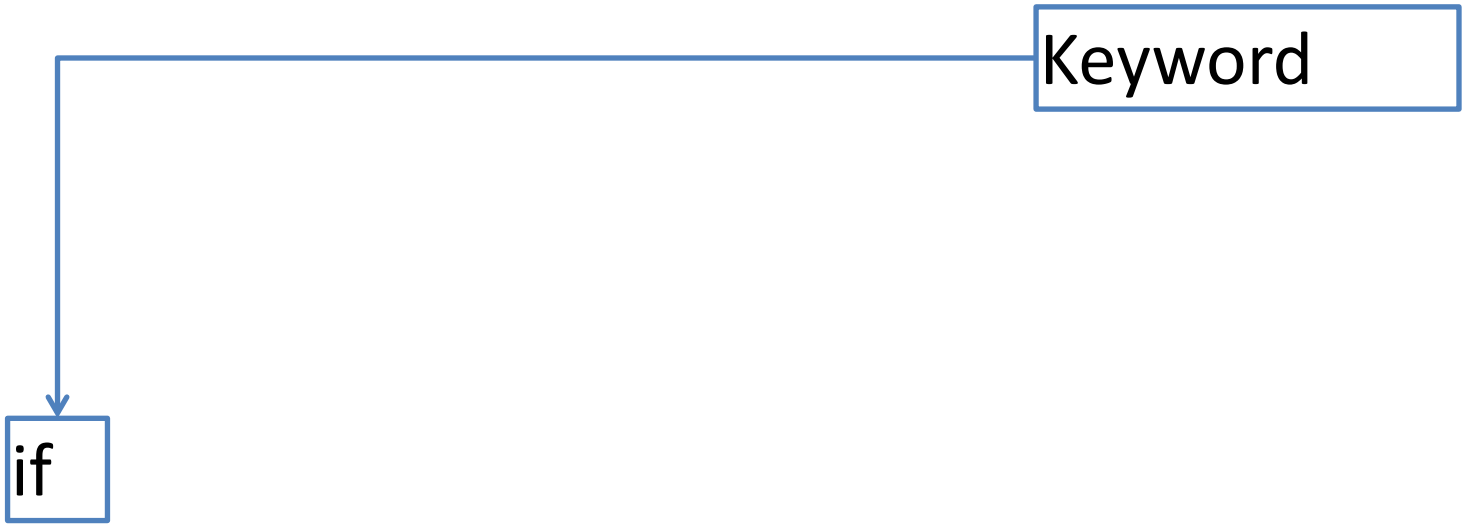
```
if <boolean expression>:
    <indented code block>
<non-indented statement>
```

```
if temp > 86:
    print('It is hot!')
    print(' drink liquids.')
print('Goodbye.')
```

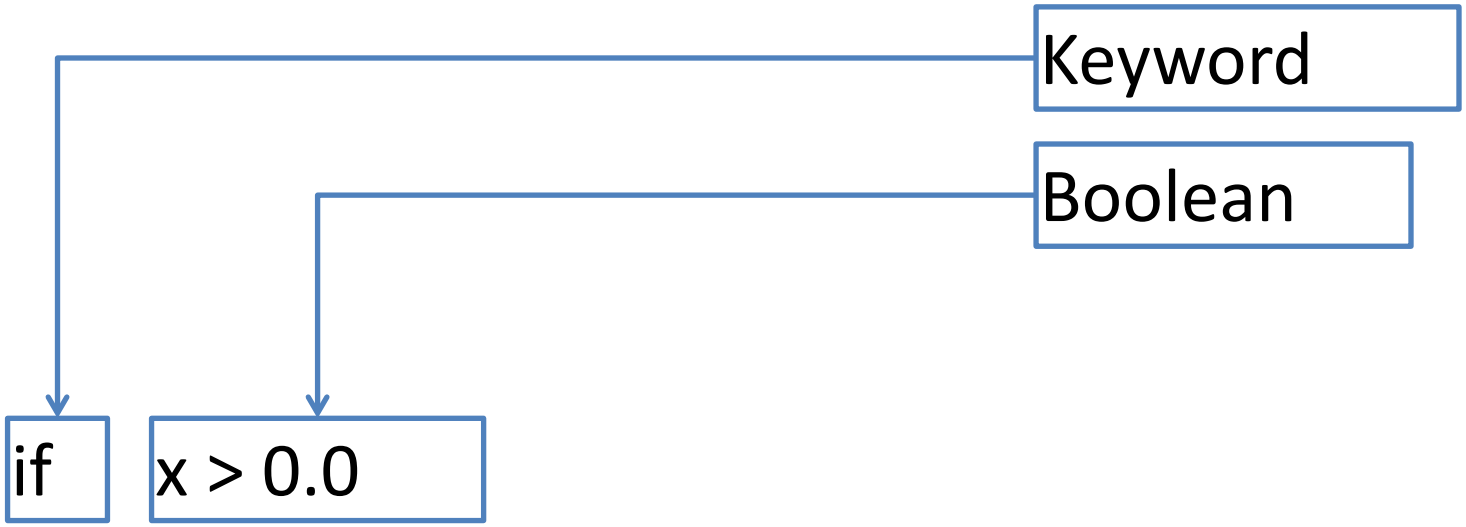


if statement

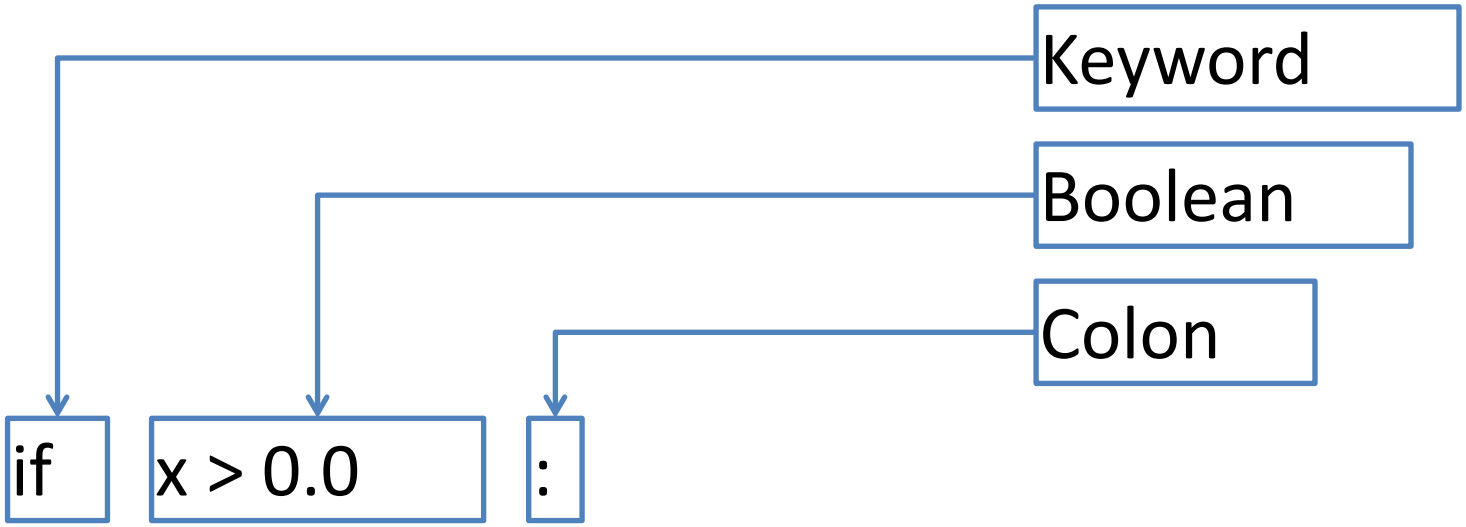
if statement



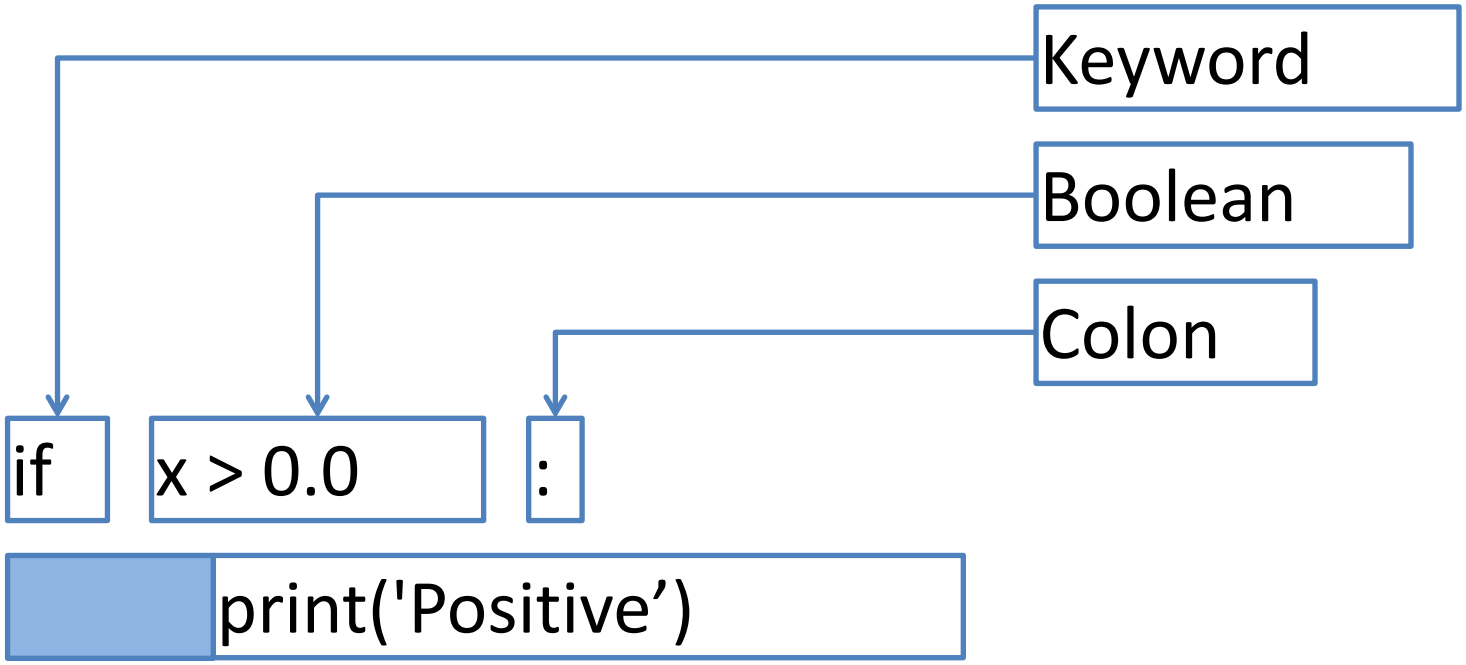
if statement



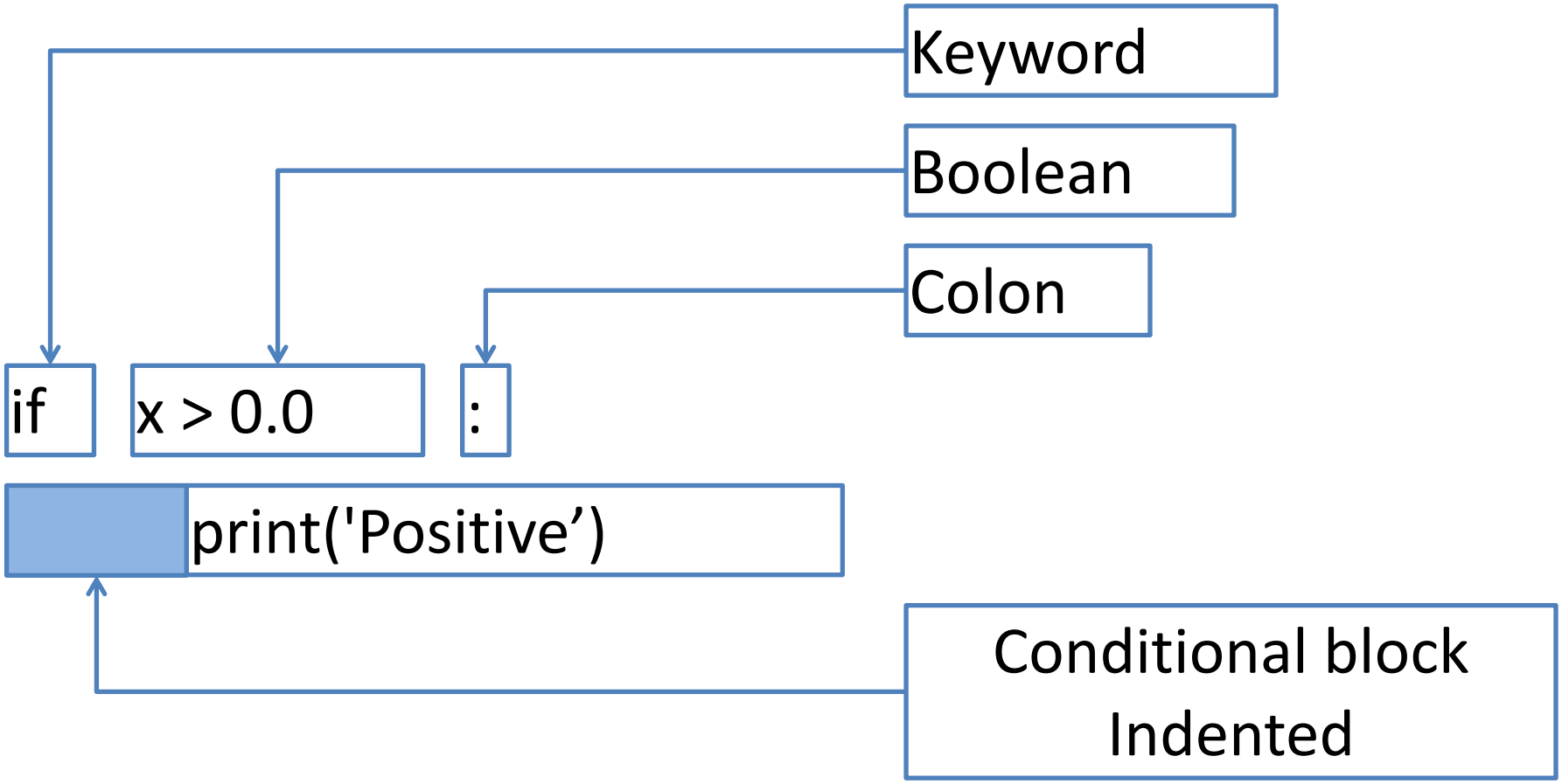
if statement



if statement



if statement



Indentation

- A Block => Group of programming statements

```
>>> age = 25
>>> if age > 20:
        print('You are too old!')
        print('Why are you here?')
```

Indentation

- A Block => Group of programming statements

```
>>> age = 25
>>> if age > 20:
        print('You are too old!')
        print('Why are you here?')

>>> age = 25
>>> if age > 20:
    □□□□print('You are too old!')
    □□□□print('Why are you here?')
```

4 հատ
space (պռաբել)

Indentation

- A Block => Group of programming statements

```
>>> age = 25
>>> if age > 20:
        print('You are too old!')
        print('Why are you here?')

>>> age = 25
>>> if age > 20:
    □□□□print('You are too old!')
    □□□□print('Why are you here?')
```

Միշտ :

4 հատ
space (պռաբել)

Indentation

- A Block => Group of programming statements

```
>>> age = 25
>>> if age > 20:
        print('You are too old!')
        print('Why are you here?')
```

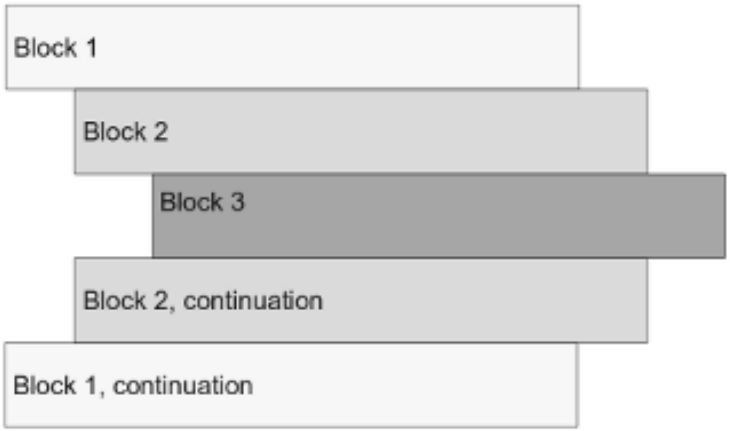
```
>>> age = 25
>>> if age > 20:
    □□□□print('You are too old!')
    □□□□print('Why are you here?')
```

Միշտ :

4 հատ
space (պռաբել)

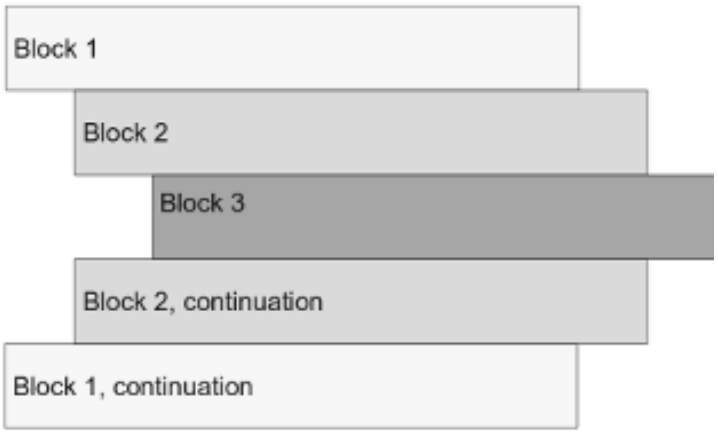
Blocks via Indentation

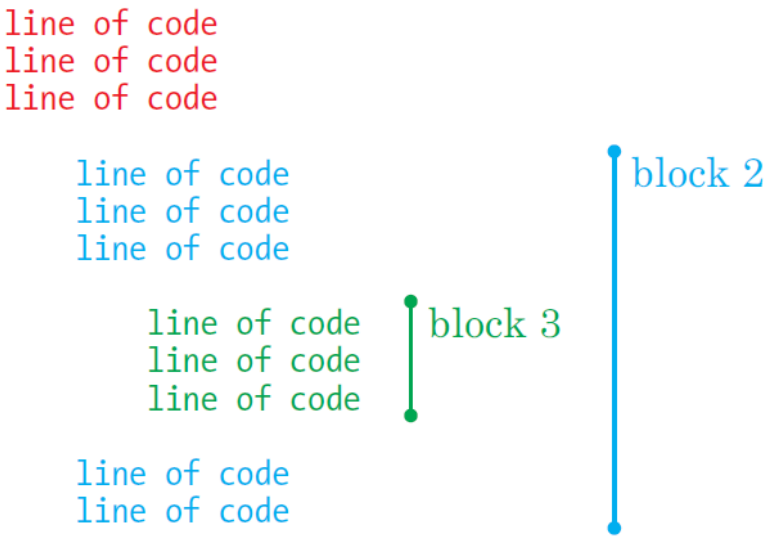
- A Block => Group of programming statements



Blocks via Indentation

- A Block => Group of programming statements





Indentation

```
x = 34 - 23        # A comment.
y = "Hello"
z = 3.45
if z == 3.45 or y == "Hello":
    x = x + 1
    y = y + " World"
print(x)
print(y)
```

Indentation

```
x = 34 - 23        # A comment.
y = "Hello"
z = 3.45
if z == 3.45 or y == "Hello":
    x = x + 1
    y = y + " World"
print(x)
print(y)
```

Exercise 5

```
Enter your age: 22
Armen, you can vote!
```

Գրեք ծրագիր որը

- Հարցնում է user-ի տարիքը
- Եթե (IF) տարիքը 20-ից > է print(“You can vote”)

Exercise 5

```
Enter your age: 22
Armen, you can vote!
```

Գրեք ծրագիր որը

- Հարցնում է user-ի տարիքը
- Եթե (IF) տարիքը 20-ից > է print(“You can vote”)

```
name = input('Enter your name: ')
age = input('Enter your age: ')
if int(age)>20:
      print('you can vote')
```

ex4.py

Indentation is critical

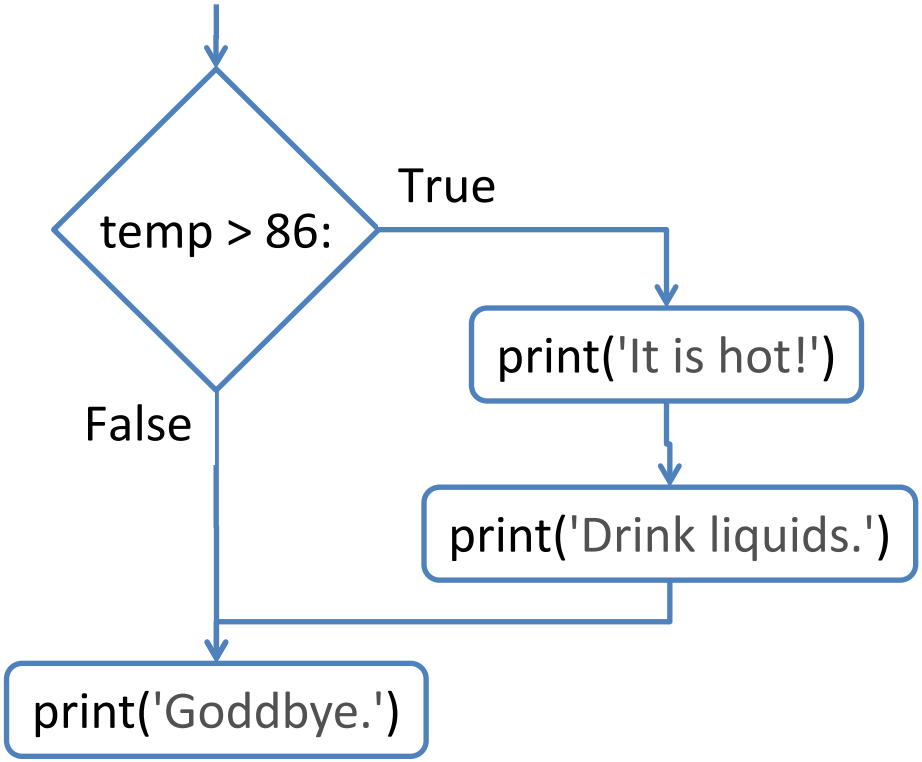
```
if temp > 86:
    print('It is hot!')
    print('Drink liquids.')
    print('Goodbye.')
```

```
if temp > 86:
    print('It is hot!')
    print('Drink liquids.')
print('Goodbye.')
```

Indentation is critical

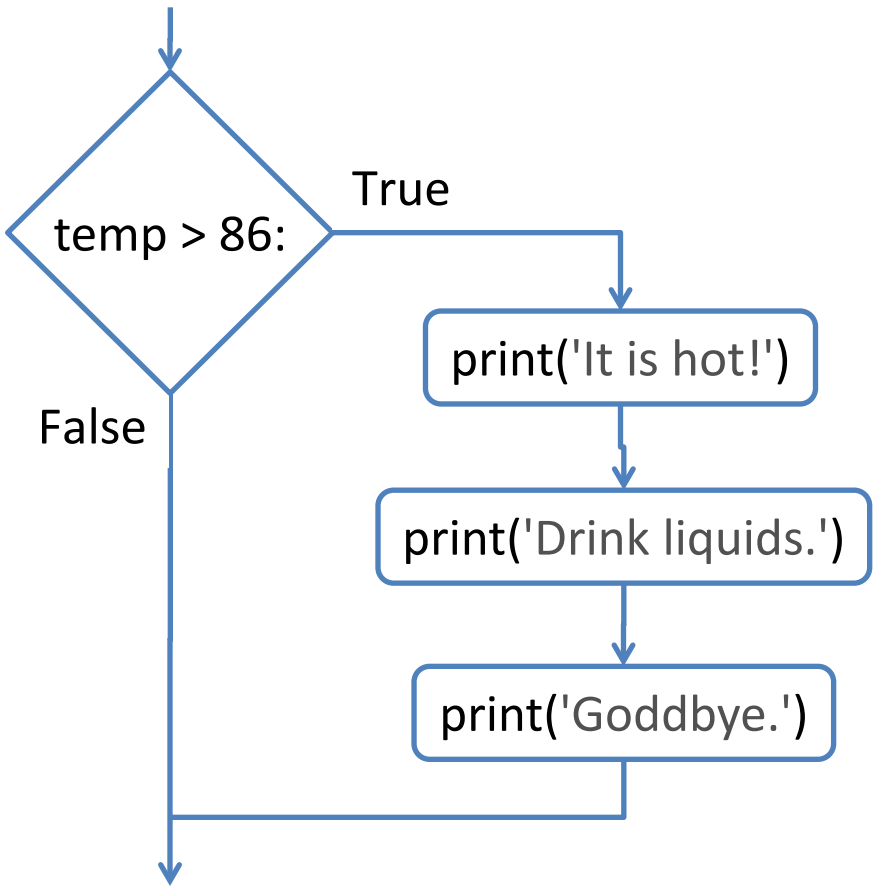
```
if temp > 86:
    print('It is hot!')
    print('Drink liquids.')
    print('Goodbye.')
```

```
if temp > 86:
    print('It is hot!')
    print('Drink liquids.')
print('Goodbye.')
```

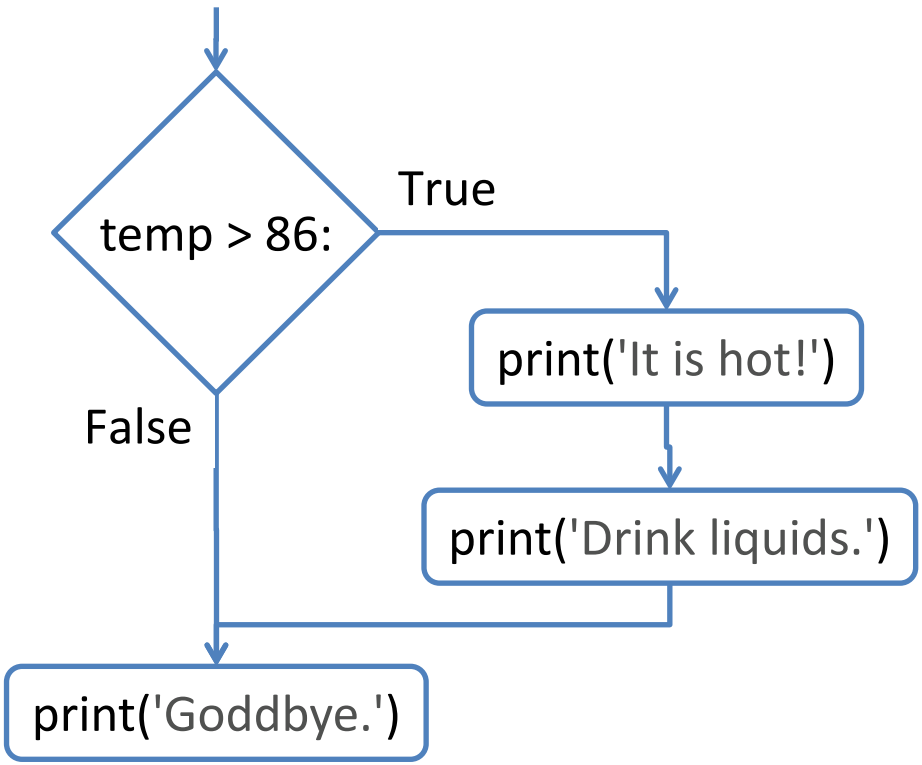


Indentation is critical

```
if temp > 86:
    print('It is hot!')
    print('Drink liquids.')
    print('Goodbye.')
```

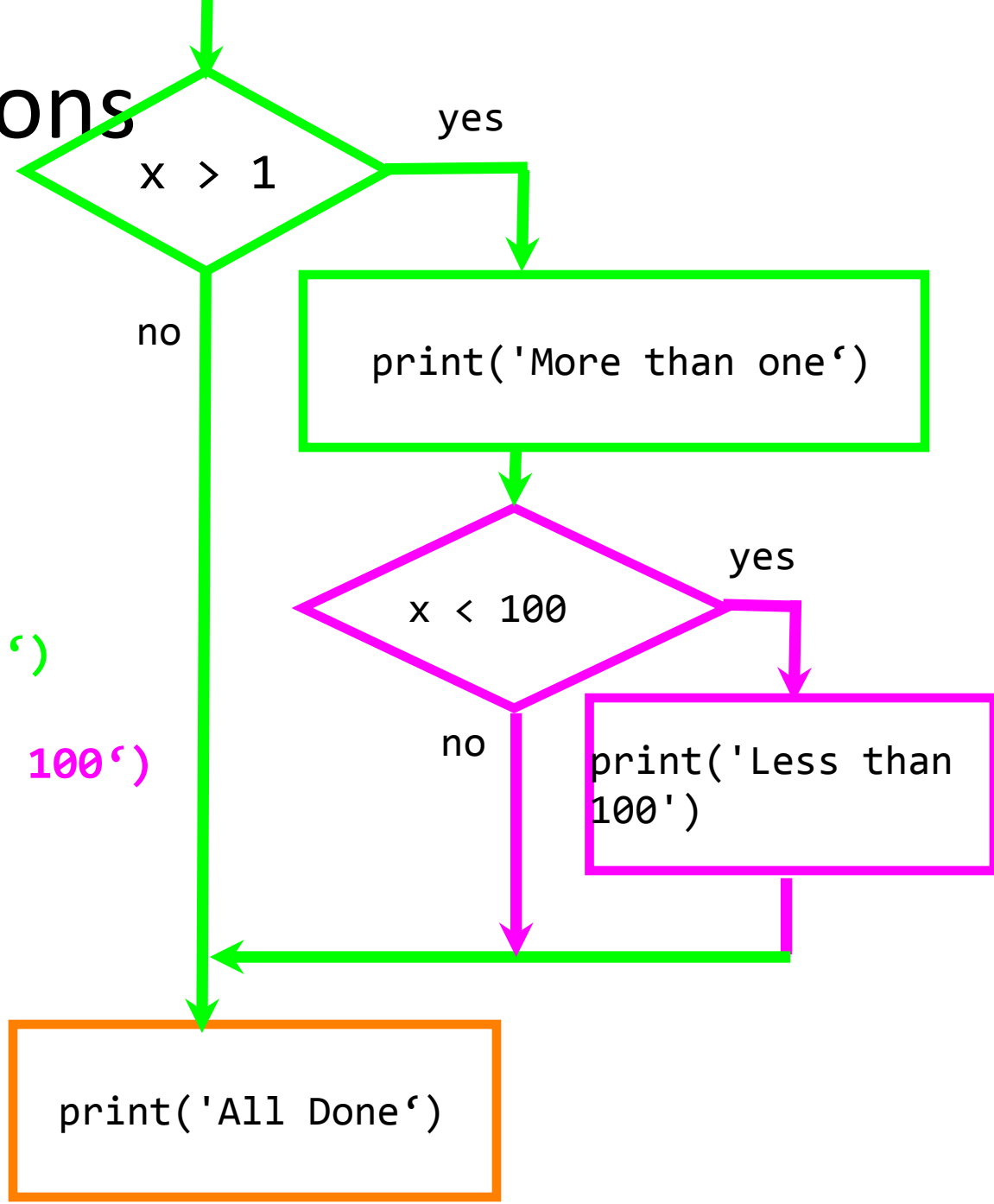



```
if temp > 86:
    print('It is hot!')
    print('Drink liquids.')
print('Goodbye.')
```



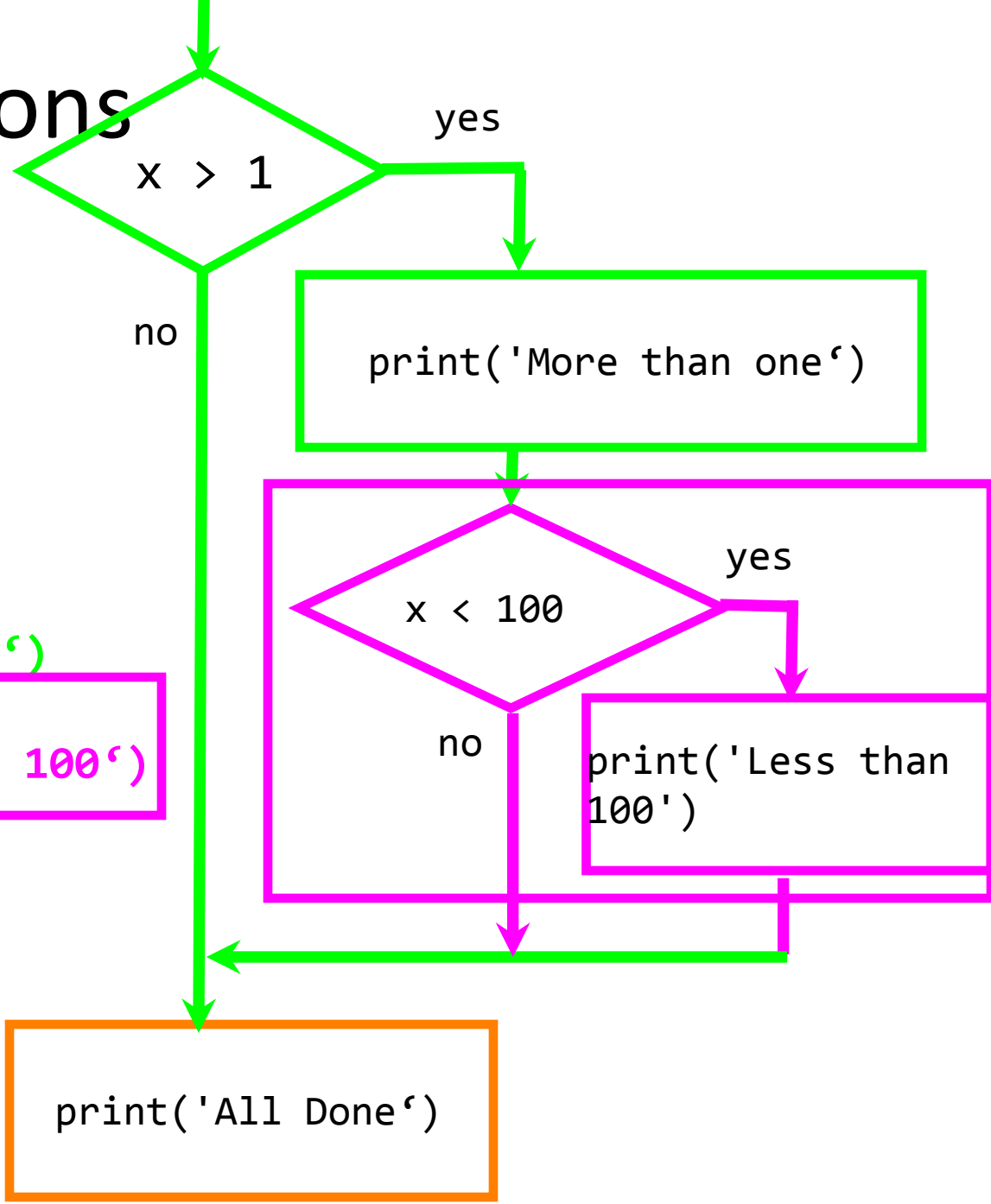
Nested Decisions

```
x = 42

if x > 1 :
    print('More than one‘)
    if x < 100 :
        print('Less than 100‘)

print('All done‘)
```



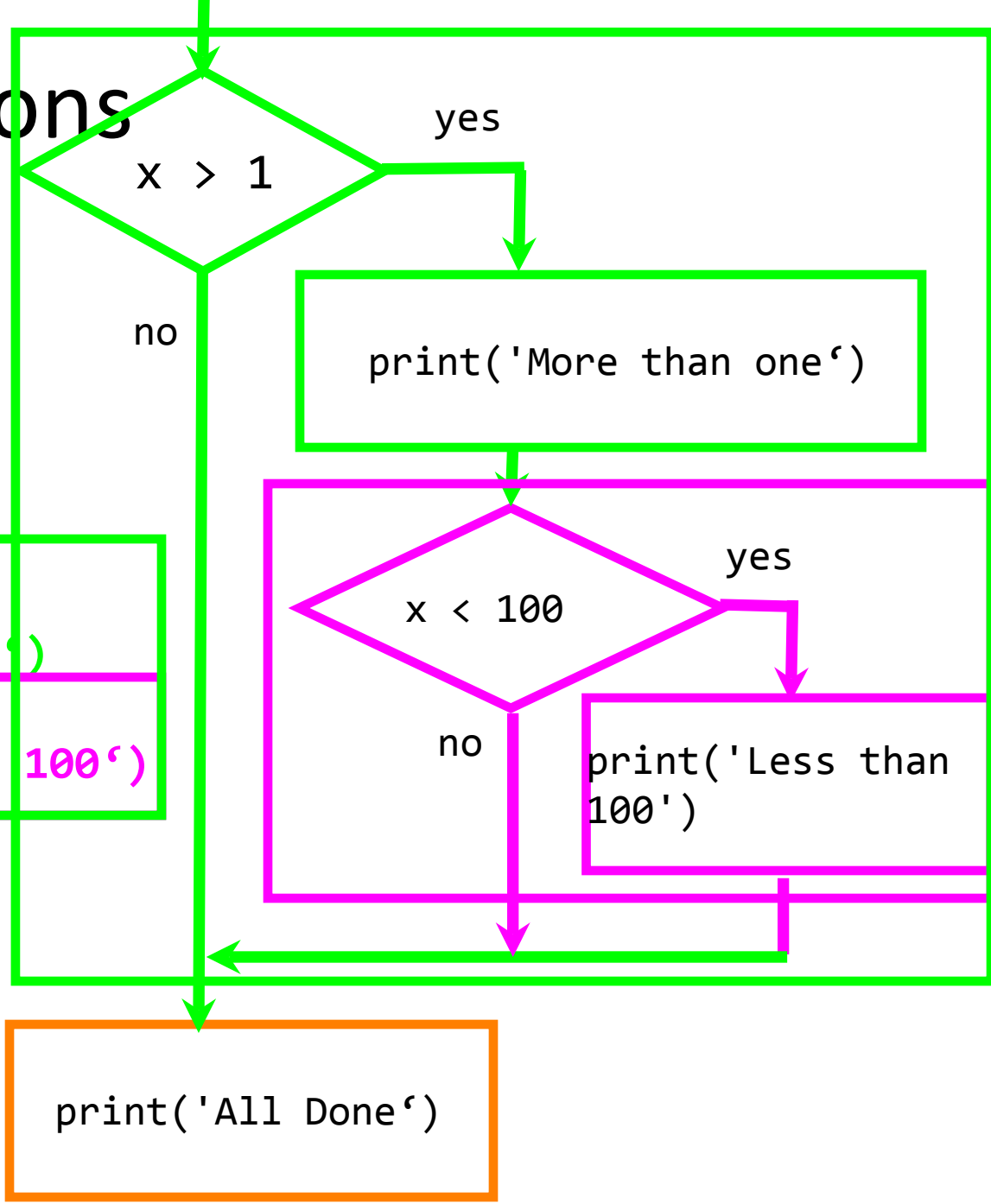
Nested Decisions

```
x = 42

if x > 1 :
    print('More than one‘)
    if x < 100 :
        print('Less than 100‘)

print('All done‘)
```

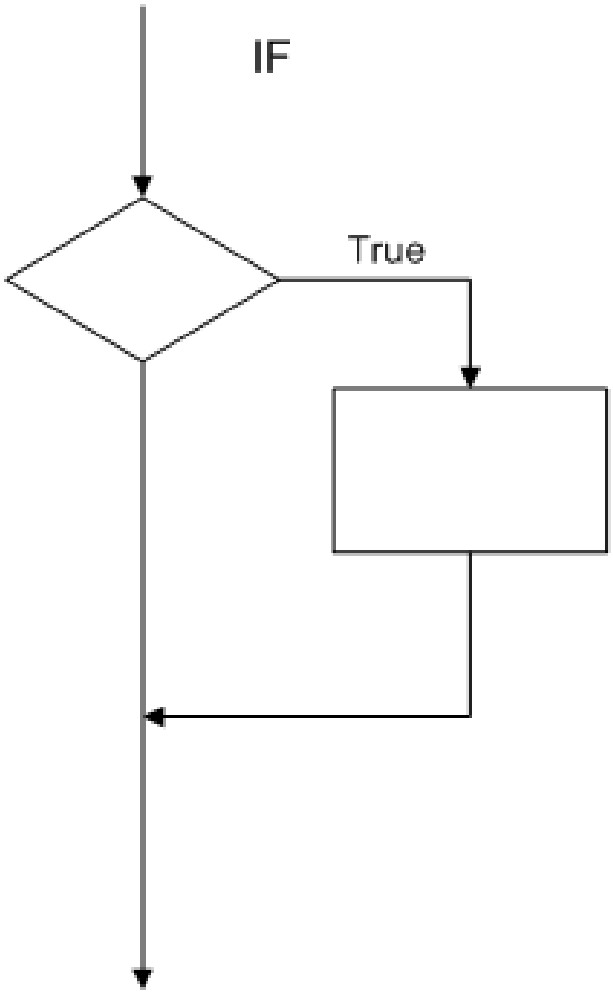


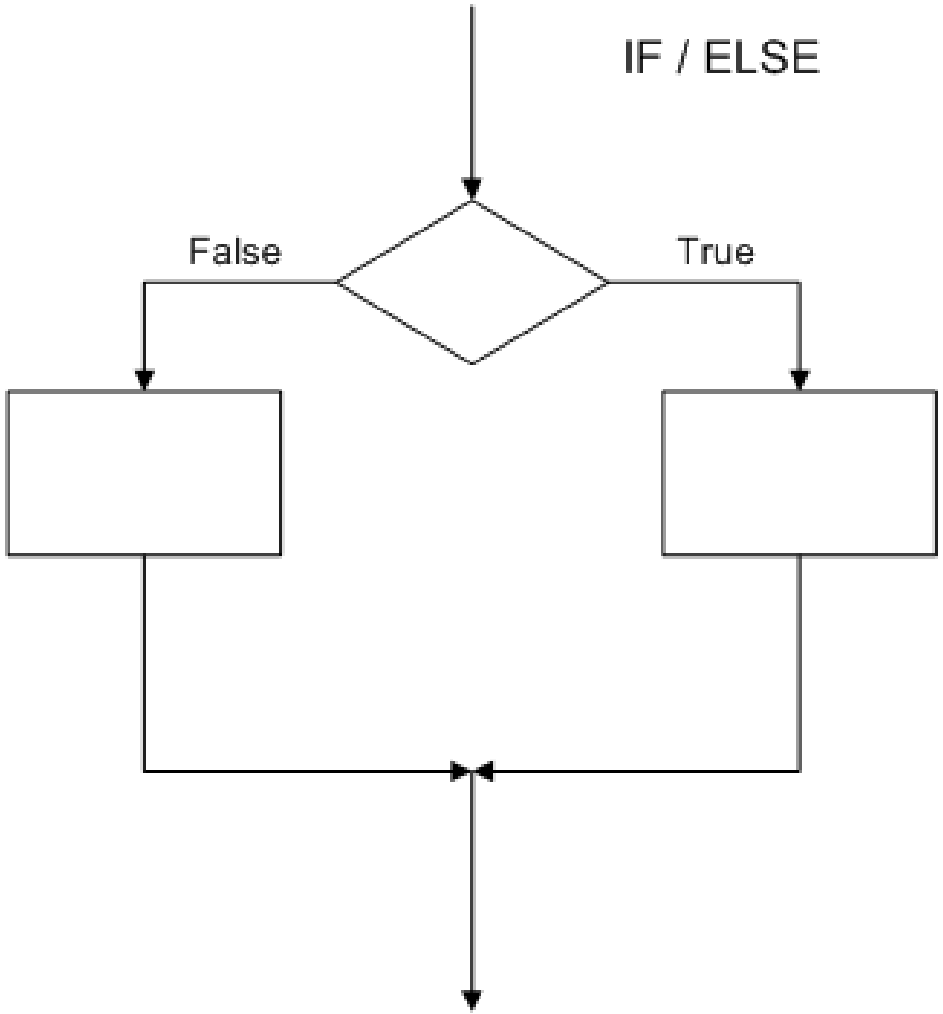
Nested Decisions

```
x = 42

if x > 1 :
    print('More than one')
    if x < 100 :
        print('Less than 100')

print('All done')
```



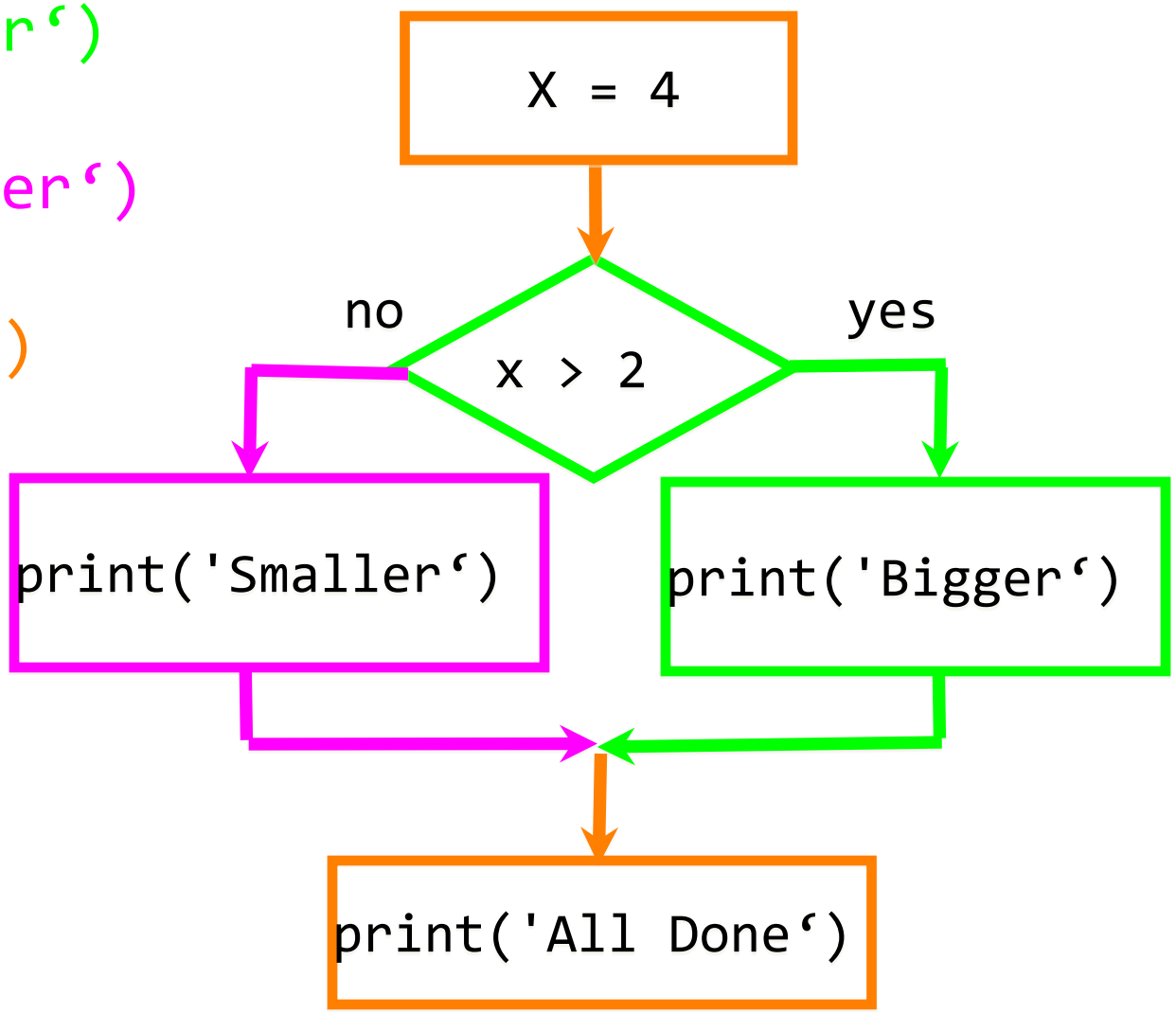
if and if/else diagrams





if/else statements

```
x = 4
if x > 2 :
    print('Bigger‘)
else :
    print('Smaller‘)

print('All done‘)
```



if/else

```
if <condition>:
    <indented code block 1>
else:
    <indented code block 2>
<non-indented statement>
```

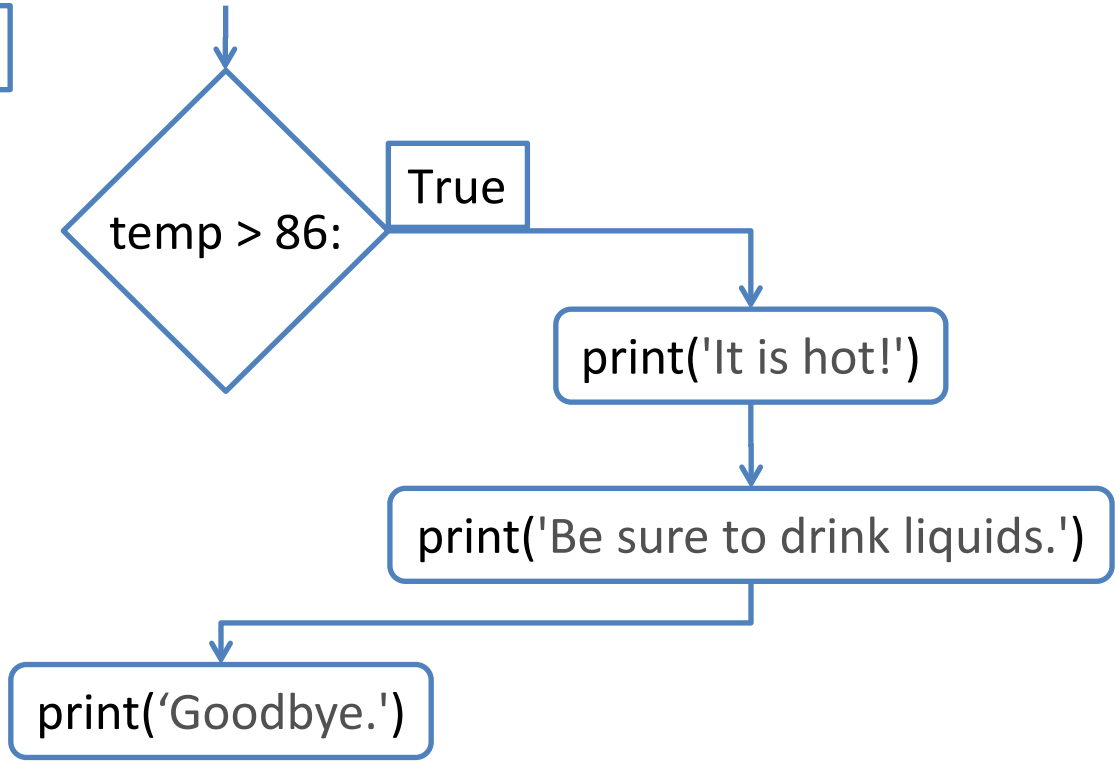
```
if temp > 86:
    print('It is hot!')
    print('Be sure to drink liquids.')
else:
    print('It is not hot.')
    print('Bring a jacket.')
print('Goodbye.')
```

if/else

```
if <condition>:
    <indented code block 1>
else:
    <indented code block 2>
<non-indented statement>
```

```
if temp > 86:
    print('It is hot!')
    print('Be sure to drink liquids.')
else:
    print('It is not hot.')
    print('Bring a jacket.')
print('Goodbye.')
```

The value of temp is 90.



if/else

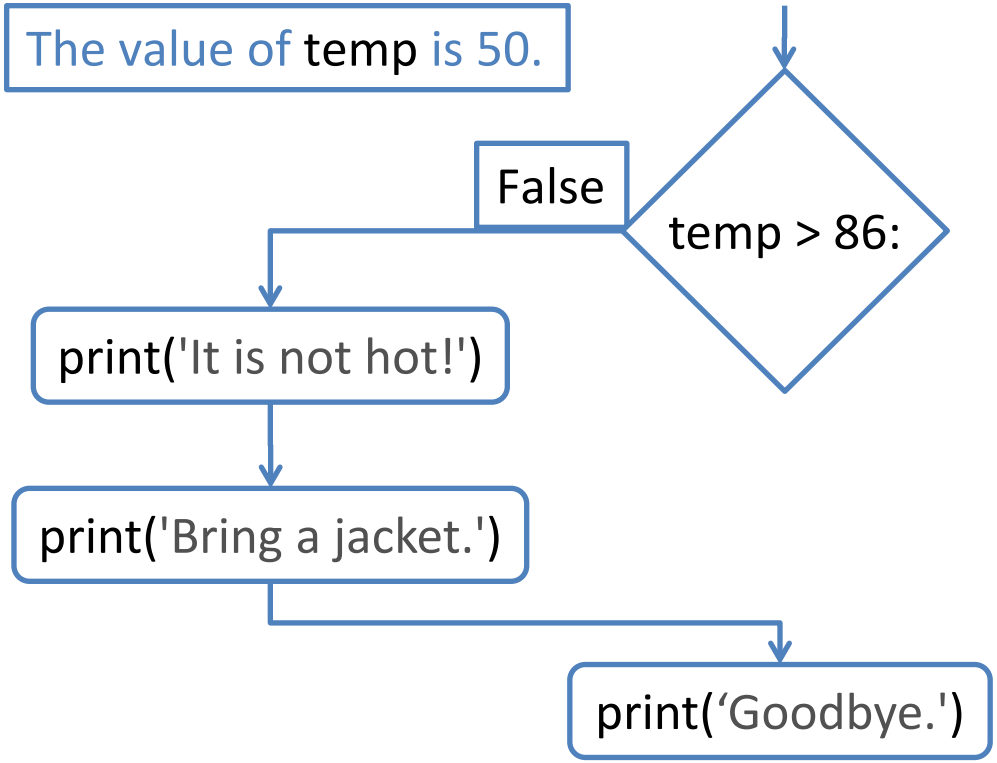
```
if <condition>:
    <indented code block 1>
else:
    <indented code block 2>
<non-indented statement>
```

```
if temp > 86:
    print('It is hot!')
    print('Be sure to drink liquids.')
else:
    print('It is not hot.')
    print('Bring a jacket.')
print('Goodbye.')
```

if/else

```
if <condition>:
    <indented code block 1>
else:
    <indented code block 2>
<non-indented statement>
```

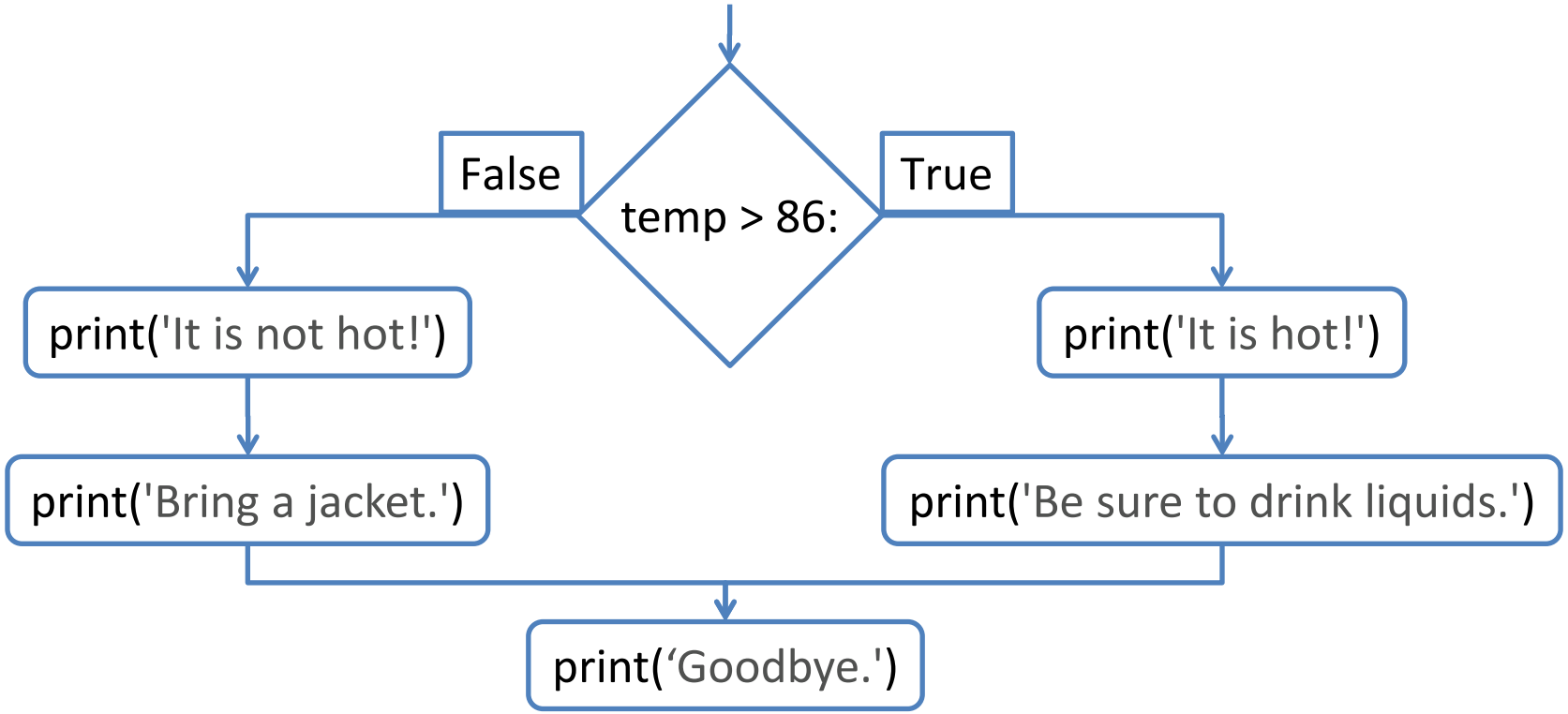
```
if temp > 86:
    print('It is hot!')
    print('Be sure to drink liquids.')
else:
    print('It is not hot.')
    print('Bring a jacket.')
print('Goodbye.')
```



if/else

```
if <condition>:
    <indented code block 1>
else:
    <indented code block 2>
<non-indented statement>
```

```
if temp > 86:
    print('It is hot!')
    print('Be sure to drink liquids.')
else:
    print('It is not hot.')
    print('Bring a jacket.')
print('Goodbye.')
```



Exercise 5/Cont

```
Enter your age: 22
you can vote!
```

Գրեք ծրագիր որը

- Հարցնում է user-ի տարիքը
- Եթե (IF) տարիքը 20-ից > է և տպում “You can vote”
- Հակառակ դեպքում (Else) տպում “You can’t vote!”

```
Enter your age: 18
you can’t vote!
```

ex5c.py

Exercise 5/Cont

Գրեք ծրագիր որը

- Հարցնում է user-ի տարիքը
- Եթե (IF) տարիքը 20-ից > է և տպում “You can vote”
- Հակառակ դեպքում (Else) տպում “You can’t vote!”

```
Enter your age: 22
you can vote!
```

```
age = int(input('Enter your age: '))
if age>20:
    print('you can vote')
else:
    print('you can\'t vote')
```

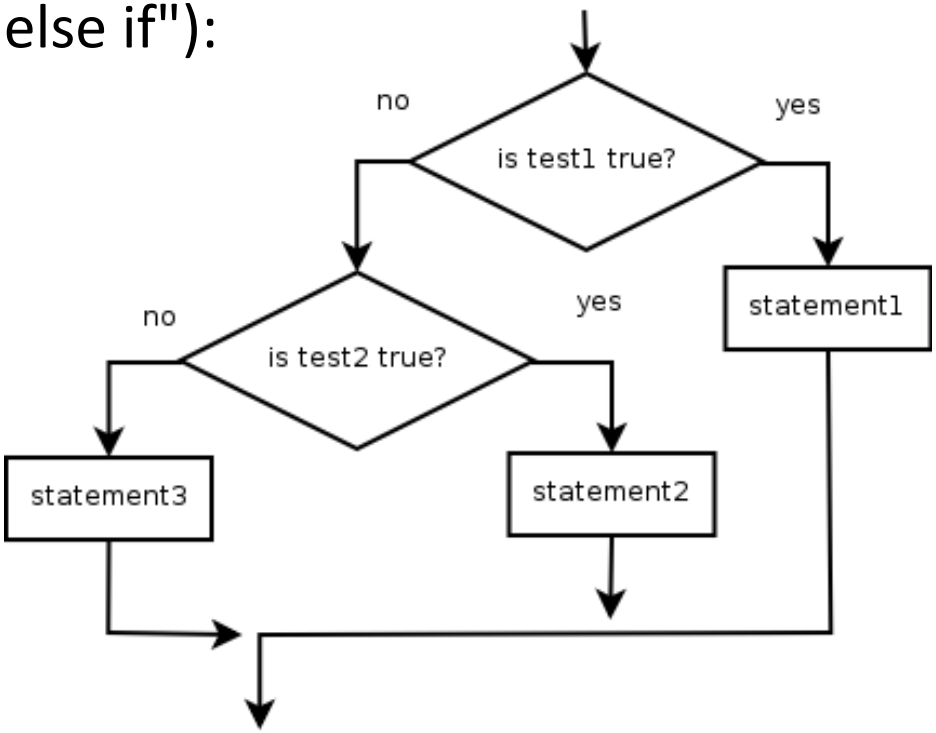
ex5c.py

```
Enter your age: 18
you can’t vote!
```

elif (else if)

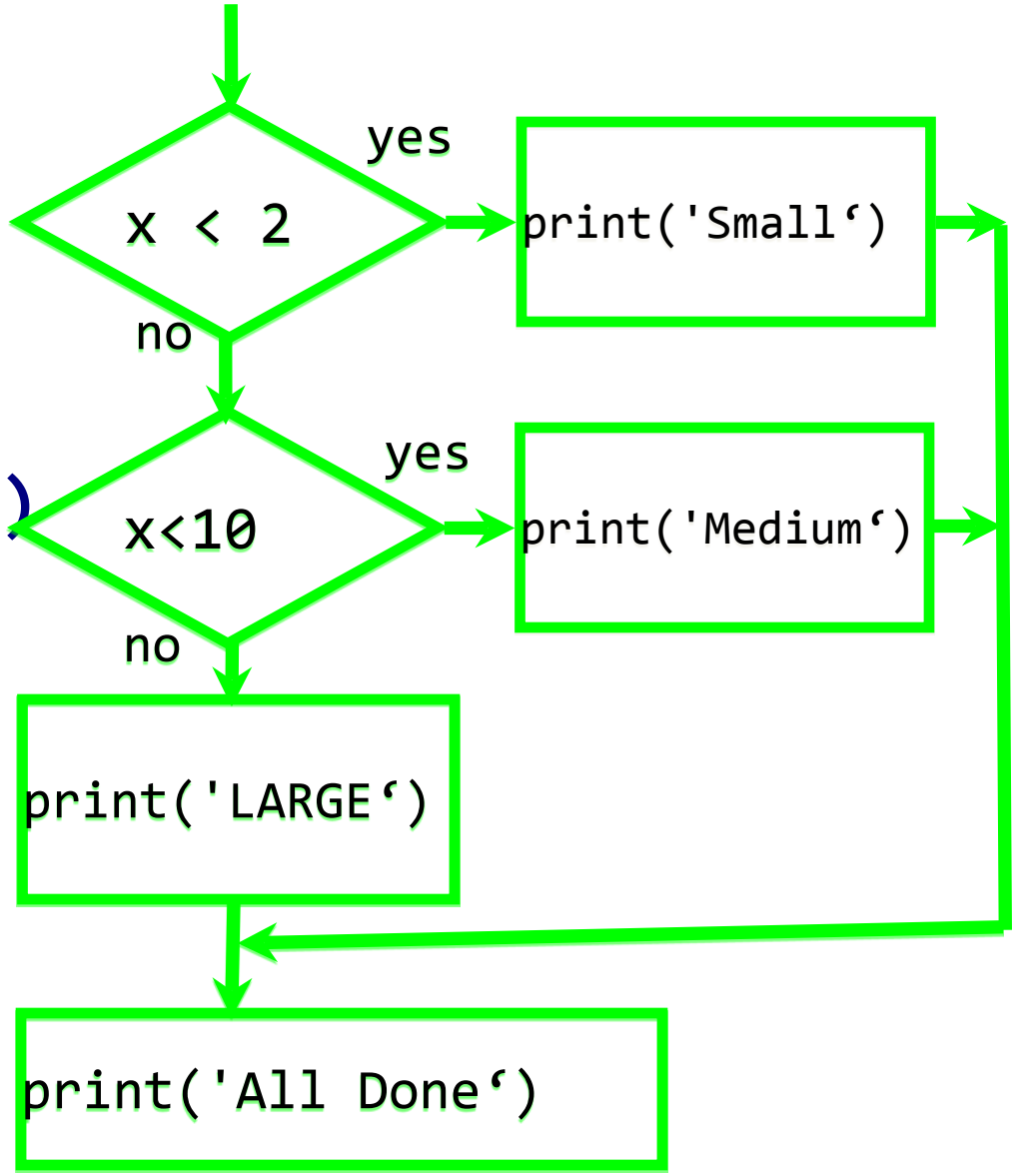
Multiple conditions with `elif` ("else if"):

```
if condition:
    statements
elif condition:
    statements
else:
    statements
```



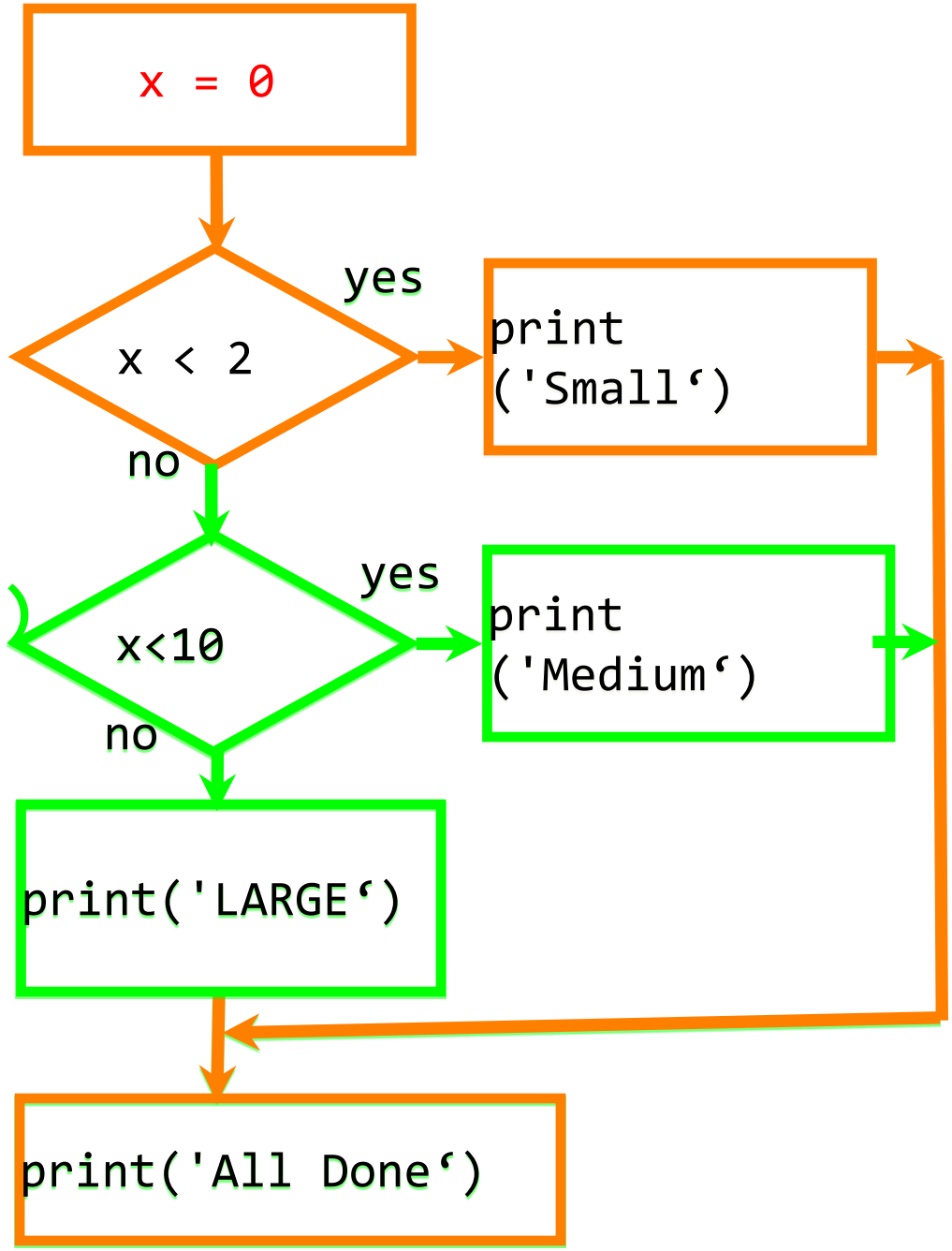
elif (else if)

```
if x < 2 :
    print('Small')
elif x < 10 :
    print('Medium')
else :
    print('LARGE')
print('All done')
```



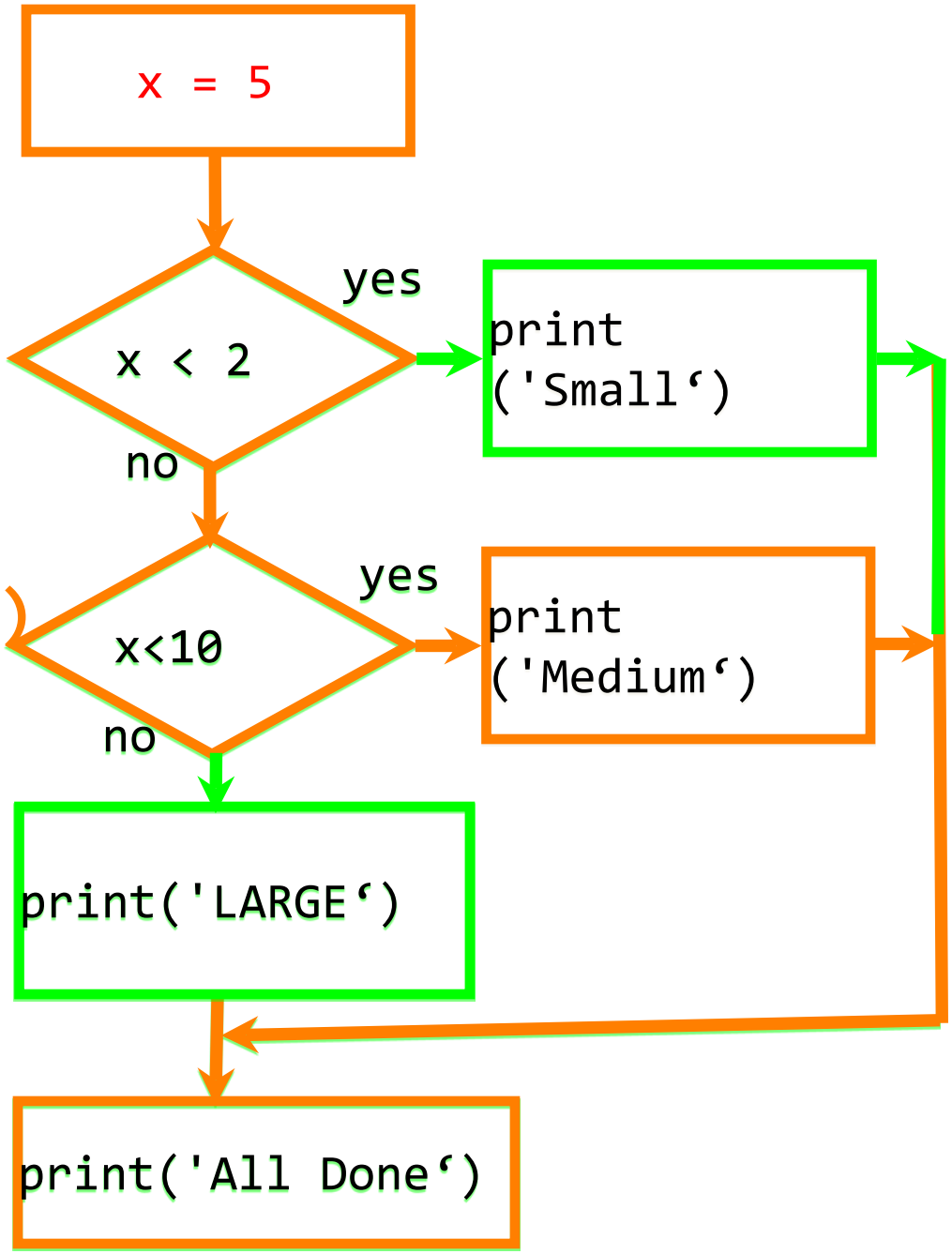
elif (else if)

```
x = 0
if x < 2 :
    print('Small‘)
elif x < 10 :
    print('Medium‘)
else :
    print('LARGE‘)
print('All done’)
```



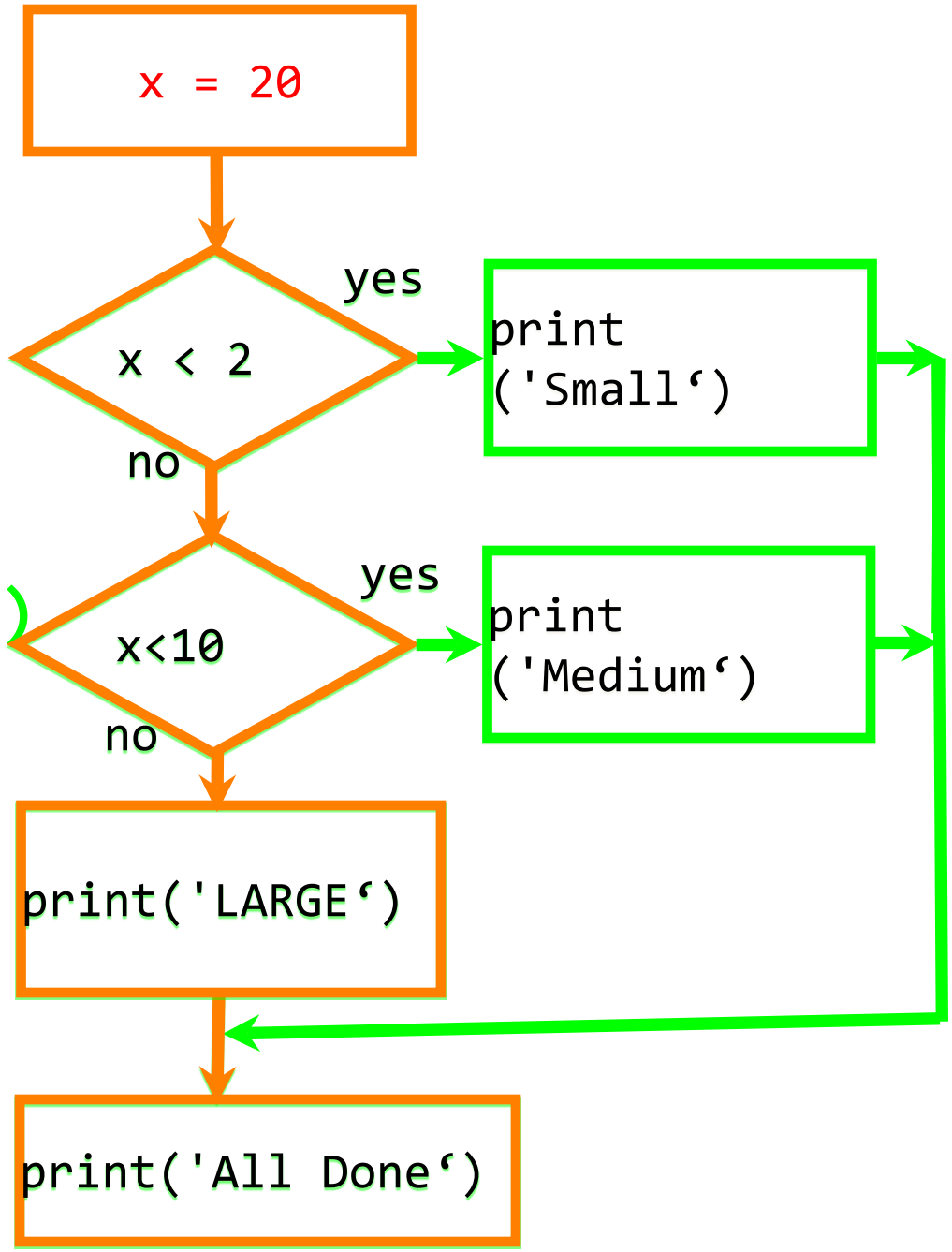
elif (else if)

```
x = 5
if x < 2 :
    print('Small‘)
elif x < 10 :
    print('Medium‘)
else :
    print('LARGE‘)
print 'All done'
```



elif (else if)

```
x = 20
if x < 2 :
    print('Small‘)
elif x < 10 :
    print('Medium‘)
else :
    print('LARGE‘)
print('All done‘)
```



Exercise 6

Գրեք ծրագիր որը

- Հարցնում է user-ից թիվ (number)
- Եթե number > 0 տպի “positive”
- Եթե number < 0 տպի ”negative”
- Հակառակ դեպքում տպի “zero”

```
>>>
Enter number: 5
positive
>>>
Enter number: -50
negative
>>>
Enter number: 0
zero
```

Exercise 6

Գրեք ծրագիր որը

- Հարցնում է user-ից թիվ (number)
- Եթե number > 0 տպի “positive”
- Եթե number < 0 տպի ”negative”
- Հակառակ դեպքում տպի “zero”

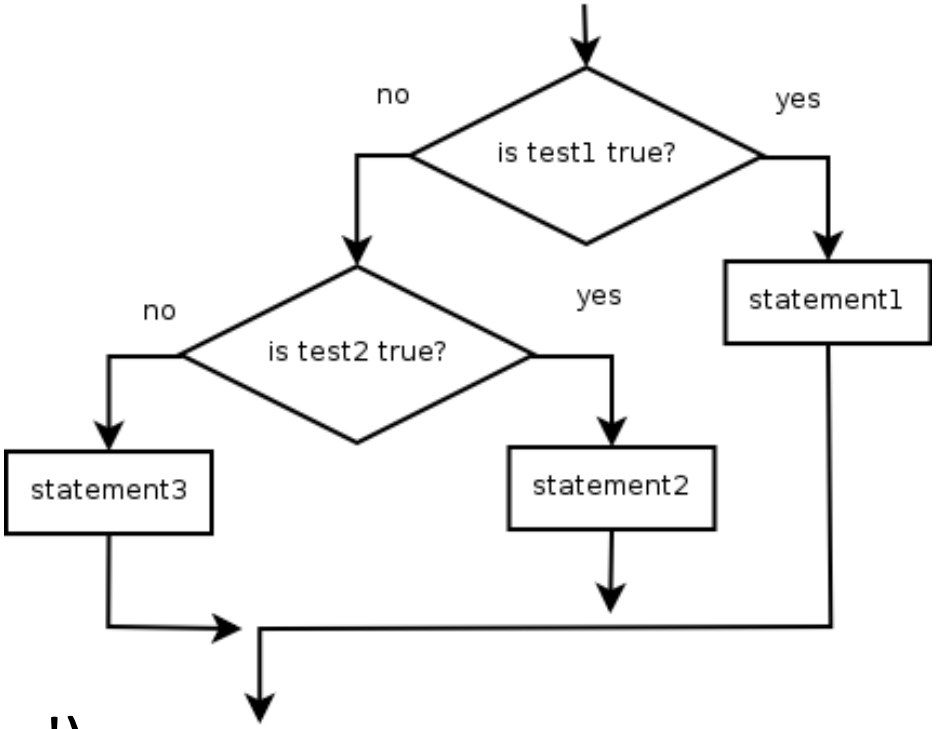
```
n = input("Enter number: ")
number = float(n)
if number > 0:
    print(“positive")
elif number < 0:
    print(“negative")
else:
    print(“zero")
```

ex6.py

```
>>>
Enter number: 5
positive
>>>
Enter number: -50
negative
>>>
Enter number: 0
zero
```

Nested indentation

```
if      x > 0.0:
        print ('Positive')
else :
        if      x < 0.0 :
                print ('Negative')
        else      :
                print ('Zero')
```



Exercise 7

Փոխեք ex6.py այնպես որ

- օգտագործեք nested indentation
- և միայն if/else

```
n = input("Enter number: ")
number = float(n)
if number > 0:
    print(“positive")
elif number < 0:
    print(“negative")
else:
    print(“zero")
```

ex6.py

Exercise 7

Փոխեք ex6.py այնպես որ

- օգտագործեք nested indentation
- և միայն if/else

```
n = input("Enter number: ")
number = float(n)
if number > 0:
    print(“positive")
elif number < 0:
    print(“negative")
else:
    print(“zero")
```

ex6.py

```
n = input("Enter number: ")
number = float(n)
if number > 0:
    print(“positive")
else:
    if number < 0:
        print(“negative")
    else:
        print(“zero")
```

ex7.py

Ցիկլեր : Loops:
for and **while**

Ցիկլեր : Loops: **for** and **while**

```
>>> print("hello")
hello
>>> print("hello")
hello
>>> print("hello")
hello
>>> print("hello")
hello
>>> print("hello")
hello
```

Ցիկլեր : Loops: **for** and **while**

```
>>> print("hello")
hello
>>> print("hello")
hello
>>> print("hello")
hello
>>> print("hello")
hello
>>> print("hello")
hello
```

```
# FOR
>>> for x in range(0, 5):
        print('hello')
hello
hello
hello
hello
hello
```

Ցիկլեր : Loops: for and while

```
>>> print("hello")
hello
>>> print("hello")
hello
>>> print("hello")
hello
>>> print("hello")
hello
>>> print("hello")
hello
```

```
# FOR
>>> for x in range(0, 5):
        print('hello')
hello
hello
hello
hello
hello
```

```
# WHILE
x = 0
while x < 5:
    print('hello')
    x = x + 1
```

for loop

Հաջորդականության ամեն անդամի համար կատարում է բլոկի գործողությունները

- Եթե հաջորդականությունը(sequence) string է
  => item-ները տառեր են
- Եթե sequence-ը list է
  => item-ները list-ի անդամներն են

name =

'	A	p	p	l	e	'

```
>>> name = 'Apple'
>>> for char in name:
        print(char)
```

for loop

Հաջորդականության ամեն անդամի համար կատարում է բլոկի գործողությունները

- Եթե հաջորդականությունը(sequence) string է
  => item-ները տառեր են
- Եթե sequence-ը list է
  => item-ները list-ի անդամներն են

name = ' A p p l e '

char = 'A'

```
>>> name = 'Apple'
>>> for char in name:
        print(char)
```

for loop

Հաջորդականության ամեն անդամի համար կատարում է բլոկի գործողությունները

- Եթե հաջորդականությունը(sequence) string է
  => item-ները տառեր են
- Եթե sequence-ը list է
  => item-ները list-ի անդամներն են

name = ' A p p l e '

```
>>> name = 'Apple'
>>> for char in name:
        print(char)


A
```

for loop

Հաջորդականության ամեն անդամի համար կատարում է բլոկի գործողությունները

- Եթե հաջորդականությունը(sequence) string է
  => item-ները տառեր են
- Եթե sequence-ը list է
  => item-ները list-ի անդամներն են

name = ' A p p l e '

char = 'p'

```
>>> name = 'Apple'
>>> for char in name:
        print(char)
```

```
A
```

for loop

Հաջորդականության ամեն անդամի համար կատարում է բլոկի գործողությունները

- Եթե հաջորդականությունը(sequence) string է
  => item-ները տառեր են
- Եթե sequence-ը list է
  => item-ները list-ի անդամներն են

name =

'	A	p	p	l	e	'

```
>>> name = 'Apple'
>>> for char in name:
        print(char)


A
p
```

for loop

Հաջորդականության ամեն անդամի համար կատարում է բլոկի գործողությունները

- Եթե հաջորդականությունը(sequence) string է
  => item-ները տառեր են
- Եթե sequence-ը list է
  => item-ները list-ի անդամներն են

name = ' A p p l e '

char = 'p'

```
>>> name = 'Apple'
>>> for char in name:
        print(char)


A
p
```

for loop

Հաջորդականության ամեն անդամի համար կատարում է բլոկի գործողությունները

- Եթե հաջորդականությունը(sequence) string է
  => item-ները տառեր են
- Եթե sequence-ը list է
  => item-ները list-ի անդամներն են

name =

'	A	p	p	l	e	'

```
>>> name = 'Apple'
>>> for char in name:
        print(char)


A
p
p
```

for loop

Հաջորդականության ամեն անդամի համար կատարում է բլոկի գործողությունները

- Եթե հաջորդականությունը(sequence) string է
  => item-ները տառեր են
- Եթե sequence-ը list է
  => item-ները list-ի անդամներն են

name = ' A p p l e '

char = 'l'

```
>>> name = 'Apple'
>>> for char in name:
        print(char)


A
p
p
```

for loop

Հաջորդականության ամեն անդամի համար կատարում է բլոկի գործողությունները

- Եթե հաջորդականությունը(sequence) string է
  => item-ները տառեր են
- Եթե sequence-ը list է
  => item-ները list-ի անդամներն են

name = ' A p p l e '

```
>>> name = 'Apple'
>>> for char in name:
        print(char)


A
p
p
l
```

for loop

Հաջորդականության ամեն անդամի համար կատարում է բլոկի գործողությունները

- Եթե հաջորդականությունը(sequence) string է
  => item-ները տառեր են
- Եթե sequence-ը list է
  => item-ները list-ի անդամներն են

name = ' A p p l e '

char = 'e'

```
>>> name = 'Apple'
>>> for char in name:
        print(char)


A
p
p
l
```

for loop

Հաջորդականության ամեն անդամի համար կատարում է բլոկի գործողությունները

- Եթե հաջորդականությունը(sequence) string է
  => item-ները տառեր են
- Եթե sequence-ը list է
  => item-ները list-ի անդամներն են

name = ' A p p l e '

```
>>> name = 'Apple'
>>> for char in name:
        print(char)


A
p
p
l
e
```

for loop

```
for <variable> in <sequence>:
    <indented code block >
<non-indented code block>
```

```
>>>
```

for loop

```
for <variable> in <sequence>:
    <indented code block >
<non-indented code block>
```

```
for word in ['stop', 'desktop', 'post', 'top']:
    if 'top' in word:
        print(word)
print('Done.')
```

```
>>>
```

for loop

```
for <variable> in <sequence>:
    <indented code block >
<non-indented code block>
```

```
for word in ['stop', 'desktop', 'post', 'top']:
    if 'top' in word:
        print(word)
print('Done.')
```

word = 'stop'

```
>>>
```

for loop

```
for <variable> in <sequence>:
    <indented code block >
<non-indented code block>
```

```
for word in ['stop', 'desktop', 'post', 'top']:
    if 'top' in word:
        print(word)
print('Done.')
```

```
>>>
stop
```

for loop

```
for <variable> in <sequence>:
    <indented code block >
<non-indented code block>
```

```
for word in ['stop', 'desktop', 'post', 'top']:
    if 'top' in word:
        print(word)
print('Done.')
```

word = 'desktop'

```
>>>
stop
```

for loop

```
for <variable> in <sequence>:
    <indented code block >
<non-indented code block>
```

```
for word in ['stop', 'desktop', 'post', 'top']:
    if 'top' in word:
        print(word)
print('Done.')
```

```
>>>
stop
desktop
```

for loop

```
for <variable> in <sequence>:
    <indented code block >
<non-indented code block>
```

```
for word in ['stop', 'desktop', 'post', 'top']:
    if 'top' in word:
        print(word)
print('Done.')
```

word = 'post'

```
>>>
stop
desktop
```

for loop

```
for <variable> in <sequence>:
    <indented code block >
<non-indented code block>
```

```
for word in ['stop', 'desktop', 'post', 'top']:
    if 'top' in word:
        print(word)
print('Done.')
```

word = 'post'

```
>>>
stop
desktop
```

for loop

```
for <variable> in <sequence>:
    <indented code block >
<non-indented code block>
```

```
for word in ['stop', 'desktop', 'post', 'top']:
    if 'top' in word:
        print(word)
print('Done.')
```

```
>>>
stop
desktop
```

word = 'top'

for loop

```
for <variable> in <sequence>:
    <indented code block >
<non-indented code block>
```

```
for word in ['stop', 'desktop', 'post', 'top']:
    if 'top' in word:
        print(word)
print('Done.')
```

```
>>>
stop
desktop
top
```

for loop

```
for <variable> in <sequence>:
    <indented code block >
<non-indented code block>
```

```
for word in ['stop', 'desktop', 'post', 'top']:
    if 'top' in word:
        print(word)
print('Done.')
```

```
>>>
stop
desktop
top
Done.
```

Function range():սիրույթ

Function range()

- is used to iterate over a sequence of numbers in a specified range

Function range():տիրույթ

Function range()
- is used to iterate over a sequence of numbers in a specified range

- To iterate(կրկնել) over the n numbers 0, 1, 2, …, n-1
  for i in range(n):

Function range():տիրույթ

Function range()
- is used to iterate over a sequence of numbers in a specified range

- To iterate(կրկնել) over the n numbers 0, 1, 2, …, n-1
  for i in range(n):

```
>>> for i in range(4):
        print(i)


0
1
2
3
```

Function range():տիրույթ

Function range()
- is used to iterate over a sequence of numbers in a specified range

- To iterate(կրկնել) over the n numbers 0, 1, 2, …, n-1
  for i in range(n):

```
>>> for i in range(1):
        print(i)


0
>>>
```

Function range():տիրույթ

Function range()
- is used to iterate over a sequence of numbers in a specified range

- To iterate(կրկնել) over the n numbers 0, 1, 2, …, n-1
  for i in range(n):

```
>>> for i in range(0):
        print(i)

>>>
```

Function range():տիրույթ

Function range()
- is used to iterate over a sequence of numbers in a specified range

- To iterate(կրկնել) over the n numbers 0, 1, 2, …, n-1
  for i in range(n):

Function range():տիրույթ

Function range()
- is used to iterate over a sequence of numbers in a specified range

- To iterate(կրկնել) over the n numbers 0, 1, 2, …, n-1
  for i in range(n):
- To iterate over the n numbers i, i+1, i+2, …, n-1
  for i in range(i, n):

Function range():տիրույթ

Function range()
- is used to iterate over a sequence of numbers in a specified range

- To iterate(կրկնել) over the n numbers 0, 1, 2, …, n-1
  for i in range(n):
- To iterate over the n numbers i, i+1, i+2, …, n-1
  for i in range(i, n):

```
>>> for i in range(2, 6):
        print(i)



2
3
4
5
```

Function range():տիրույթ

Function range()
- is used to iterate over a sequence of numbers in a specified range

- To iterate(կրկնել) over the n numbers 0, 1, 2, …, n-1
  for i in range(n):
- To iterate over the n numbers i, i+1, i+2, …, n-1
  for i in range(i, n):

```
>>> for i in range(2, 3):
        print(i)


2
>>>
```

Function range():տիրույթ

Function range()
- is used to iterate over a sequence of numbers in a specified range

- To iterate(կրկնել) over the n numbers 0, 1, 2, …, n-1
  for i in range(n):
- To iterate over the n numbers i, i+1, i+2, …, n-1
  for i in range(i, n):

```
>>> for i in range(2, 2):
        print(i)


>>>
```

Function range():տիրույթ

Function range()
- is used to iterate over a sequence of numbers in a specified range

- To iterate(կրկնել) over the n numbers 0, 1, 2, …, n-1
  for i in range(n):
- To iterate over the n numbers i, i+1, i+2, …, n-1
  for i in range(i, n):

Function range():տիրույթ

Function range()
- is used to iterate over a sequence of numbers in a specified range

- To iterate(կրկնել) over the n numbers 0, 1, 2, …, n-1
  for i in range(n):
- To iterate over the n numbers i, i+1, i+2, …, n-1
  for i in range(i, n):
- To iterate over the n numbers i, i+c, i+2c, i+3c, …, n-1
  for i in range(i, n, c):

Function range():տիրույթ

Function range()
- is used to iterate over a sequence of numbers in a specified range

- To iterate(կրկնել) over the n numbers 0, 1, 2, …, n-1
  for i in range(n):
- To iterate over the n numbers i, i+1, i+2, …, n-1
  for i in range(i, n):
- To iterate over the n numbers i, i+c, i+2c, i+3c, …, n-1
  for i in range(i, n, c):

```
>>> for i in range(2, 16, 4):
        print(i)


2
6
10
14
```

Function range():տիրույթ

Function range()
- is used to iterate over a sequence of numbers in a specified range

- To iterate(կրկնել) over the n numbers 0, 1, 2, …, n-1
  for i in range(n):
- To iterate over the n numbers i, i+1, i+2, …, n-1
  for i in range(i, n):
- To iterate over the n numbers i, i+c, i+2c, i+3c, …, n-1
  for i in range(i, n, c):

```
>>> for i in range(0, 16, 4):
        print(i)



0
4
8
12
```

Function range():տիրույթ

Function range()
- is used to iterate over a sequence of numbers in a specified range

- To iterate(կրկնել) over the n numbers 0, 1, 2, …, n-1
  for i in range(n):
- To iterate over the n numbers i, i+1, i+2, …, n-1
  for i in range(i, n):
- To iterate over the n numbers i, i+c, i+2c, i+3c, …, n-1
  for i in range(i, n, c):

```
>>> for i in range(2, 16, 10):
        print(i)


2
12
>>>
```

Function range():տիրույթ

Function range()
- is used to iterate over a sequence of numbers in a specified range

- To iterate(կրկնել) over the n numbers 0, 1, 2, …, n-1
  for i in range(n):
- To iterate over the n numbers i, i+1, i+2, …, n-1
  for i in range(i, n):
- To iterate over the n numbers i, i+c, i+2c, i+3c, …, n-1
  for i in range(i, n, c):

```
range(start, stop, step)
```

```
>>> for i in range(2, 16, 10):
        print(i)

2
12
>>>
```

Exercise 7

Գրեք for loop-եր որոնք կտպեն հետևյալ sequence-ները:

a) 0, 1, 2, 3, 4, 5, 6, 7, 8 , 9, 10
b) 1, 2, 3, 4, 5, 6, 7, 8, 9
c) 0, 2, 4, 6, 8
d) 1, 3, 5, 7, 9
e) 20, 30, 40, 50, 60

Exercise 7

Գրեք for loop-եր որոնք կտպեն հետևյալ sequence-ները:

a) 0, 1, 2, 3, 4, 5, 6, 7, 8 , 9, 10

b) 1, 2, 3, 4, 5, 6, 7, 8, 9

c) 0, 2, 4, 6, 8

d) 1, 3, 5, 7, 9

e) 20, 30, 40, 50, 60

```
for i in range(11):
    print(i)
for i in range(1,10):
    print(i)
for i in range(0,9,2):
    print(i)
for i in range(1,10,2):
    print(i)
for i in range(20,61,10):
    print(i)
```

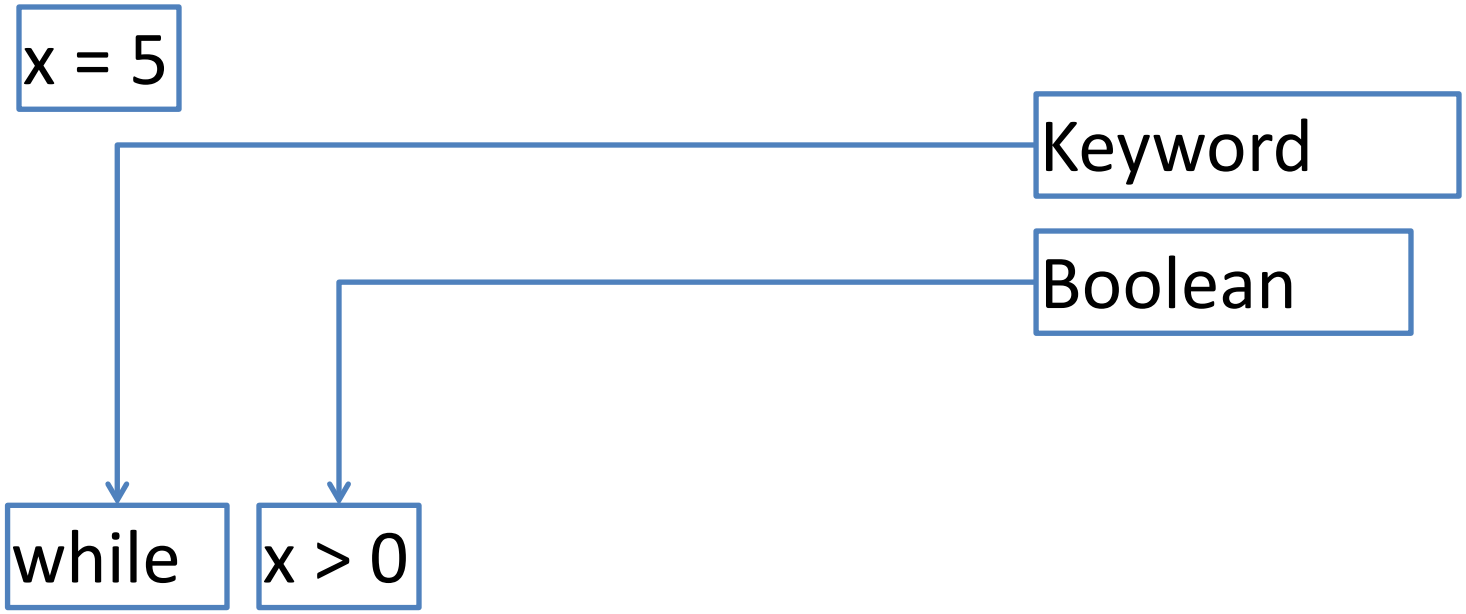
ex7.py

while Loop

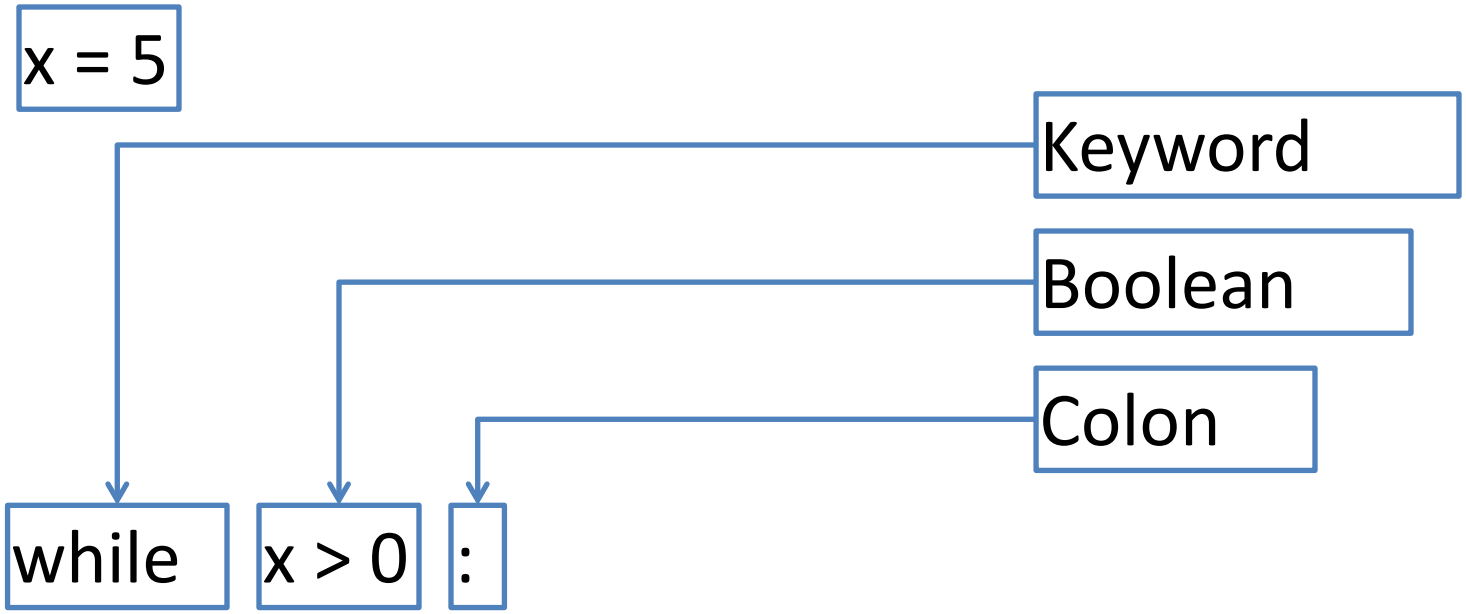
x = 5

while Loop

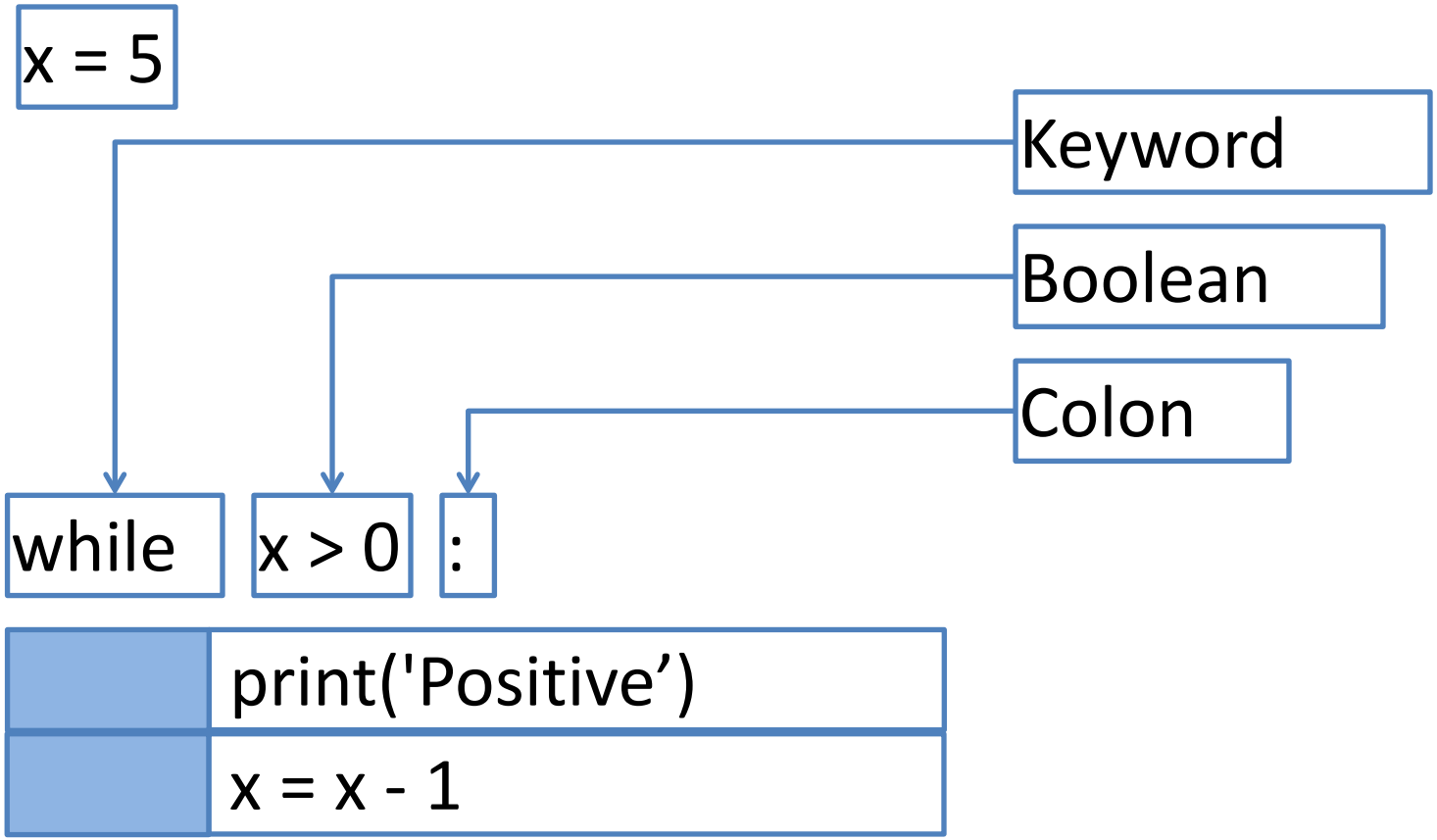
while Loop



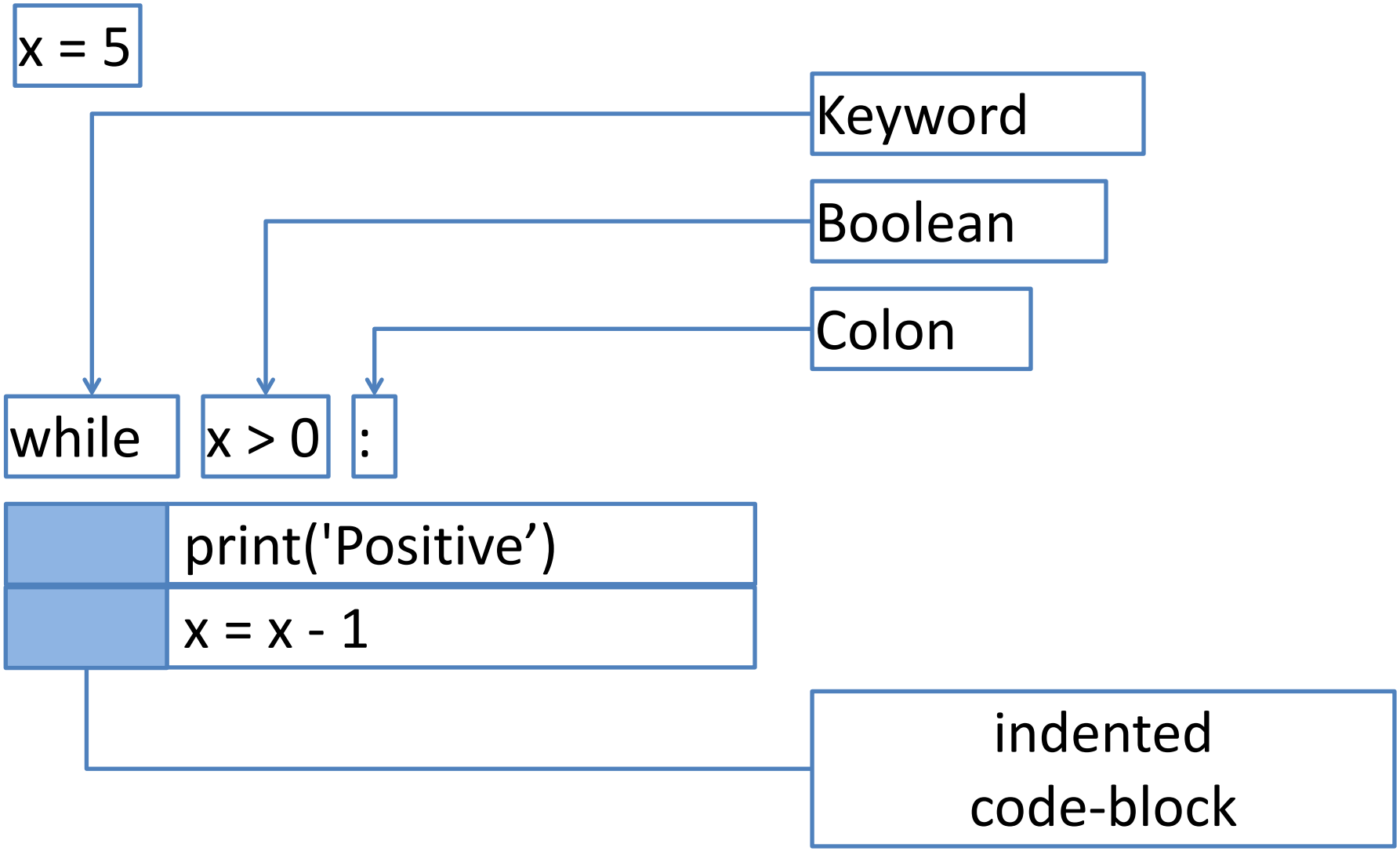
while Loop



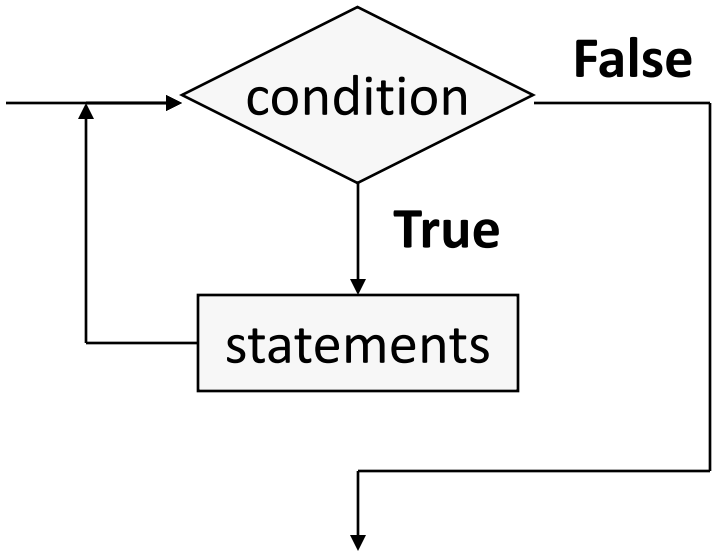
while Loop



while Loop

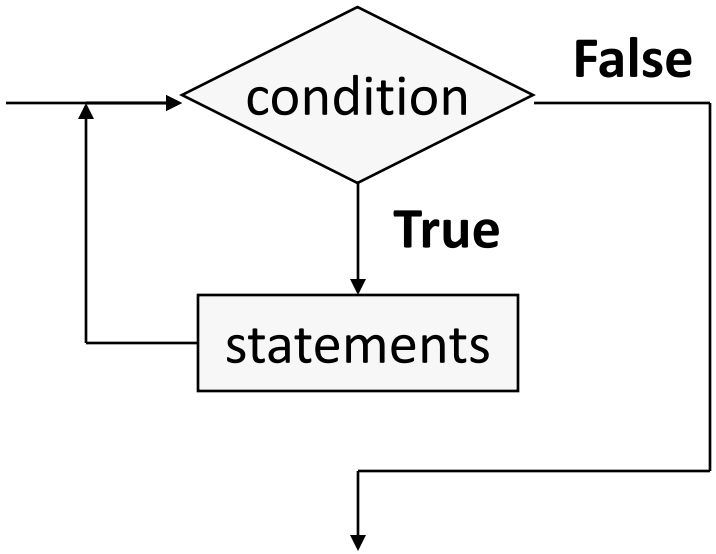

x = 5
Keyword
Boolean
Colon
while
x > 0
:
print('Positive’)
x = x - 1
indented
code-block

while Loop




```
while <condition>:
    <indented code-block >
```

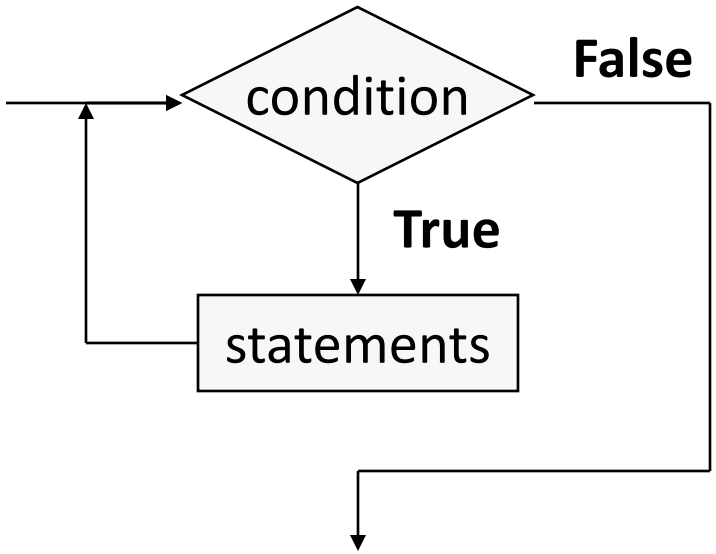
while Loop




```
while <condition>:
    <indented code-block >
```

```
count=0
while count<=10:
    print (count)
    count=count+1
```

while Loop

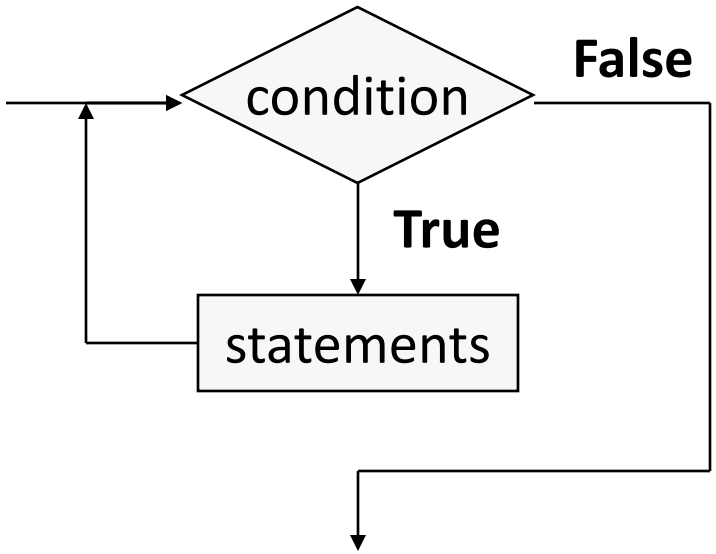



```
while <condition>:
    <indented code-block >
```

```
count=0
while count<=10:
    print (count)
    count=count+1
```

```
x='Python'
while x:
    print(x)
    x=x[1:]
```

while Loop




```
while <condition>:
    <indented code-block >
```

```
count=0
while count<=10:
    print (count)
    count=count+1
```

```
x='Python'
while x:
    print(x)
    x=x[1:]
```

```
while 1:
    print("Type Ctrl-C to stop me!!")

while True:
    print("Type Ctrl-C to stop me!!")
```

Որ loop-ը օգտագործել?

Որ loop-ը օգտագործել?

- A **`for`** loop works well
  - when the number of iterations is predictable
    (Երբ իտերացիաների թիվը հայտնի է)

Որ loop-ը օգտագործել?

- A **`for`** loop works well
  - when the number of iterations is predictable
    (Երբ իտերացիաների թիվը հայտնի է)
- A **`while`** loop works well
  - when the number of iterations is not predictable
    (Երբ իտերացիաների թիվը հայտնի չէ)

Որ loop-ը օգտագործել?

- A **`for`** loop works well
  - when the number of iterations is predictable
    (Երբ իտերացիաների թիվը հայտնի է)
- A **`while`** loop works well
  - when the number of iterations is not predictable
    (Երբ իտերացիաների թիվը հայտնի չէ)
- Երբ է իտերացիաների թիվը անհայտ?

Երբ է իտերացիաների թիվը անհայտ?

```
answer = "no"
while answer == "no":
        answer = input("Are we there? ")
print("We're there!")
```

Երբ է իտերացիաների թիվը անհայտ?

```
answer = "no"
while answer == "no":
        answer = input("Are we there? ")
print("We're there!")
```

```
>>>
Are we there? no
Are we there? no
Are we there? no
Are we there? no
Are we there? yes
We're there!
>>>
```

Exercise 8

- Print this using while loop

```
******
*****
****
***
**
*
```

Exercise 8

- Print this using while loop

```
******
*****
****
***
**
*
```

```
num=6
while num > 0:
    print ("*" * num)
    num = num-1
```

ex8.py

while Loop

```
epsilon = 1.0
while 1.0 + epsilon > 1.0:
      epsilon = epsilon / 2.0
      print(epsilon)

epsilon = 2.0 * epsilon
print(epsilon)
```

epsilon.py

References

1. Lambert, Kenneth. “Computer Science 111.” Accessed July 11, 2014.

2. Briggs, Jason R. *Python for Kids: A Playful Introduction to Programming*. 1 edition. San Francisco: No Starch Press, 2012.

3. Franek. “CS 1MD3 Introduction to Programming.” Accessed July 8, 2014.

Introduction to Programming
Lesson 3

Outline

- Importing Modules
- Abstraction
- User defined functions
- Parameter passing
- Assignment and mutability

Lesson Code

goo.gl/1uCTgV

importing modules

```
>>> dir(__builtins__)
```

- Built-in functions and classes

  (“int”, “len”, “sum”, “range”, “min”, “max”)

- Core python functions

Python Standard Library(PSL)

Python Standard Library(PSL)

Ավելի շատ function-ներ և class-եր կան PSL-ում

Python Standard Library(PSL)

Ավելի շատ function-ներ և class-եր կան PSL-ում

- Network programming
- Database programming
- Mathematical functions
- Pseudorandom generator
- ...

Python Standard Library(PSL)

Ավելի շատ function-ներ և class-եր կան PSL-ում

- Network programming
- Database programming
- Mathematical functions
- Pseudorandom generator
- …

The PSL-ի function-ները և class-երը դասավորված են module/package-ների մեջ

PSL module math

sqrt() սահմանված է PSL-ի math module-ում

PSL module math

```
>>> import math
>>>
```

sqrt() սահմանված է PSL-ի math module-ում

Module իմպորտ անելու համար:

import <module>

PSL module math

sqrt() սահմանված է PSL-ի math module-ում

Module իմպորտ անելու համար:

import <module>

math.sqrt()

```
>>> import math
>>> math.sqrt(4)
2.0
>>> sqrt(4)
Traceback (most recent call
last):
  File "<pyshell#10>", line
1, in <module>
    sqrt(4)
NameError: name 'sqrt' is
not defined
>>>
```

PSL module math

sqrt() սահմանված է PSL-ի math module-ում

Module իմպորտ անելու համար:

import <module>

math.sqrt()

math module-ը պարունակում է մաթ ֆունկցիաներ և հաստատուններ

```
>>> import math
>>> math.sqrt(4)
2.0
>>> sqrt(4)
Traceback (most recent call
last):
  File "<pyshell#10>", line
1, in <module>
    sqrt(4)
NameError: name 'sqrt' is
not defined
>>> help(math)
Help on module math:
…
>>> math.cos(0)
1.0
>>> math.log(8)
2.079441541679835 7
>>> math.log(8, 2)
3.0
>>> math.pi
3.141592653589793
```

PSL module math

PSL module math

```
>>> import math
>>> math.pi
3.1415926535897931
>>> math.cos(0)
1.0
>>> math.cos(math.pi)
-1.0
```

PSL module math

```
>>> import math
>>> math.pi
3.1415926535897931
>>> math.cos(0)
1.0
>>> math.cos(math.pi)
-1.0
>>> dir(math)
['__doc__', '__file__', '__name__', '__package__',
  …,'cos',
'cosh', 'degrees', 'e', 'exp',…, 'pi', 'pow',
  'radians', 'sin', 'sinh', 'sqrt', 'tan',
'tanh', 'trunc']
```

PSL module math

```
>>> import math
>>> math.pi
3.1415926535897931
>>> math.cos(0)
1.0
>>> math.cos(math.pi)
-1.0
>>> dir(math)
['__doc__', '__file__', '__name__', '__package__',
  …,'cos',
'cosh', 'degrees', 'e', 'exp',…, 'pi', 'pow',
  'radians', 'sin', 'sinh', 'sqrt', 'tan',
'tanh', 'trunc']
>>> help(math)
>>> help(math.cos)
```

Exercise 1

Գրեք ծրագիր որը հաշվում է

a) Ուղղանկյուն եռանկյան ներքնաձիգի երկարությունը երբ a=3, b=4
b) Շրջանի մակերեսը երբ r=10

Exercise 1

Գրեք ծրագիր որը հաշվում է

a) Ուղղանկյուն եռանկյան ներքնաձիգի երկարությունը երբ a=3, b=4
b) Շրջանի մակերեսը երբ r=10

```
import math

a = 3
b = 4
r = 10

c = math.sqrt(a**2 + b**2)
s = math.pi*r**2
```

ex31.py

importing modules

Module import անելու տարբեր ձևեր կան:

importing modules

Module import անելու տարբեր ձևեր կան:

```
import somemodule
```

importing modules

Module import անելու տարբեր ձևեր կան:

```
import somemodule
from somemodule import *
```

importing modules

Module import անելու տարբեր ձևեր կան:

```
import somemodule
from somemodule import *
from somemodule import name
```

importing modules

Module import անելու տարբեր ձևեր կան:

```
import somemodule
from somemodule import *
from somemodule import name
from somefile import name as nm
```

importing modules

importing modules

```
>>> dir()
['__builtins__', '__doc__', '__loader__', '__name__',
   '__package__', '__spec__', 'cls']
```

importing modules

```
>>> dir()
['__builtins__', '__doc__', '__loader__', '__name__',
   '__package__', '__spec__', 'cls']
>>> import math
>>> dir()
['__builtins__', '__doc__', '__loader__', '__name__',
   '__package__', '__spec__', 'cls', 'math']
```

importing modules

```
>>> dir()
['__builtins__', '__doc__', '__loader__', '__name__',
   '__package__', '__spec__', 'cls']
>>> import math
>>> dir()
['__builtins__', '__doc__', '__loader__', '__name__',
   '__package__', '__spec__', 'cls', 'math']
>>> from math import cos
>>> dir()
['__builtins__', '__doc__', '__loader__', '__name__',
   '__package__', '__spec__', 'cls', 'cos', 'math']
>>> cos(0)
1.0
```

importing modules

```
>>> dir()
['__builtins__', '__doc__', '__loader__', '__name__',
   '__package__', '__spec__', 'cls']
>>> import math
>>> dir()
['__builtins__', '__doc__', '__loader__', '__name__',
   '__package__', '__spec__', 'cls', 'math']
>>> from math import cos
>>> dir()
['__builtins__', '__doc__', '__loader__', '__name__',
   '__package__', '__spec__', 'cls', 'cos', 'math']
>>> cos(0)
1.0
>>> from math import sin, pi
>>> dir()
['__builtins__', '__doc__', '__loader__', '__name__',
   '__package__', '__spec__', 'cls', 'cos', 'math', 'pi', 'sin']
>>> sin(pi)
1.2246467991473532e-16
```

time module

```
import time
```

```
time.daylight
```

```
0
```

```
time.gmtime()
```

```
time.struct_time(tm_year=2014, tm_mon=12, tm_mday=5, tm_hou
r=10, tm_min=2, tm_sec=41, tm_wday=4, tm_yday=339, tm_isdst
=0)
```

```
time.localtime()
```

```
time.struct_time(tm_year=2014, tm_mon=12, tm_mday=5, tm_hou
r=14, tm_min=3, tm_sec=48, tm_wday=4, tm_yday=339, tm_isdst
=0)
```

```
time.sleep(5) # հինգ վայրկյան ոչինչ մի արա
```

sys module

```
import sys
```

```
sys.platform
```

```
sys.version
```

```
sys.version_info
```

```
print("Start")
sys.exit()
print("Never Printed!")
```

```
print(dir(sys))
```

```
sys.path
```