AI VIET NAM
@aivietnam.edu.vn

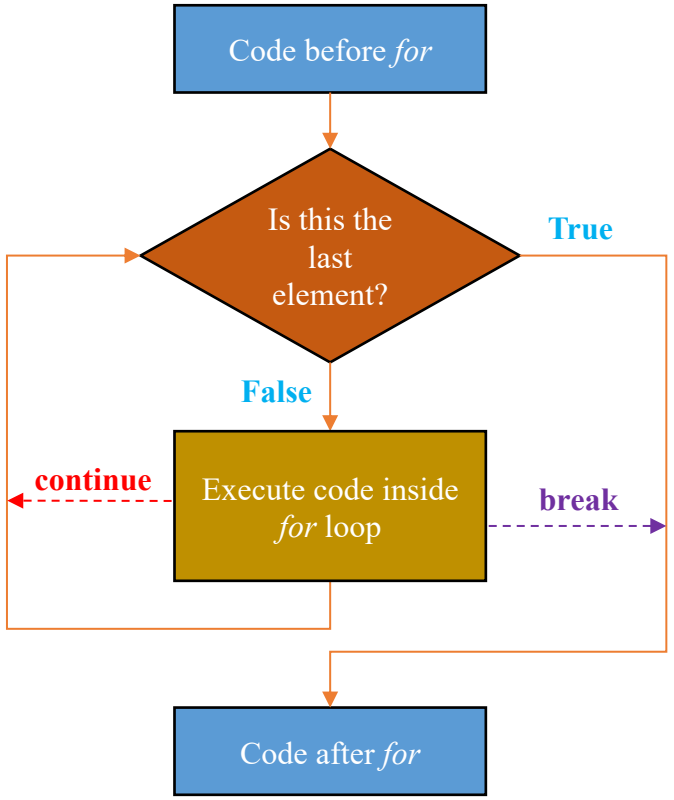
Basic Python for AI

Loops in Python

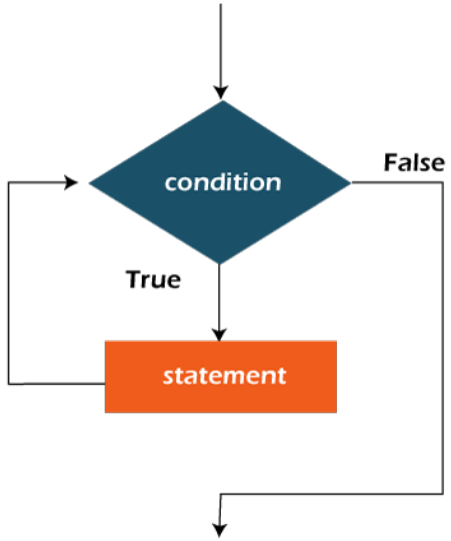
Quang-Vinh Dinh
PhD in Computer Science

Objectives

FOR Loop



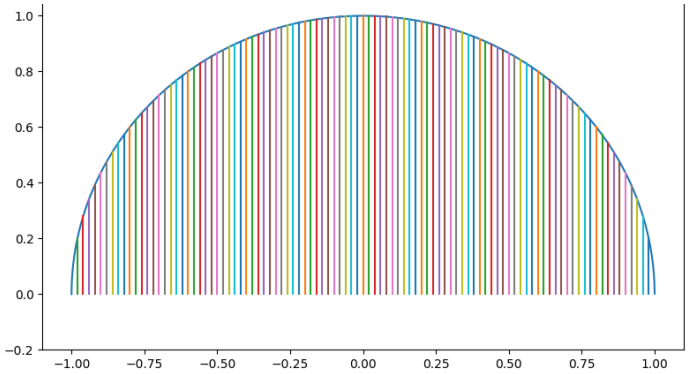
WHILE Loop

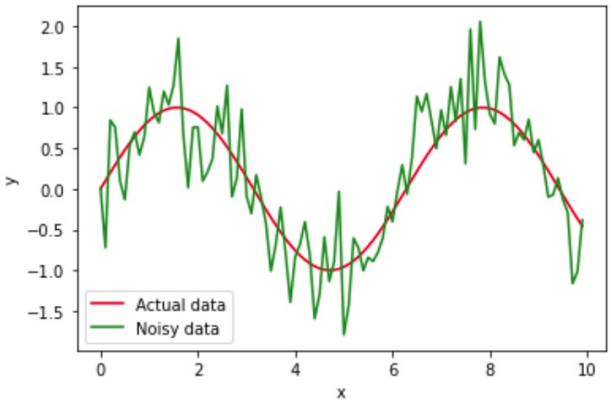



```
# …
while condition:
  # code inside while
# …
```

```
True/False ← condition
```

Examples





Review: Common Errors

❖ Error 1

```
4.  # khai báo biến a = 5
5.  a = 5
6.
7.  # thực hiên a + b, sau đó lưu vào biến c
8.  c = a + b
9.
10. # in giá trị c
11. print(c)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-1-eae96ee94f9f> in <module>
      6
      7 # thực hiên a + b, sau đó lưu vào biến c
----> 8 c = a + b
      9
     10 # in giá trị c

NameError: name 'b' is not defined
```

Review: Common Errors

❖ Error 2

```
4.  # khai báo biến a = 5
5.  a = 5
6.
7.  # in giá trị a
8.  Print(a)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-2-f09db6b2bf7e> in <module>
      6
      7 # in giá trị a
----> 8 Print(a)

NameError: name 'Print' is not defined
```

Review: Common Errors

❖ Error 3

```
4.  # khai báo biến chuỗi s
5.  s = 'Hello AIVIETNAM"
6.
7.  # in giá trị s
8.  print(s)
```

```
  File "<ipython-input-3-96feed73c6b1>", line 5
    s = 'Hello AIVIETNAM"
                        ^
SyntaxError: EOL while scanning string literal
```

Review: Common Errors

❖ Error 4

```
4.  # khai báo biến a và b
5.  a = 5
6.  b = 0
7.
8.  # tính giá trị c bằng a chia cho b
9.  c = a / b
10.
11. # in giá trị c
12. print(c)
```

```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
<ipython-input-4-298e1112d534> in <module>
      7
      8 # tính giá trị c bằng a chia cho b
----> 9 c = a / b
     10
     11 # in giá trị c

ZeroDivisionError: division by zero
```

Review: Common Errors

❖ Error 5

```
4.  # khai báo biến chuỗi s
5.  s = 'AI'
6.
7.  # khai báo biến n có kiểu integer
8.  n = 5
9.
10. # tính giá trị c
11. c = s + n
12.
13. # in giá trị c
14. print(c)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-5-f1e2455fae51> in <module>
      9
     10 # tính giá trị c
---> 11 c = s + n
     12
     13 # in giá trị c

TypeError: must be str, not int
```

Review: Common Errors

❖ Error 6

```
4.  # khai báo biến chuỗi s
5.  s = 'AI'
6.
7.  # khai báo biến n có kiểu integer
8.  n = 5
9.
10. # tính giá trị c
11. c = n + s
12.
13. # in giá trị c
14. print(c)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-8-cecd5289546d> in <module>
      9
     10 # tính giá trị c
---> 11 c = n + s
     12
     13 # in giá trị c

TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

Review: Common Errors

❖ Error 7

```
4.  # khai báo biến a và b
5.  a = 5
6.    b = 6
7.
8.  # thực hiện a + b, sau đó lưu vào biến c
9.  c = a + b
10.
11. # in giá trị c
12. print(c)
```

```
  File "<ipython-input-7-31f64166c395>", line 6
    b = 6
    ^
IndentationError: unexpected indent
```

Review: Common Errors

❖ Error 8

```
4.  import math
5.
6.  number = 20.2
7.  print(math.floor(number)
8.  print(math.pi)
```

```
  File "<ipython-input-15-920005110c33>", line 8
    print(math.pi)
          ^
SyntaxError: invalid syntax
```

❖ Error 9

```
3.
4.  print "aivietnam.ai"
```

```
  File "<ipython-input-3-a46b1c9e05ed>", line 4
    print "aivietnam.ai"
                       ^
SyntaxError: Missing parentheses in call to 'print'. Did you mean print("aivietnam.ai")?
```

Review: Common Errors

❖ Error 10

```
4.  import mymodule
5.
6.  print("aivietnam.ai")
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
<ipython-input-5-1b242c59080b> in <module>
      2 # Lỗi khai báo module không tồn tại
      3
----> 4 import mymodule
      5
      6 print("aivietnam.ai")

ModuleNotFoundError: No module named 'mymodule'
```

Review: Common Errors

❖ Error 11

```
3.
4.  l = [1, 2, 3, 4, 5]
5.  print(l[0])
6.  print(l[5])
```

```
1

---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-6-b8c53176edc0> in <module>
      4 l = [1, 2, 3, 4, 5]
      5 print(l[0])
----> 6 print(l[5])

IndexError: list index out of range
```

Review: Common Errors

❖ Error 12

```
4.  name = "aivietname.ai"
5.  print(name[0])
6.  print(name[50])
```

```
a

---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-7-357ff533411a> in <module>
      4 name = "aivietname.ai"
      5 print(name[0])
----> 6 print(name[50])

IndexError: string index out of range
```

Review: Common Errors

❖ Error 13

```
4.  number = 15
5.  if number < 10
6.      print("A small number")
7.  else:
8.      print("A large number")
```

```
  File "<ipython-input-15-fedf173614ac>", line 5
    if number < 10
                 ^
SyntaxError: invalid syntax
```

❖ Error 14

```
4.  number = 15
5.  if number < 10:
6.      print("A small number")
7.  else
8.      print("A large number")
```

```
  File "<ipython-input-16-699752908646>", line 7
    else
        ^
SyntaxError: invalid syntax
```

Review: Common Errors

❖ Error 15

```
1. import math
2.
3. number = -4
4. print(math.sqrt(number))
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-8-f25b4b744f6e> in <module>
      2
      3 number = -4
----> 4 print(math.sqrt(number))

ValueError: math domain error
```

❖ Error 16

```
4. # import pytorch
5. import torch
6.
7. print(torch.cuda.is_available())
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
<ipython-input-14-680f8ea2b256> in <module>
      1 # import thư viện pytorch
----> 2 import torch
      3
      4 print(torch.cuda.is_available())

ModuleNotFoundError: No module named 'torch'
```

Review: Common Errors

❖ Error 17

```
4 a_number = 5
5 a_string = 'value '
6 result = a_string + a_number
7
8 print(result)
```

```
---------------------------------------------------------------
TypeError                                 Traceback (most
<ipython-input-7-9772d690ff0d> in <module>
      4 a_number = 5
      5 a_string = 'value '
----> 6 result = a_string + a_number
      7
      8 print(result)

TypeError: can only concatenate str (not "int") to str
```

❖ Error 18

```
3 def a_function(x):
4     a_variable = 4
5     result = x*a_variable
6
7     return result
8
9 print(a_variable)
```

```
---------------------------------------------------------------
NameError                                 Traceback (mo
<ipython-input-43-2cce0357dce7> in <module>
      6
      7
----> 8 print(a_variable)

NameError: name 'a_variable' is not defined
```

Review: Common Errors

❖ Error 19

```
4 str1 = '5'
5 str2 = 'hello'
6
7 value1 = int(str1)
8 value2 = int(str2)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-9-1f0b23b26eb1> in <module>
      6
      7 value1 = int(str1)
----> 8 value2 = int(str2)

ValueError: invalid literal for int() with base 10: 'hello'
```

Review: Common Errors

❖ Error 20

```
3
4  def a_function(n):
5      return a_function(n)
6
7  a_function(5)
```

```
---------------------------------------------------------------------------
RecursionError                            Traceback (most recent call last)
<ipython-input-10-bda7ef50bf68> in <module>
      5     return a_function(n)
      6
----> 7 a_function(5)

<ipython-input-10-bda7ef50bf68> in a_function(n)
      3
      4 def a_function(n):
----> 5     return a_function(n)
      6
      7 a_function(5)

... last 1 frames repeated, from the frame below ...

<ipython-input-10-bda7ef50bf68> in a_function(n)
      3
      4 def a_function(n):
----> 5     return a_function(n)
      6
      7 a_function(5)

RecursionError: maximum recursion depth exceeded
```

Outline

SECTION 1

FOR Loop

SECTION 2

WHILE Loop

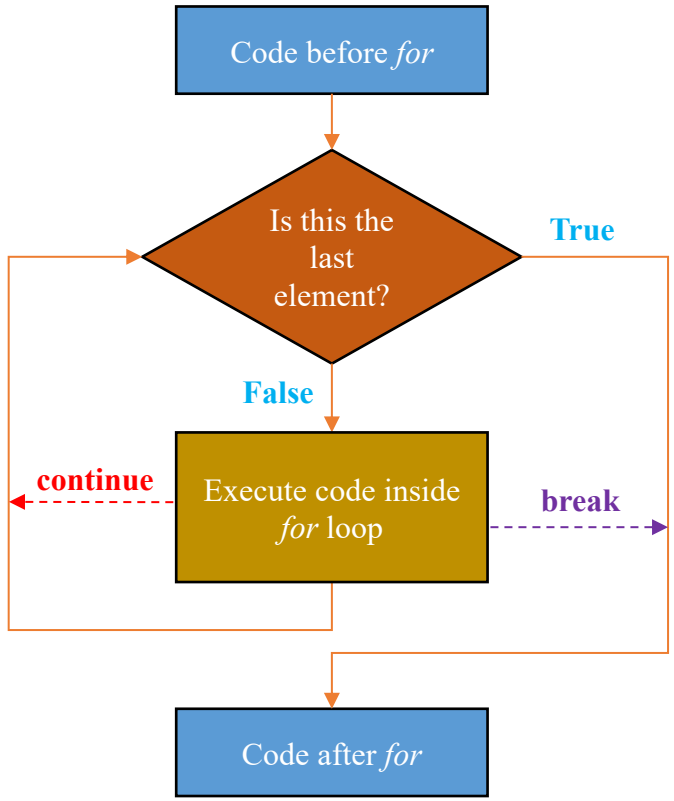
SECTION 3

Examples

***for* syntax**

```
# code before for
for element in iterable:
    # code inside for
# code after for
```

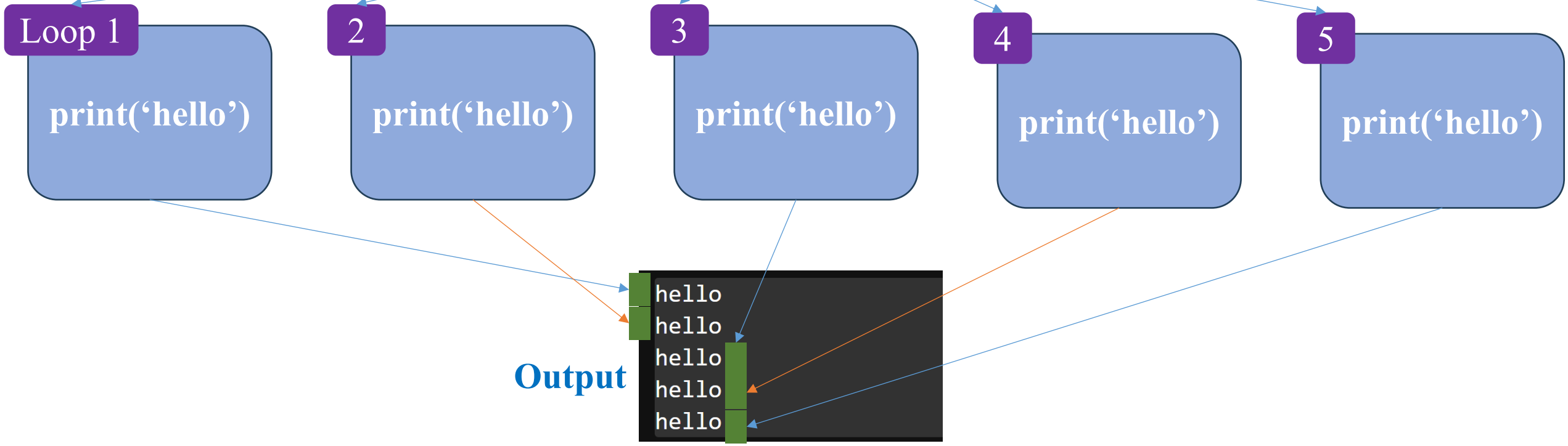
indentation



For Loop

❖ Repeat action in Python

```
# repeat 5 times
for _ in range(5):
    print('hello')
```



For Loop

keyword

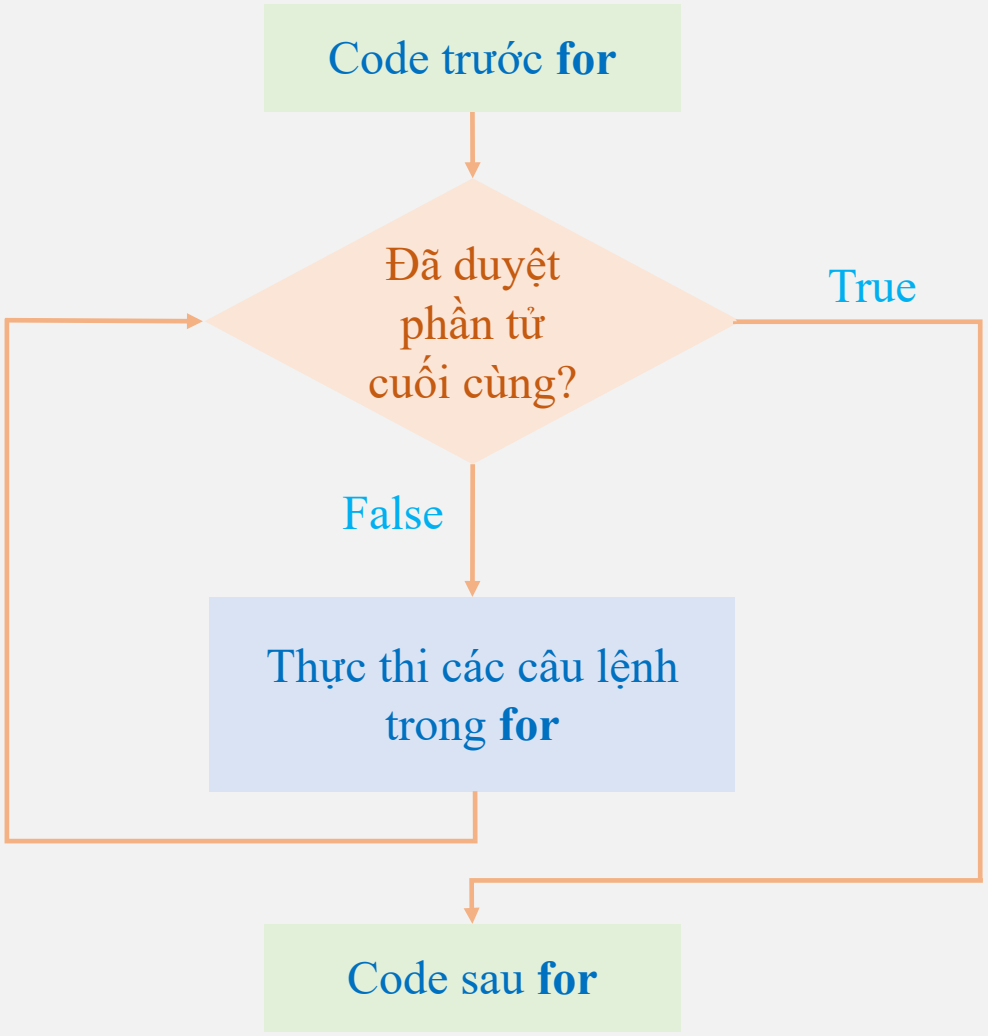
colon

```
# code trước for

for element in iterable:
    # code trong for

# code sau for
```

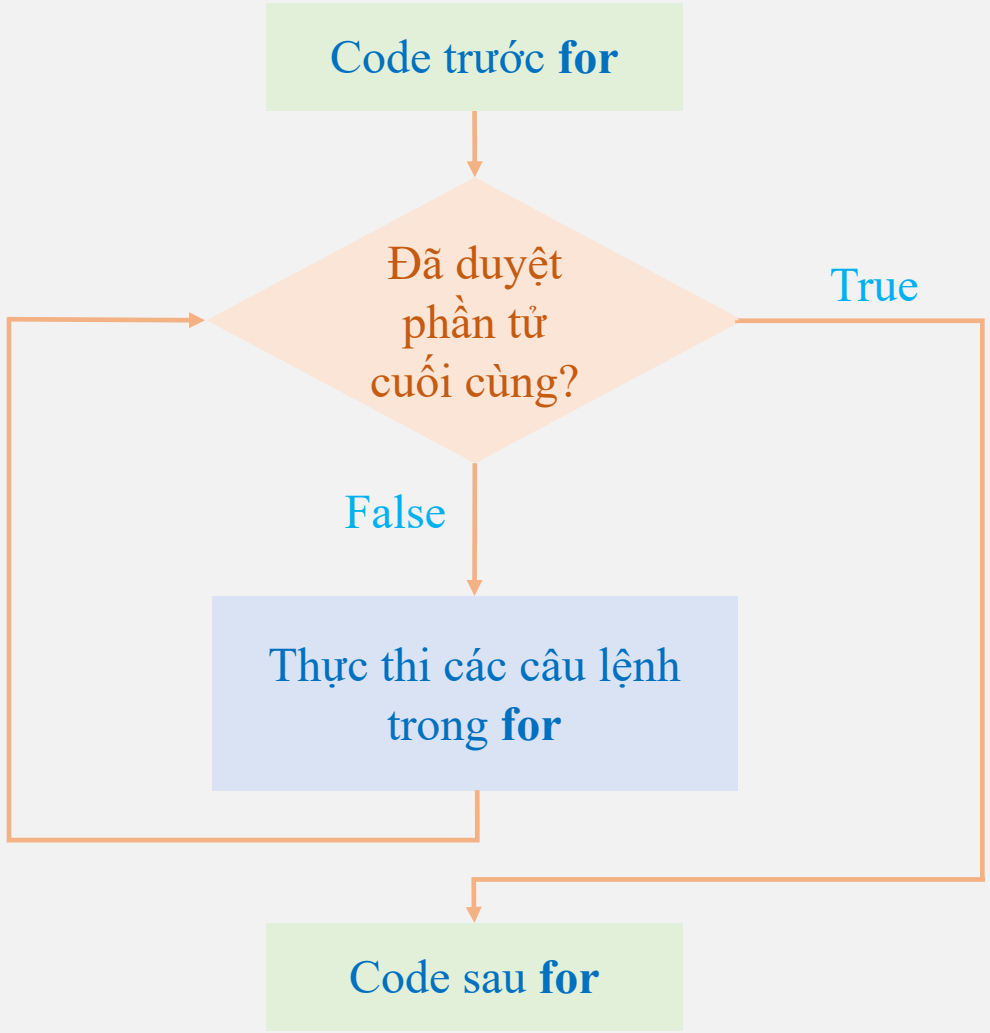
indentation

Iterables

- String
- Tuple
- List
- Dictionary
- range()



For Loop



range(start=0, stop, step=1)

range(start=0, stop=5, step=1)

⬇

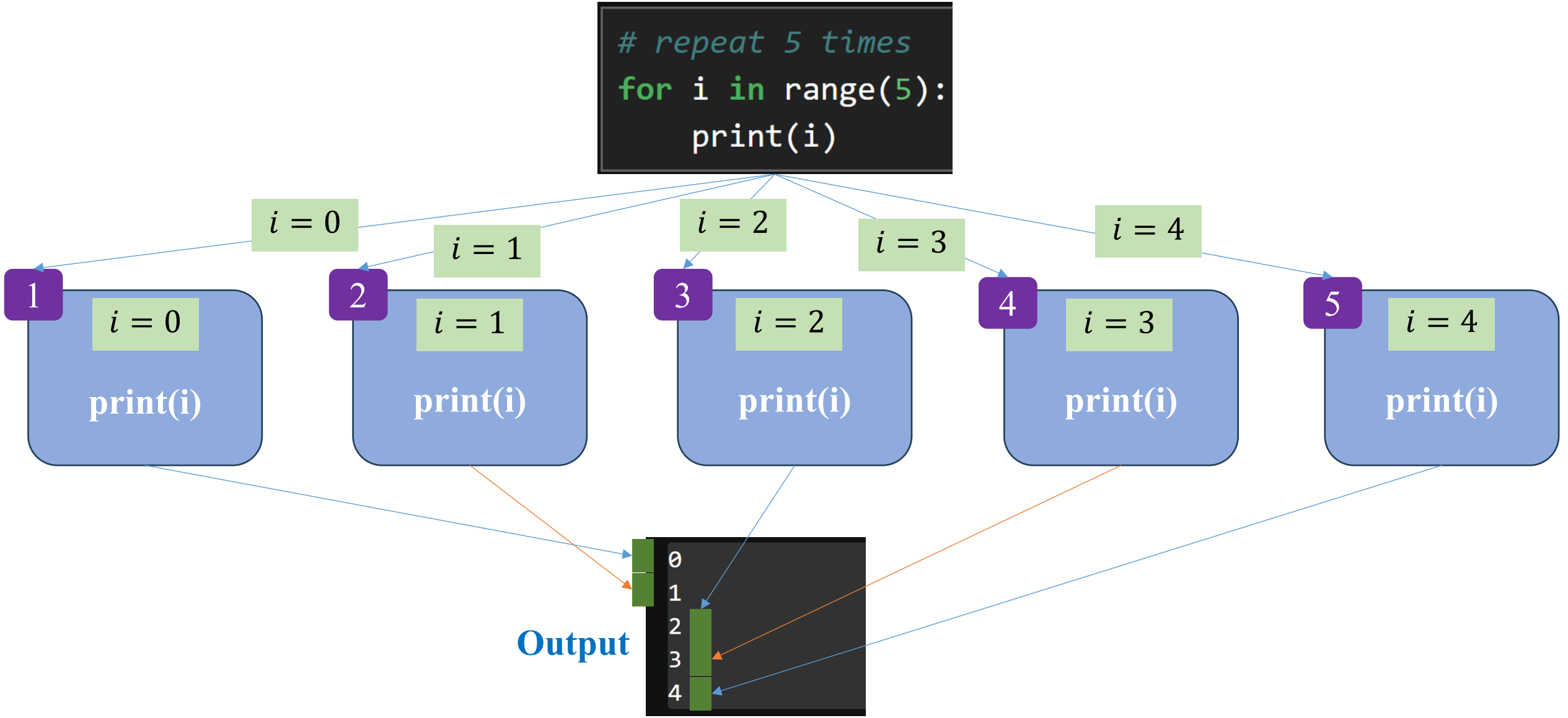
0, 1, 2, 3, 4

range(5)

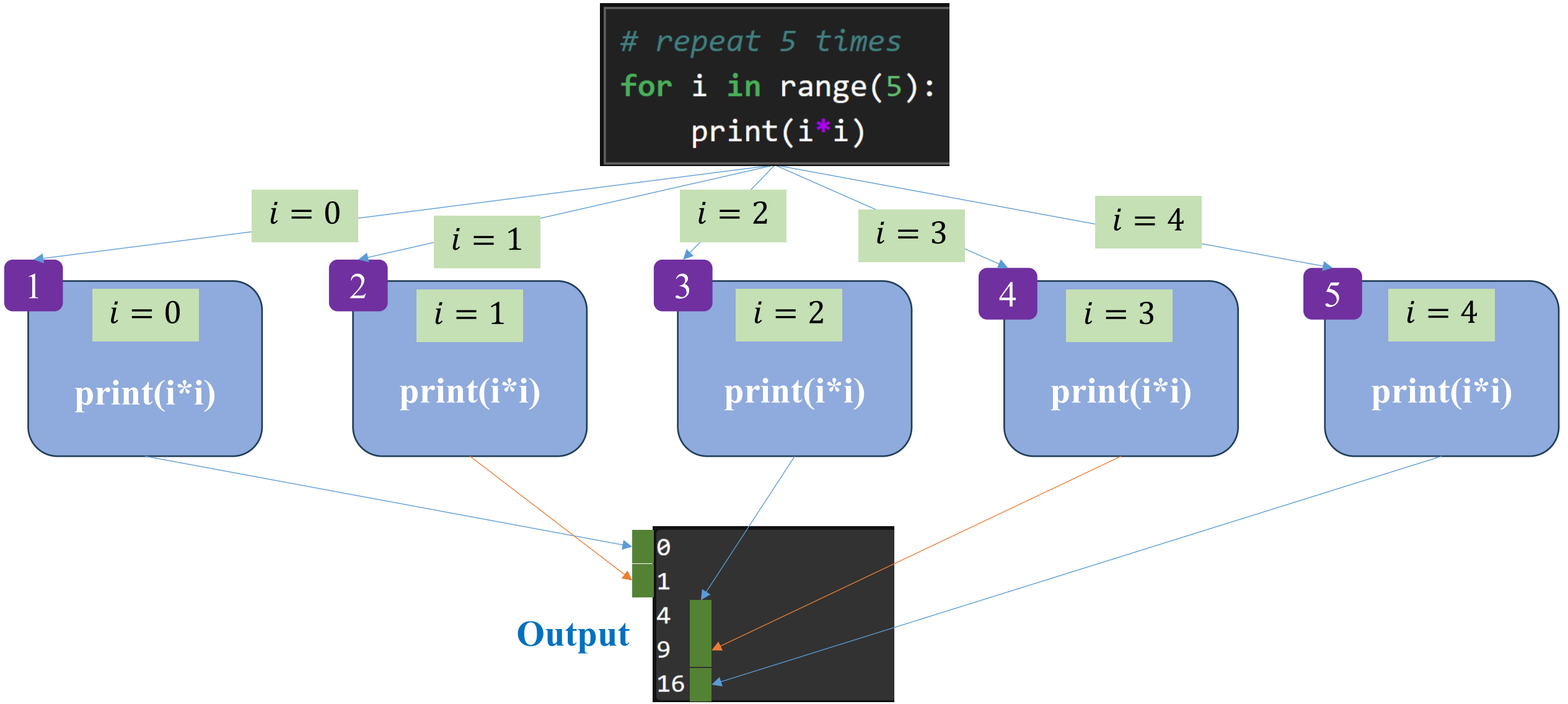
⬇

0, 1, 2, 3, 4

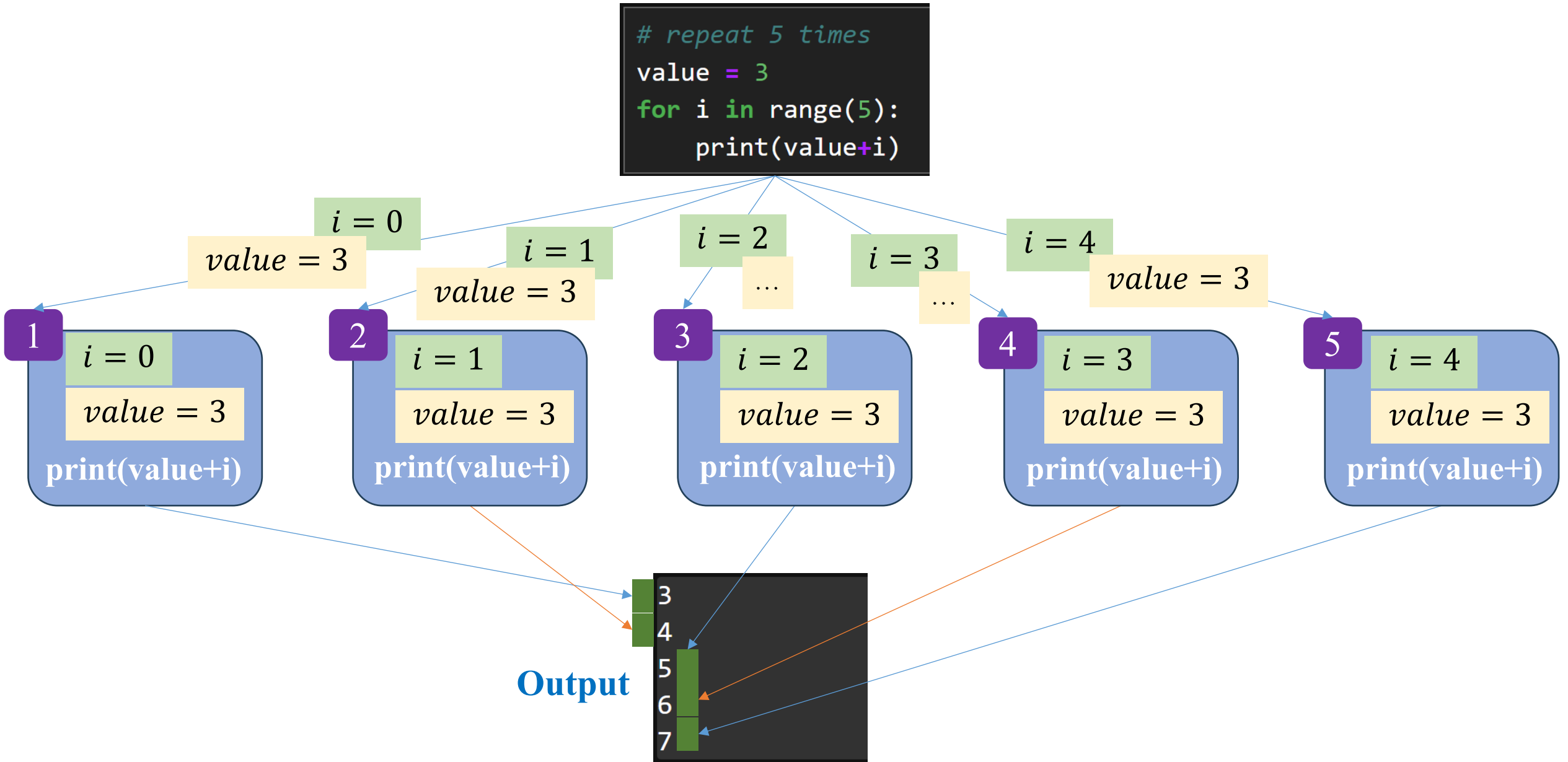
For Loop


# repeat 5 times
for i in range(5):
print(i)
i = 0
i = 1
i = 2
i = 3
i = 4
1
i = 0
print(i)
2
i = 1
print(i)
3
i = 2
print(i)
4
i = 3
print(i)
5
i = 4
print(i)
Output
0
1
2
3
4

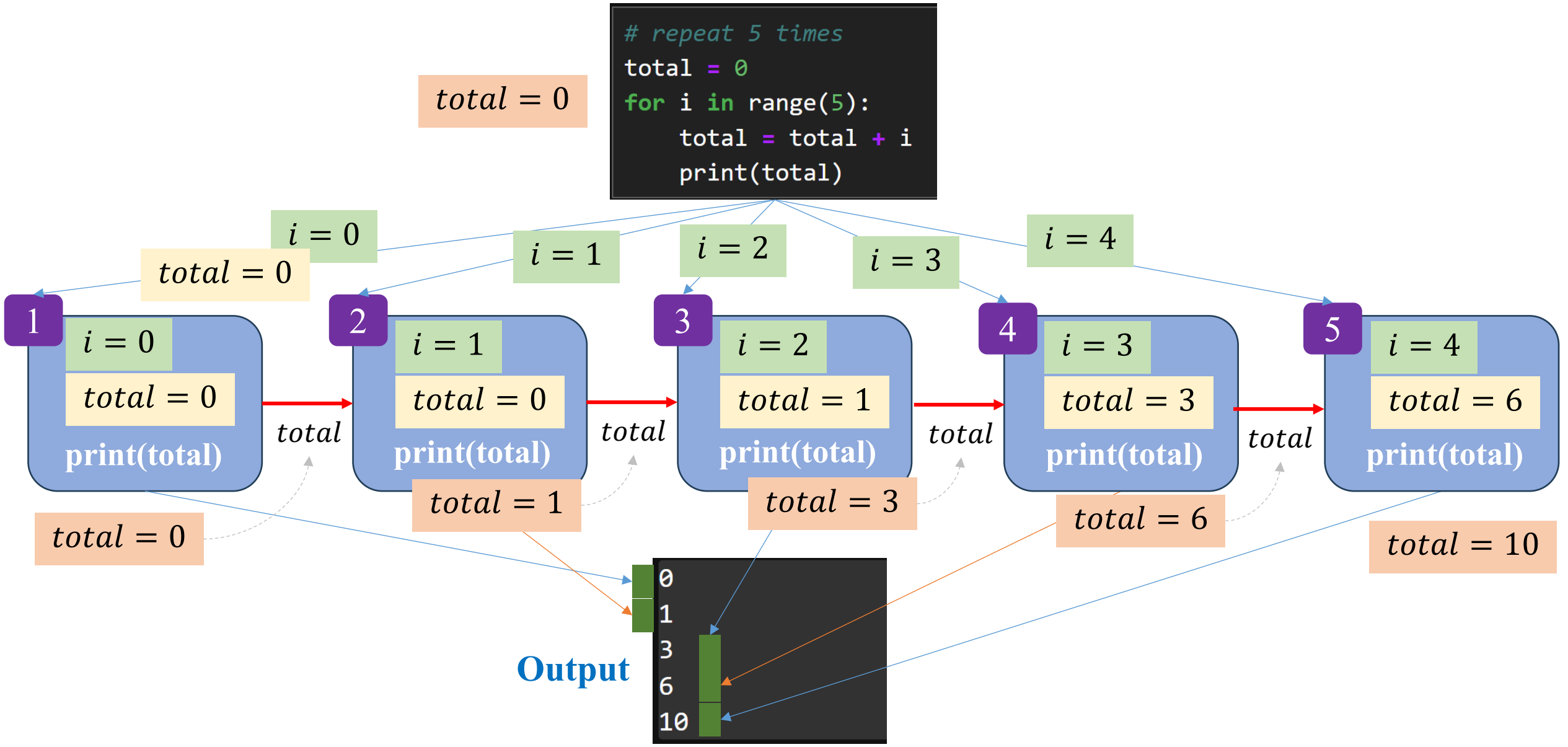
For Loop


# repeat 5 times
for i in range(5):
print(i*i)
i = 0
i = 1
i = 2
i = 3
i = 4
1
2
3
4
5
i = 0
i = 1
i = 2
i = 3
i = 4
print(i*i)
print(i*i)
print(i*i)
print(i*i)
print(i*i)
Output
0
1
4
9
16

For Loop

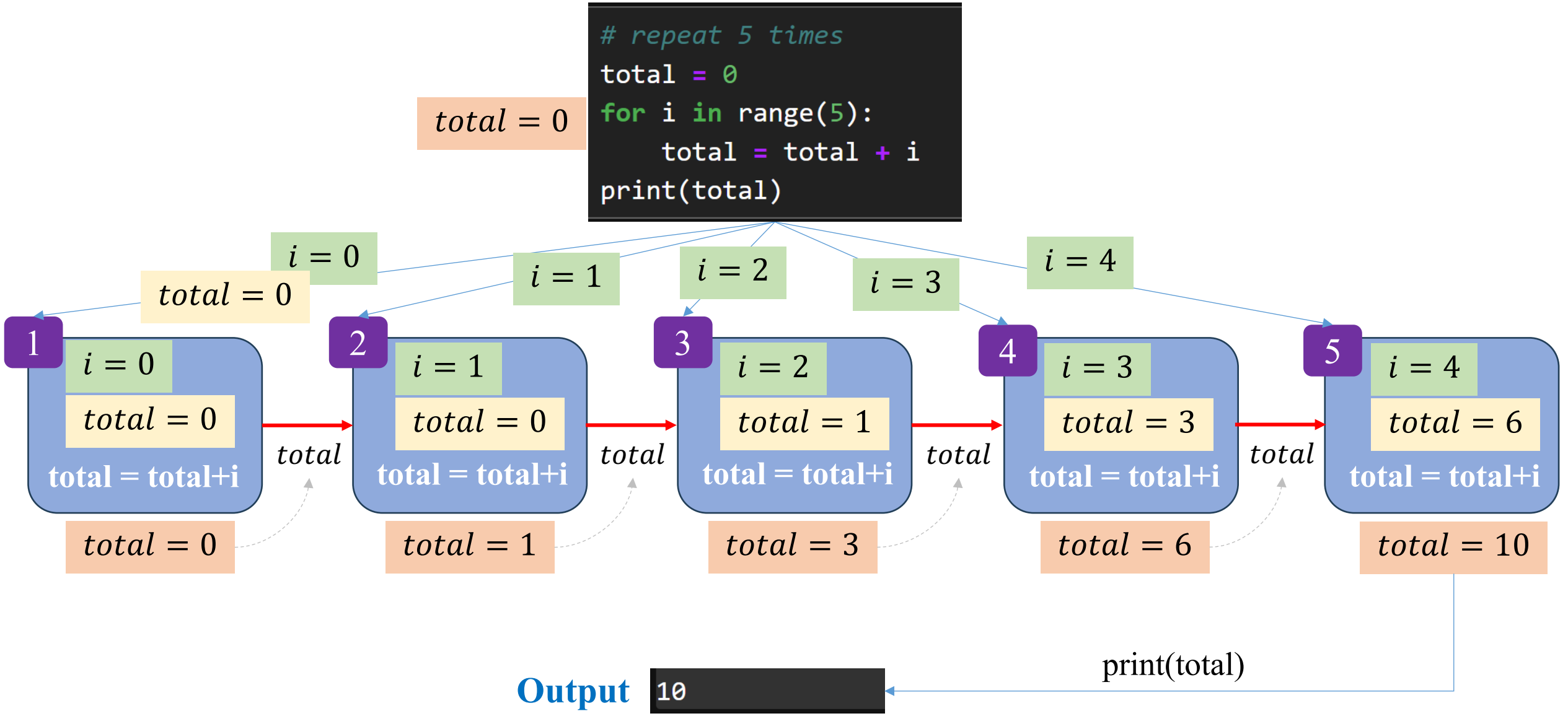

# repeat 5 times
value = 3
for i in range(5):
print(value+i)
i = 0
value = 3
i = 1
value = 3
i = 2
...
i = 3
...
i = 4
value = 3
1
i = 0
value = 3
print(value+i)
2
i = 1
value = 3
print(value+i)
3
i = 2
value = 3
print(value+i)
4
i = 3
value = 3
print(value+i)
5
i = 4
value = 3
print(value+i)
Output
3
4
5
6
7

For Loop


# repeat 5 times
total = 0
for i in range(5):
total = total + i
print(total)
total = 0
i = 0
i = 1
i = 2
i = 3
i = 4
total = 0
1
i = 0
total = 0
print(total)
total
2
i = 1
total = 0
print(total)
total
3
i = 2
total = 1
print(total)
total
4
i = 3
total = 3
print(total)
total
5
i = 4
total = 6
print(total)
total = 0
total = 1
total = 3
total = 6
total = 10
Output
0
1
3
6
10

For Loop


```python
# repeat 5 times
total = 0
for i in range(5):
    total = total + i
print(total)
```

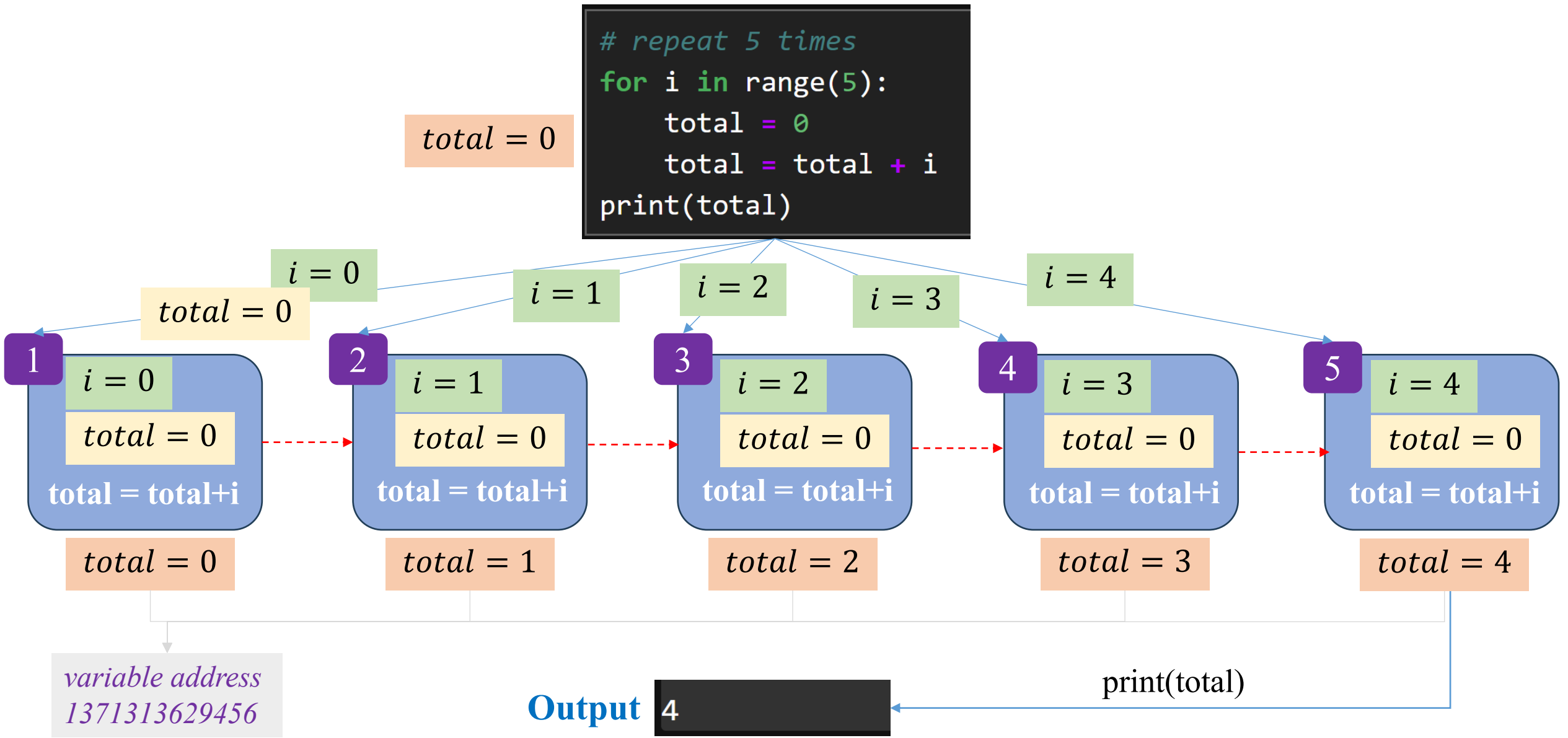


Output

```
10
```

For Loop


```
# repeat 5 times
for i in range(5):
    total = 0
    total = total + i
print(total)
```



For Loop

1

```
begin = 0
end = 5
step = 1
for i in range(begin, end, step):
    print(i)
```

```
0
1
2
3
4
```

2

```
begin = 1
end = 5
step = 1
for i in range(begin, end, step):
    print(i)
```

```
1
2
3
4
```

5

```
n = 3
for i in range(n):
    print(i)
```

```
0
1
2
```

3

```
begin = 1
end = 7
step = 2
for i in range(begin, end, step):
    print(i)
```

```
1
3
5
```

4

```
begin = 1
end = 6
step = 2
for i in range(begin, end, step):
    print(i)
```

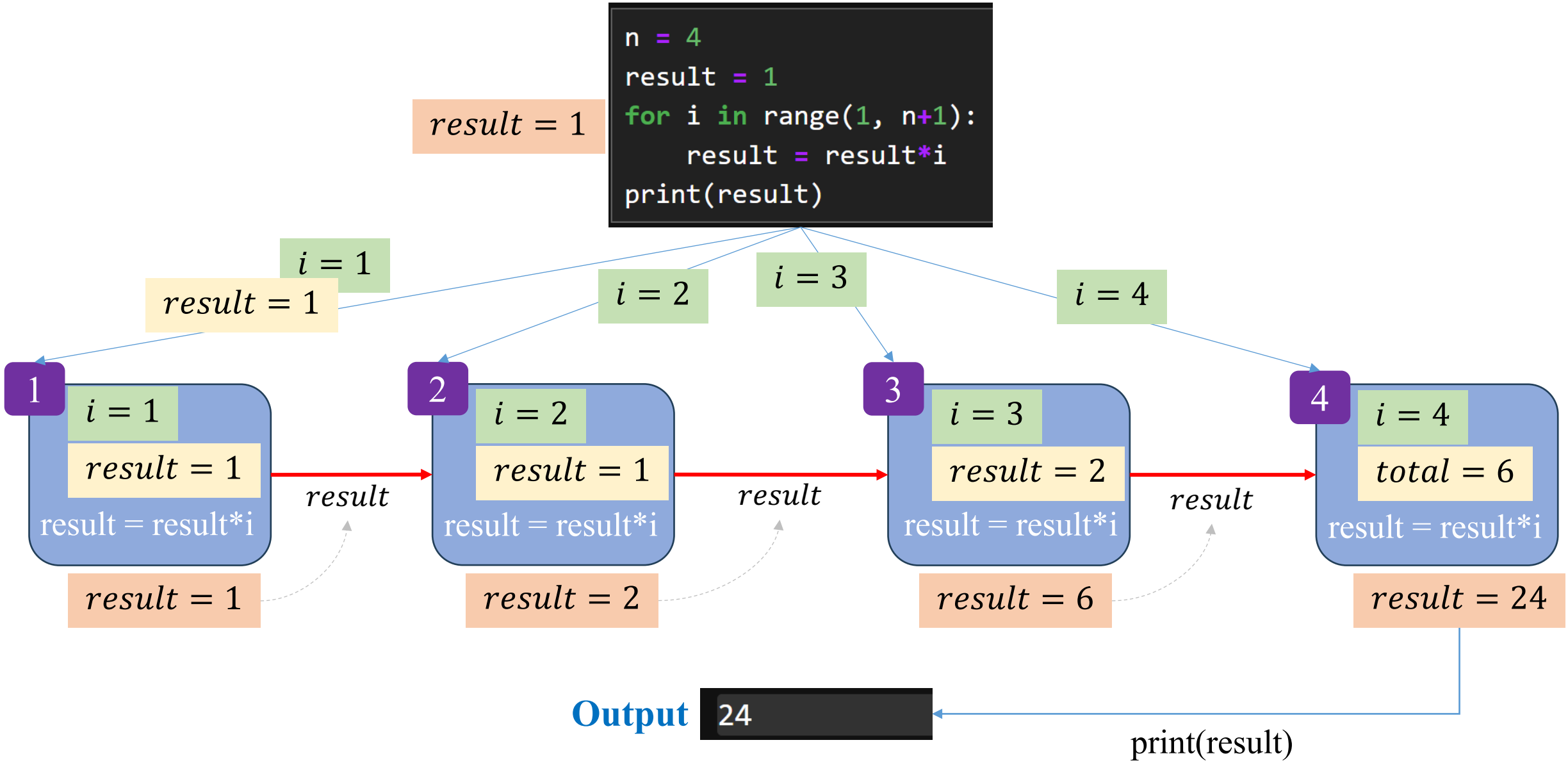
```
1
3
5
```

6

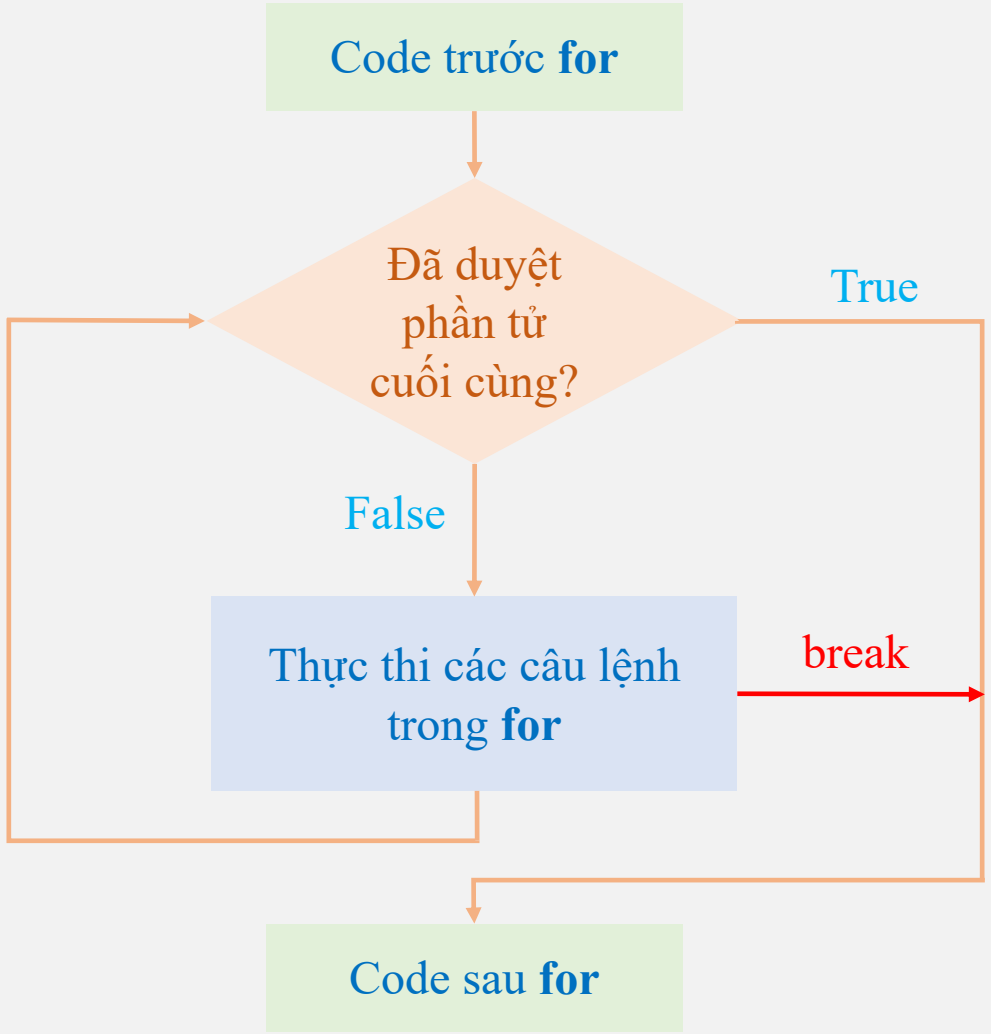
```
n = 3
for i in range(1, n+1):
    print(i)
```

```
1
2
3
```

For Loop


n = 4
result = 1
for i in range(1, n+1):
result = result*i
print(result)
result = 1
i = 1
result = 1
i = 2
i = 3
i = 4
1
i = 1
result = 1
result = result*i
result = 1
result
2
i = 2
result = 1
result = result*i
result = 2
result
3
i = 3
result = 2
result = result*i
result = 6
result
4
i = 4
total = 6
result = result*i
result = 24
Output
24
print(result)

For Loop

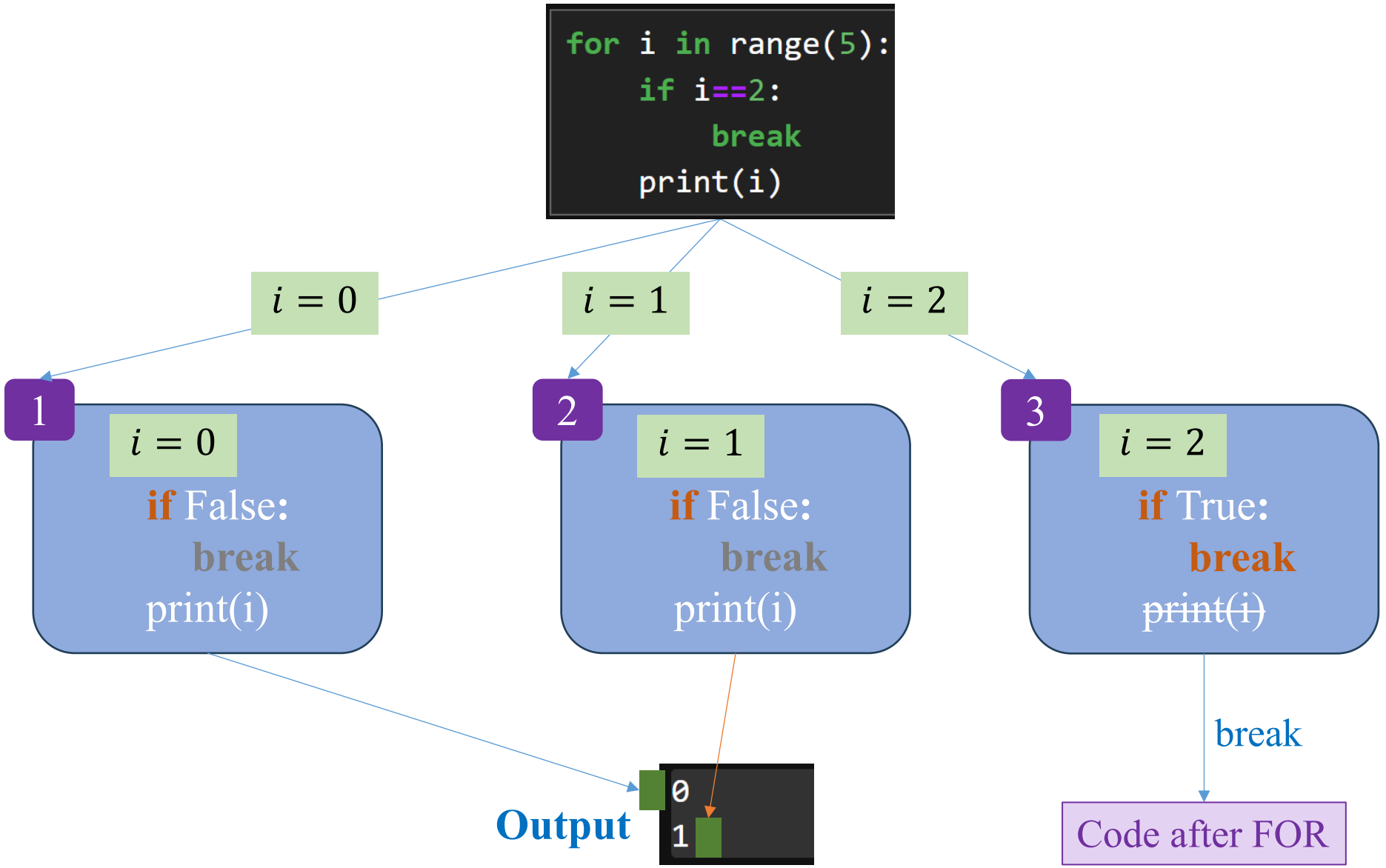


break keyword

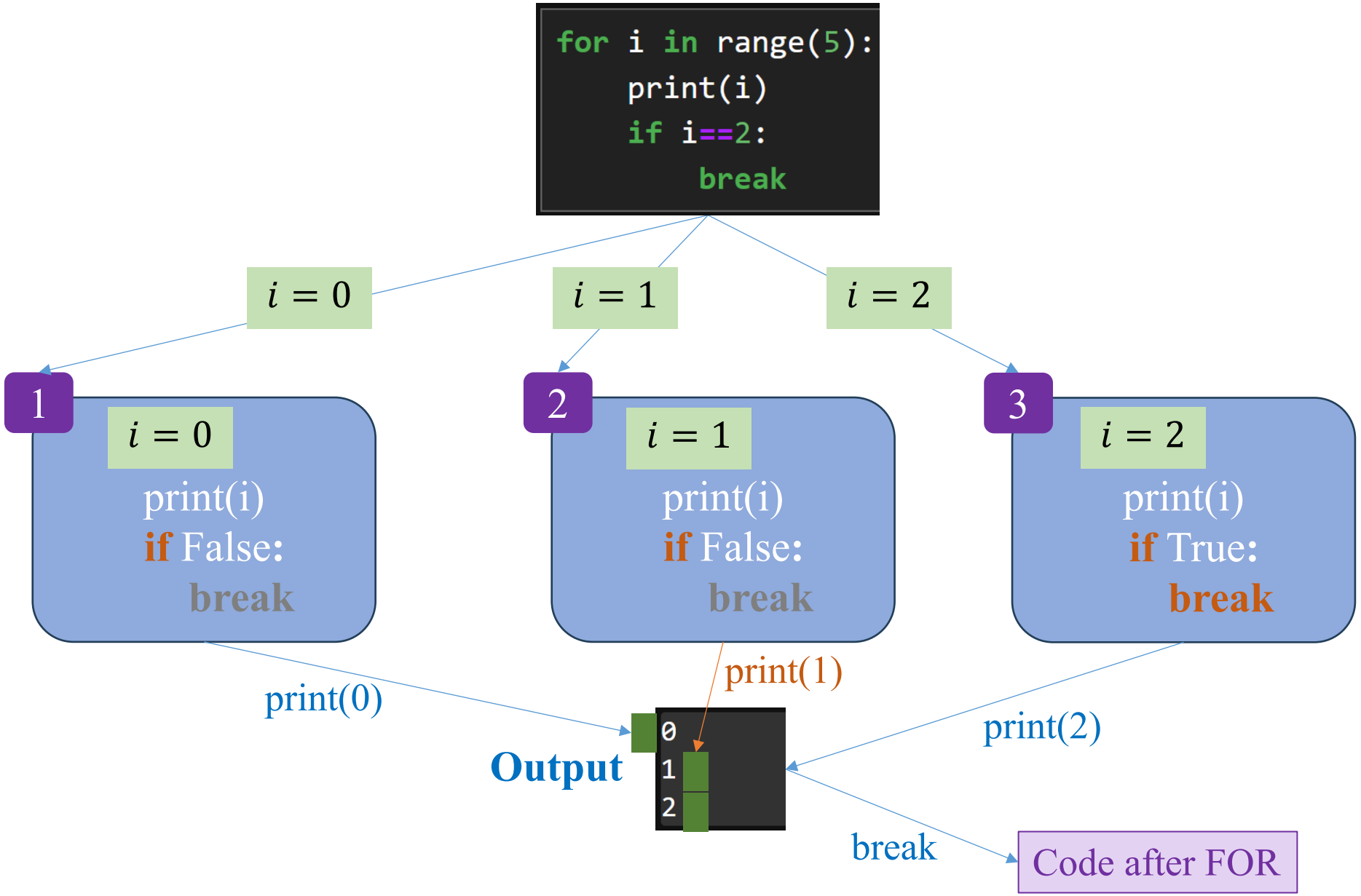
```python
# duyệt phần tử trong range(10)
for i in range(10):
    # hỏi phần tử i có bằng 5 không?
    if i == 5:
        # nếu bằng thì thoát vòng lặp for này
        break

    # làm gì đó với i
    print('Giá trị i là', i)
```

```
Giá trị i là 0
Giá trị i là 1
Giá trị i là 2
Giá trị i là 3
Giá trị i là 4
```

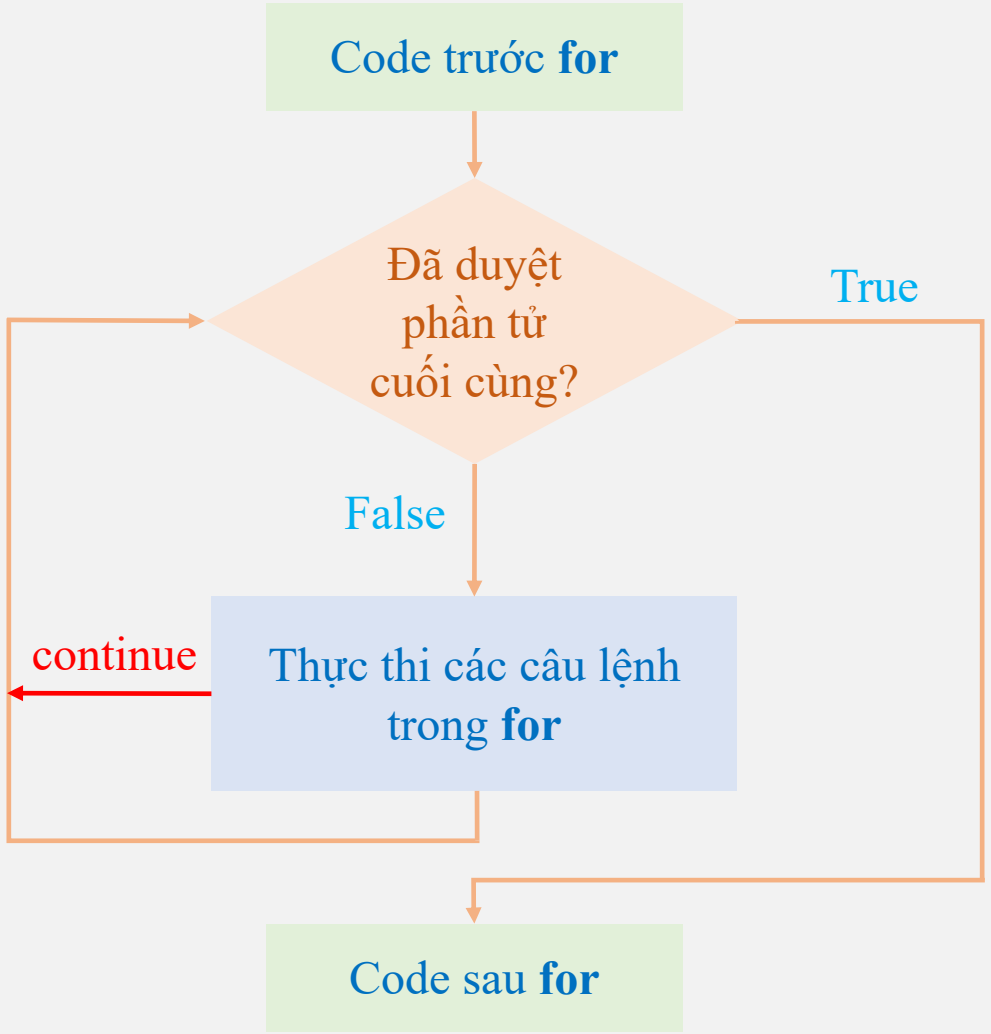
For Loop


for i in range(5):
if i==2:
break
print(i)
i = 0
i = 1
i = 2
1
i = 0
if False:
break
print(i)
2
i = 1
if False:
break
print(i)
3
i = 2
if True:
break
print(i)
Output
0
1
break
Code after FOR

For Loop


for i in range(5):
print(i)
if i==2:
break
i = 0
i = 1
i = 2
1
i = 0
print(i)
if False:
break
2
i = 1
print(i)
if False:
break
3
i = 2
print(i)
if True:
break
print(0)
print(1)
print(2)
Output
0
1
2
break
Code after FOR

For Loop

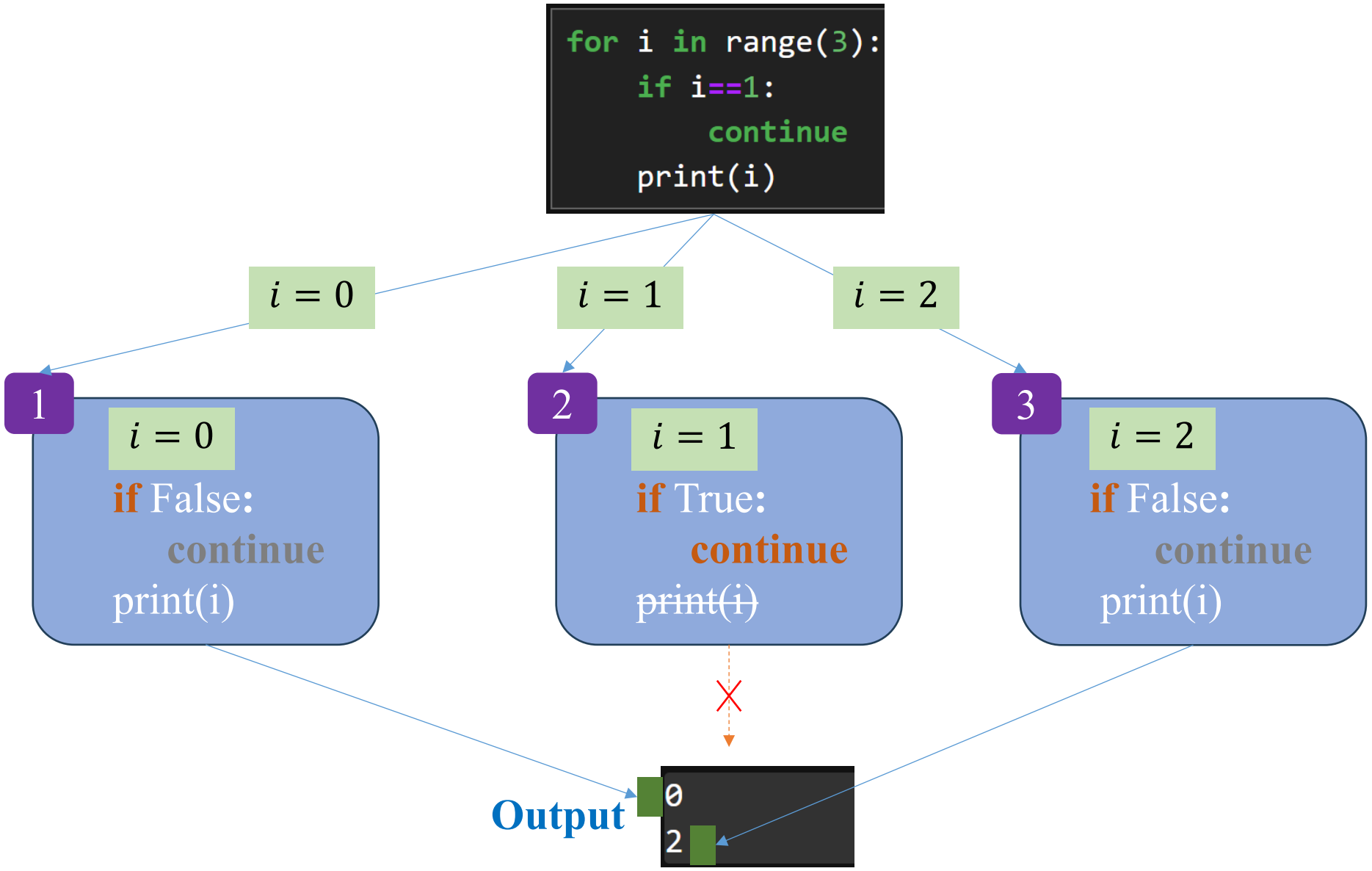


continue keyword

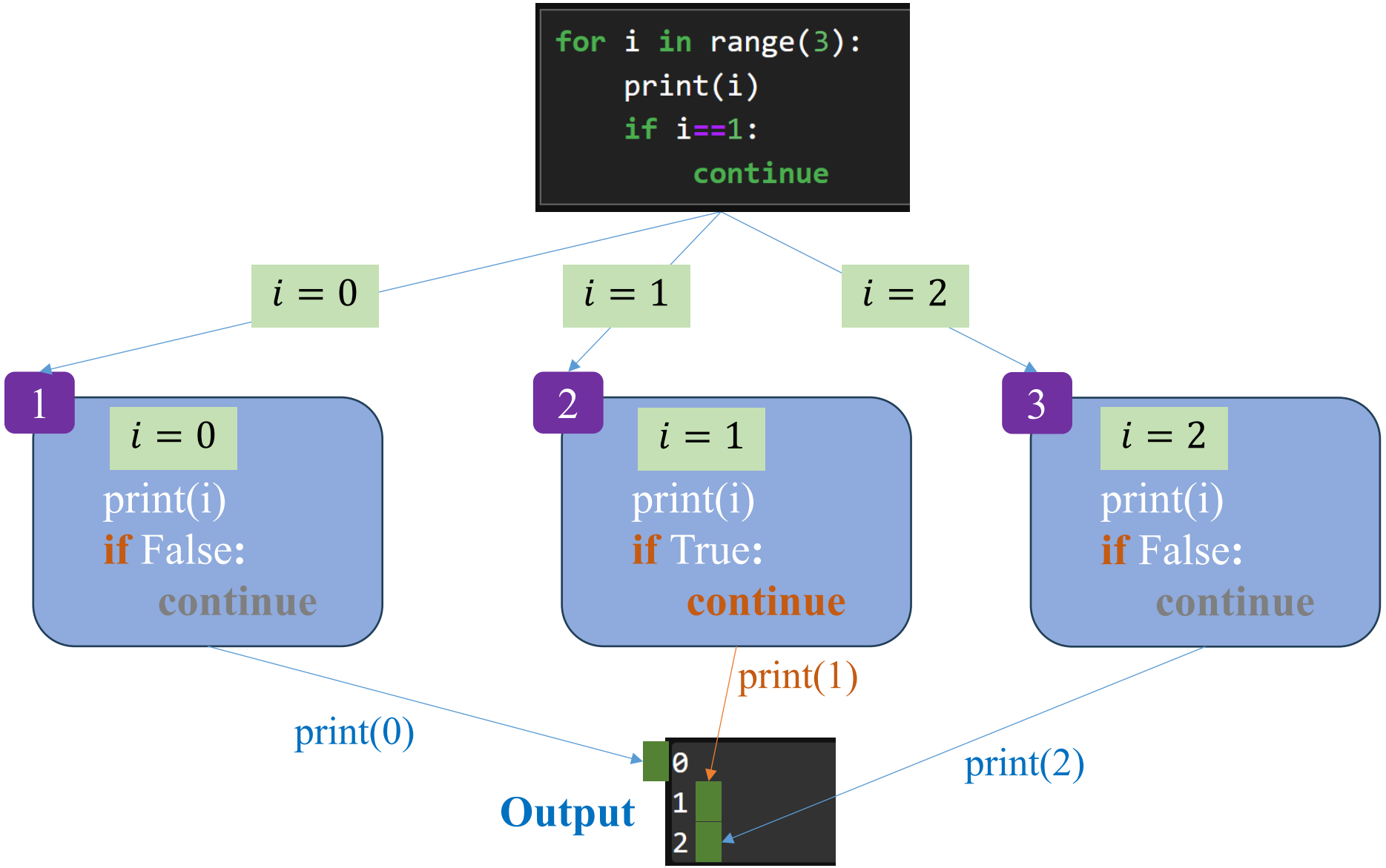
```python
# duyệt phần tử trong range(10)
for i in range(10):
    # hỏi phần tử i có bằng 5 không?
    if i == 5:
        # nếu bằng thì gọi continue
        # phần code sau continue sẽ không
        # được thực thi trong lần lặp này
        continue

    # làm gì đó với i
    print('Giá trị i là', i)
```

```
Giá trị i là 0
Giá trị i là 1
Giá trị i là 2
Giá trị i là 3
Giá trị i là 4
Giá trị i là 6
Giá trị i là 7
Giá trị i là 8
Giá trị i là 9
```

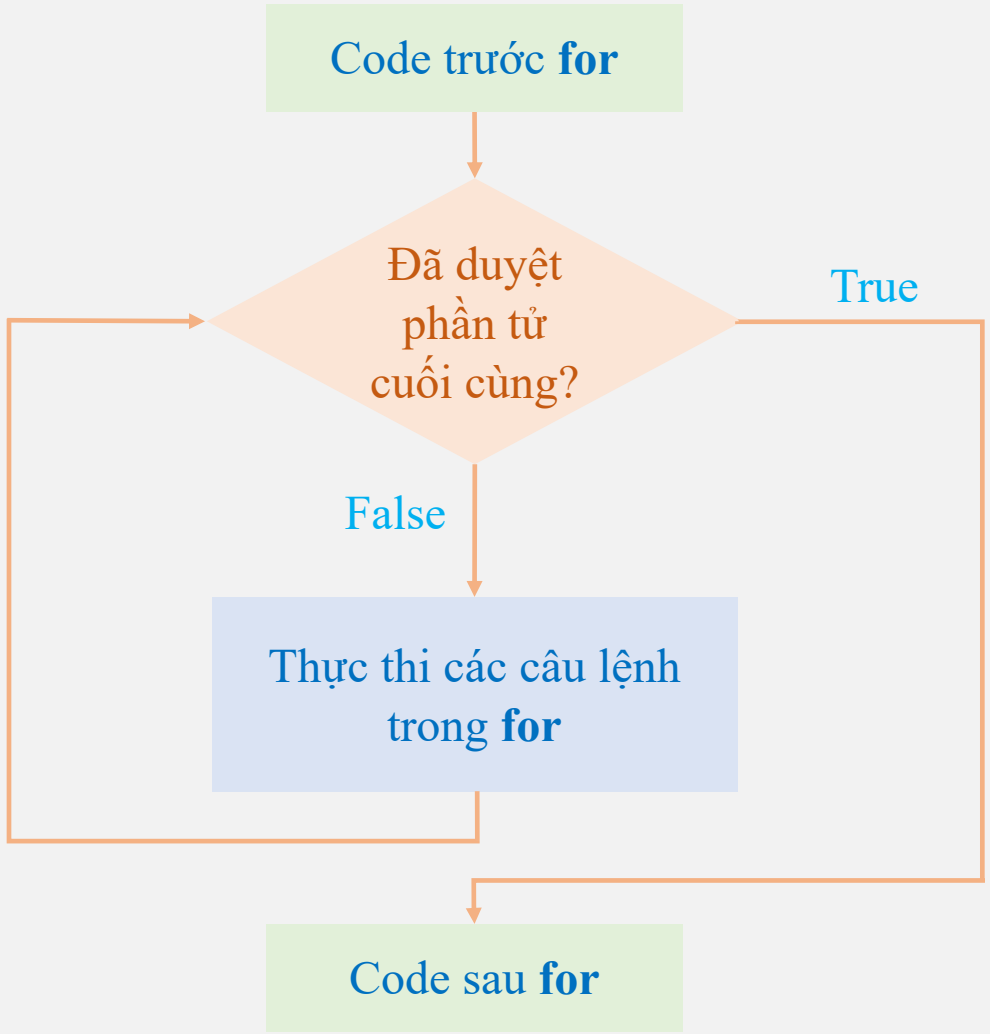
For Loop


for i in range(3):
if i==1:
continue
print(i)
i = 0
i = 1
i = 2
1
i = 0
if False:
continue
print(i)
2
i = 1
if True:
continue
print(i)
3
i = 2
if False:
continue
print(i)
Output
0
2

For Loop


for i in range(3):
print(i)
if i==1:
continue
i = 0
i = 1
i = 2
1
i = 0
print(i)
if False:
continue
2
i = 1
print(i)
if True:
continue
3
i = 2
print(i)
if False:
continue
print(0)
print(1)
print(2)
Output
0
1
2

For Loop




```python
# iterate a list

fruits = ['apple', 'banana', 'melon', 'peach']

for fruit in fruits:
    print(fruit)
```

```
apple
banana
melon
peach
```

```python
# iterate a string

greeting = 'Hello'

for char in greeting:
    print(char)
```

```
H
e
l
l
o
```

```python
# iterate a dictionary

parameters = {'learning_rate': 0.1,
              'optimizer': 'Adam',
              'metric': 'Accuracy'}

for key in parameters:
    print(key, parameters.get(key))
```

```
learning_rate 0.1
optimizer Adam
metric Accuracy
```

```python
# use range()

for i in range(5):
    print(i)
```

```
0
1
2
3
4
```

```python
# iterate a tuple

fruits = ('apple', 'banana', 'melon')

for fruit in fruits:
    print(fruit)
```

```
apple
banana
melon
```

Some Examples

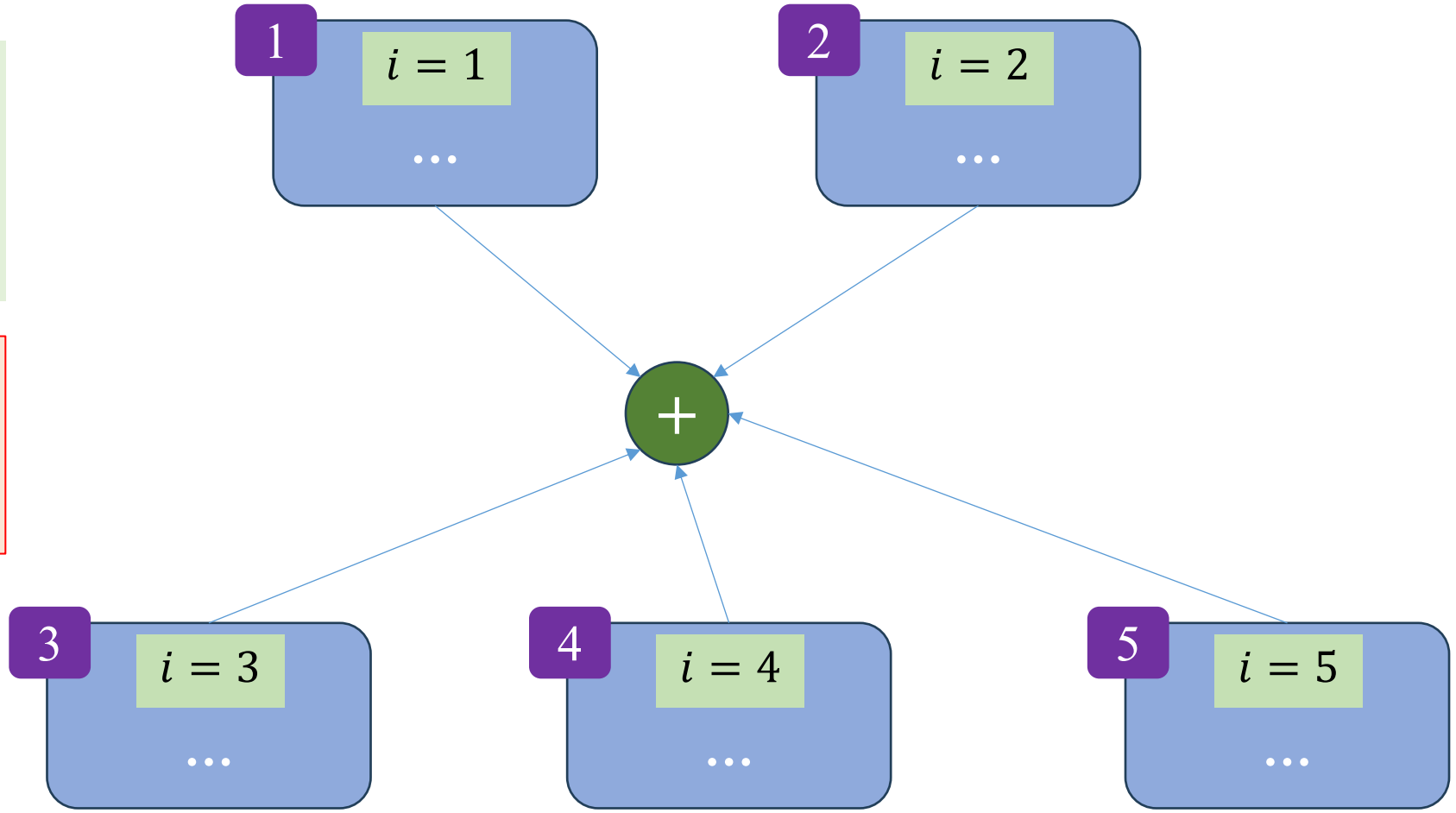
Example

❖ PI estimation

Gregory-Leibniz Series

$$PI \approx 4 \sum_{i=1}^{n} \frac{(-1)^{i+1}}{2i-1}$$

```
n = 5
for i in range(1, n+1):
    …
```



Example

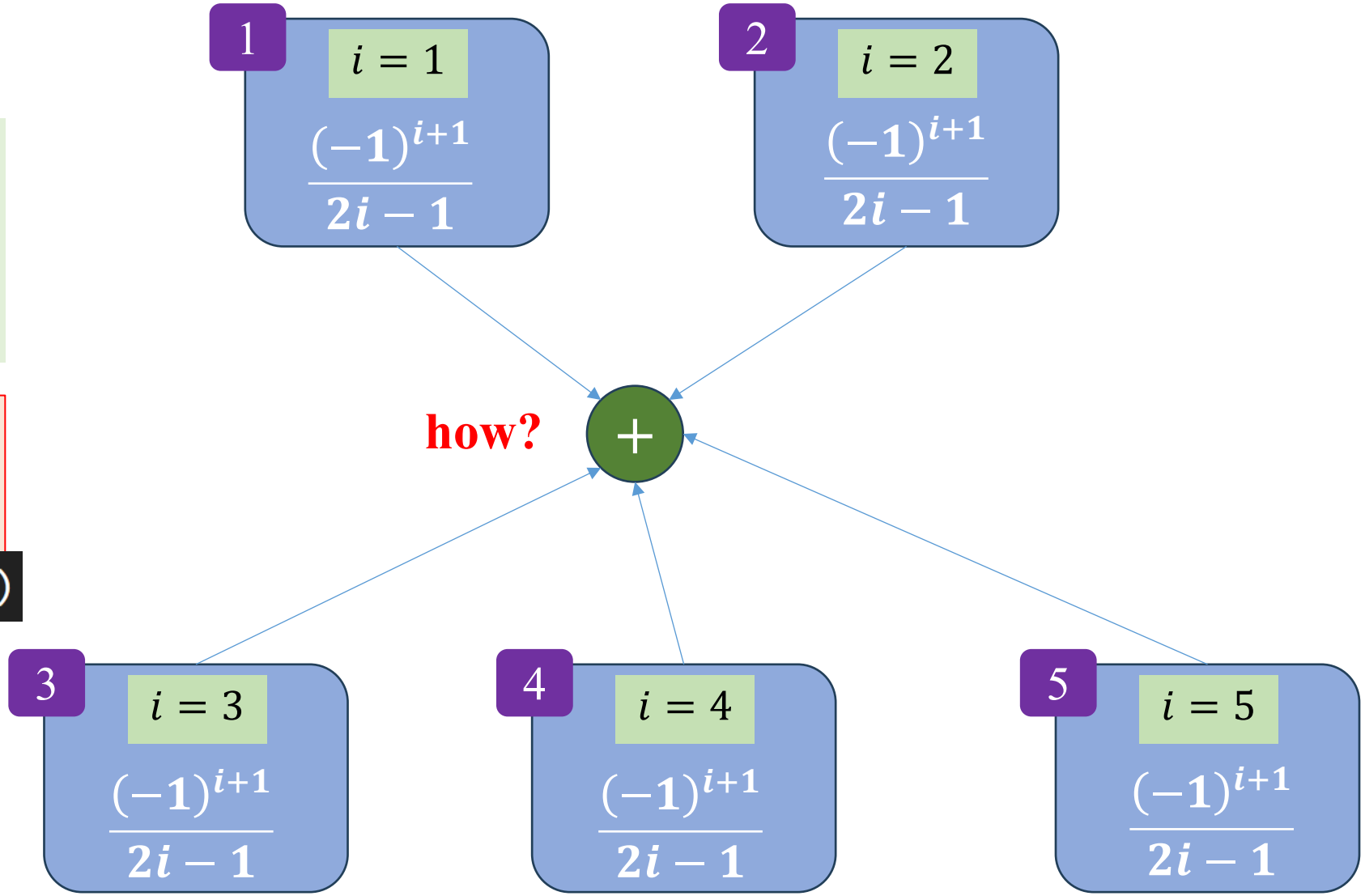
❖ PI estimation

Gregory-Leibniz Series

$$PI \approx 4 \sum_{i=1}^{n} \frac{(-1)^{i+1}}{2i-1}$$

n = 5
for *i* in range(1, n+1):

```
(-1)**(i+1) / (2*i - 1)
```



Example

❖ PI estimation

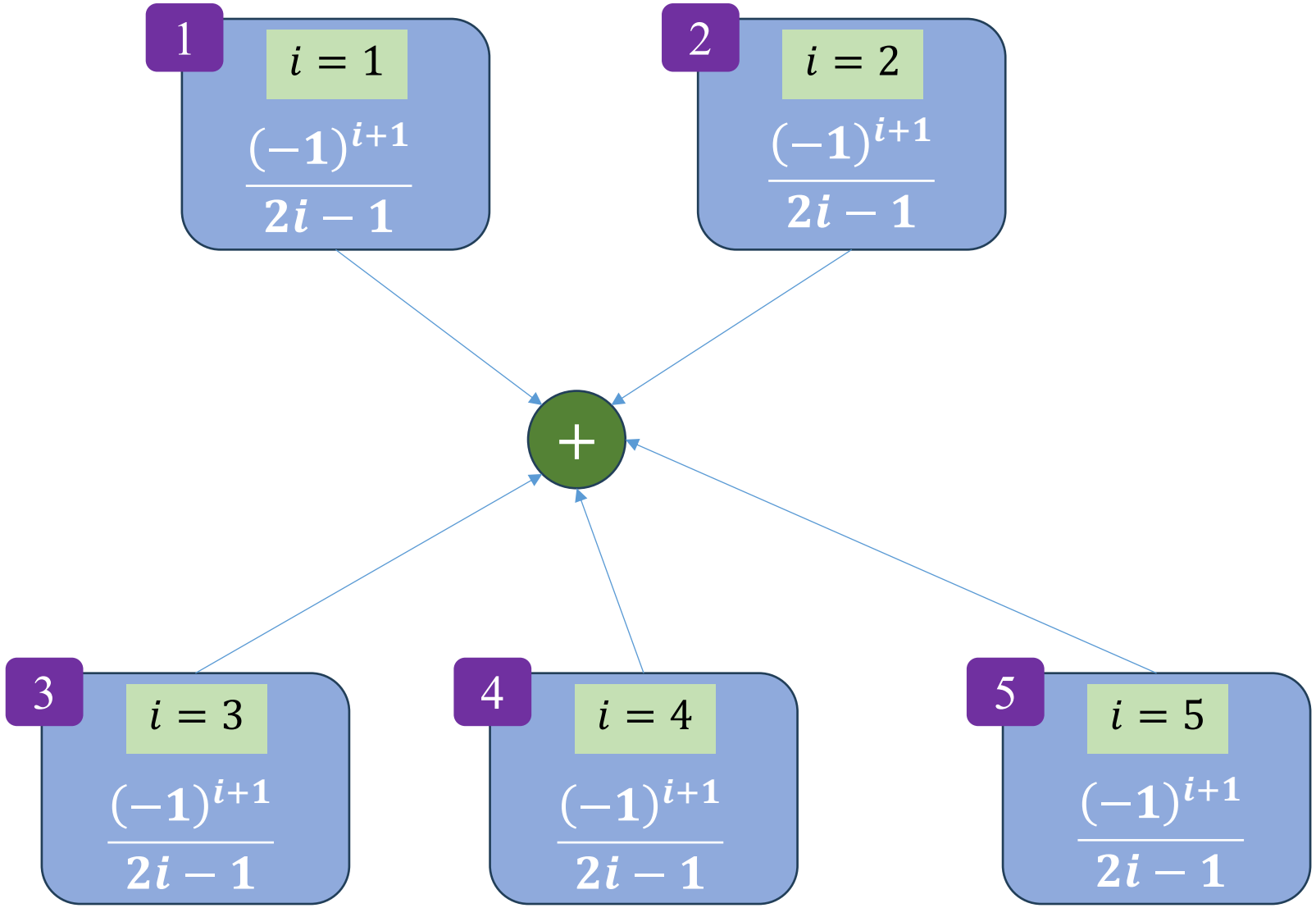
Gregory-Leibniz Series

$$PI \approx 4 \sum_{i=1}^{n} \frac{(-1)^{i+1}}{2i-1}$$

n = 5
result = 0
for *i* in range(1, n+1):
    result = result + …

```
3  n = 1000
4  PI = 0
5  for i in range(1, n):
6      PI = PI + (-1)**(i+1) / (2*i - 1)
7  PI = PI*4
8
9  print('Estimated PI is ', PI)
```

```
Estimated PI is  3.142593654340044
```



Example

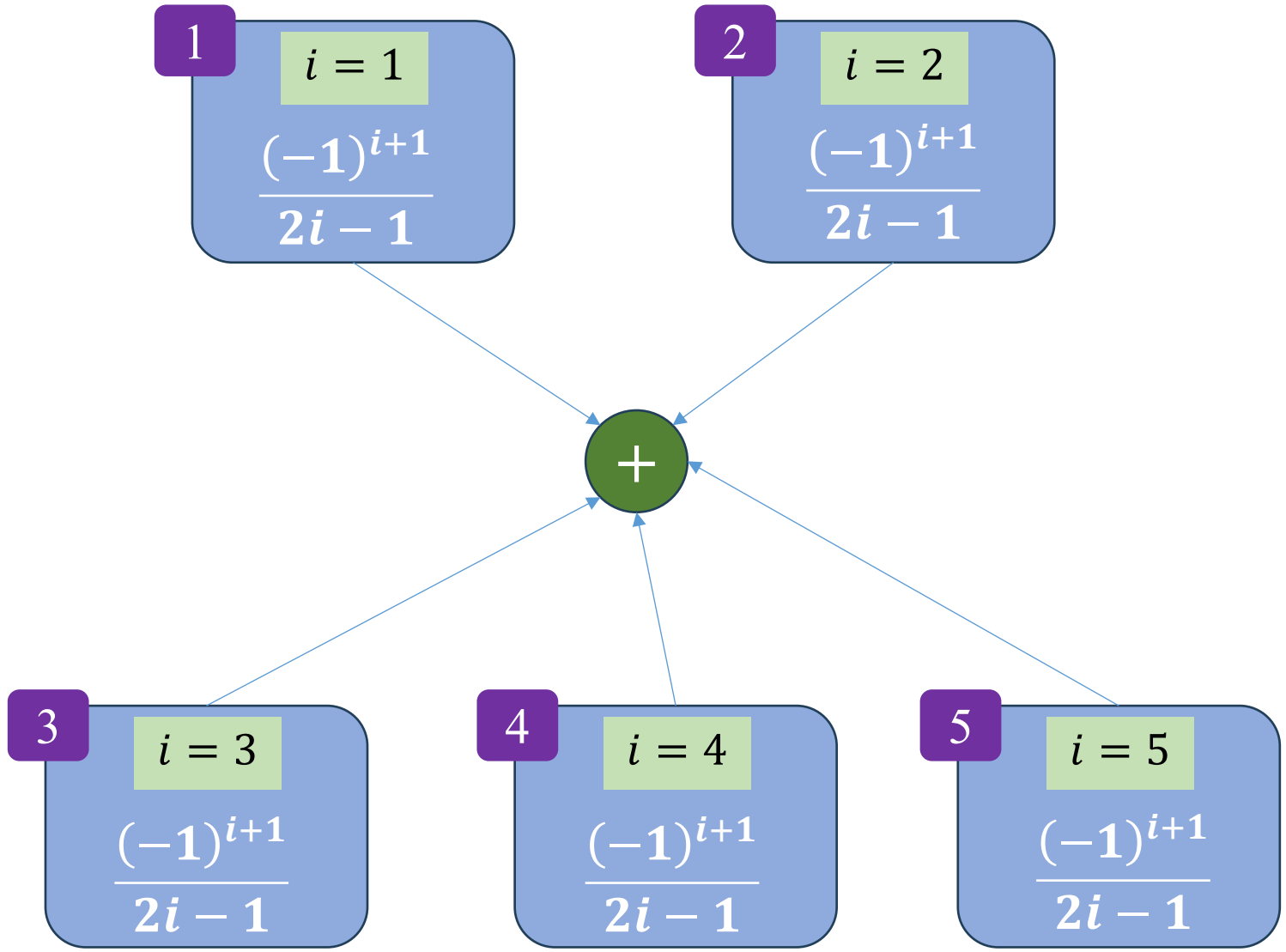
❖ PI estimation

Gregory-Leibniz Series

$$PI \approx 4 \sum_{i=1}^{n} \frac{(-1)^{i+1}}{2i-1}$$

```
3  n = 1000
4  PI = 0
5  for i in range(1, n):
6      PI = PI + (-1)**(i+1) / (2*i - 1)
7  PI = PI*4
8
9  print('Estimated PI is ', PI)
```

```
Estimated PI is  3.142593654340044
```



Example

❖ PI estimation

Gregory-Leibniz Series

$$PI \approx 4 \sum_{i=1}^{n} \frac{(-1)^{i+1}}{2i-1}$$

```python
# Gregory-Leibniz Series

n = 1000
PI = 0
for i in range(1, n):
    PI = PI + (-1)**(i+1) / (2*i - 1)
PI = PI*4

print('Estimated PI is ', PI)
```

```
Estimated PI is  3.142593654340044
```

Nilakantha Series

$$PI \approx 3 + 4 \sum_{i=0}^{n} \frac{-1^{i}}{(2i+2)(2i+3)(2i+4)}$$

```python
# Nilakantha Series

n = 1000
PI = 0
for i in range(n):
    PI = PI + (-1)**(i) / ((2*i+2)*(2*i+3)*(2*i+4))
PI = 3 + 4*PI

print('Estimated PI is ', PI)
```

```
Estimated PI is  3.1415926533405423
```

Example: Quadratic Root

❖ Compute quadratic root for the number N

Newton Method

Given a and n = 0
Set a value for x_0 (x_0 = a/2)

$$x_{n+1} = \frac{x_n + \frac{a}{x_n}}{2}$$

n = n + 1

Compute $\sqrt{9}$

$$a = 9$$
$$set\ x_0 = \frac{9}{2} = 4.5$$
$$n = 0$$

$n = 0$

$$x_1 = \frac{x_0 + \frac{a}{x_0}}{2} = \frac{4.5 + \frac{9}{4.5}}{2} = \frac{6.5}{2} = 3.25$$

$n = 1$

$$x_2 = \frac{x_1 + \frac{a}{x_1}}{2} = \frac{3.25 + \frac{9}{3.25}}{2} = \frac{6.019}{2} = 3.009$$

$n = 2$

$$x_3 = \frac{x_2 + \frac{a}{x_2}}{2} = \frac{3.009 + \frac{9}{3.009}}{2} = 3.00001$$

Example: Quadratic Root

❖ Compute quadratic root for the number N

Newton Method

Given **a** and n = 0
Set a value for x_0 (x_0 = **a**/2)

$$x_{n+1} = \frac{x_n + \frac{a}{x_n}}{2}$$

n = n + 1

$$x_0 = \frac{a}{2}$$

$$x_1 = \frac{x_0 + \frac{a}{x_0}}{2}$$

$$x_2 = \frac{x_1 + \frac{a}{x_1}}{2}$$

$$x_3 = \frac{x_2 + \frac{a}{x_2}}{2}$$

...

x_0, x_1, x_2, x_3

1

```
n = 4
for i in range(n):
    ...
```

2

```
n = 4
for _ in range(n):
    ...
```

Which one is better?

How to propagate information?

Example: Quadratic Root

❖ Compute quadratic root for the number N

Newton Method

Given **a** and n = 0
Set a value for x_0 (x_0 = **a**/2)

$$x_{n+1} = \frac{x_n + \frac{a}{x_n}}{2}$$

n = n + 1

x_0

$$x_0 = \frac{a}{2}$$

x_1

$$x_1 = \frac{x_0 + \frac{a}{x_0}}{2}$$

x_2

$$x_2 = \frac{x_1 + \frac{a}{x_1}}{2}$$

x_3

$$x_3 = \frac{x_2 + \frac{a}{x_2}}{2}$$

...

```
n = 4
global_info = …
for _ in range(n):
    # do something
    # update global_info
```

Example: Quadratic Root

❖ Compute quadratic root for the number N

Newton Method

Given **a** and n = 0
Set a value for x_0 (x_0 = **a**/2)

$$x_{n+1} = \frac{x_n + \frac{a}{x_n}}{2}$$

n = n + 1

$r = \frac{a}{2}$

$r = \frac{r + \frac{a}{r}}{2}$

$r = \frac{r + \frac{a}{r}}{2}$

$r = \frac{r + \frac{a}{r}}{2}$

$r = \frac{r + \frac{a}{r}}{2}$

...

Compute the right side first, then assign to r

```
n = 4
global_info = …
for _ in range(n):
    # do something
    # update global_info
```

```
n = 4
result = a/2
for _ in range(n):
    value = (result + a/result)/2
    result = value
```

Example: Quadratic Root

❖ Compute quadratic root for the number N

Newton Method

Set a value for x_0; n = 0
(x_0 = a/2)

$$x_{n+1} = \frac{x_n + \frac{a}{x_n}}{2}$$

n = n + 1

```python
def compute_square_root(a, n):
    '''
    This function aims to compute square root for the number a

    a -- the number needs to take the square root
    n -- the number of loops used for this optimization
    '''
    result = a/2.0

    for _ in range(n):
        result = (result + a/result) / 2.0

    return result

print(compute_square_root(a=9, n=5))
print(compute_square_root(a=16, n=5))
```

```
3.0
4.000000000000004
```

Outline

SECTION 1

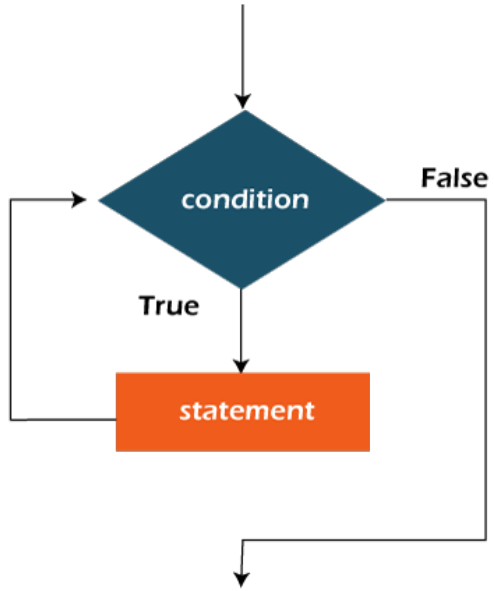
FOR Loop

SECTION 2

WHILE Loop

SECTION 3

Examples




```
# …
while condition:
  # code inside while
# …
```

True/False ← condition

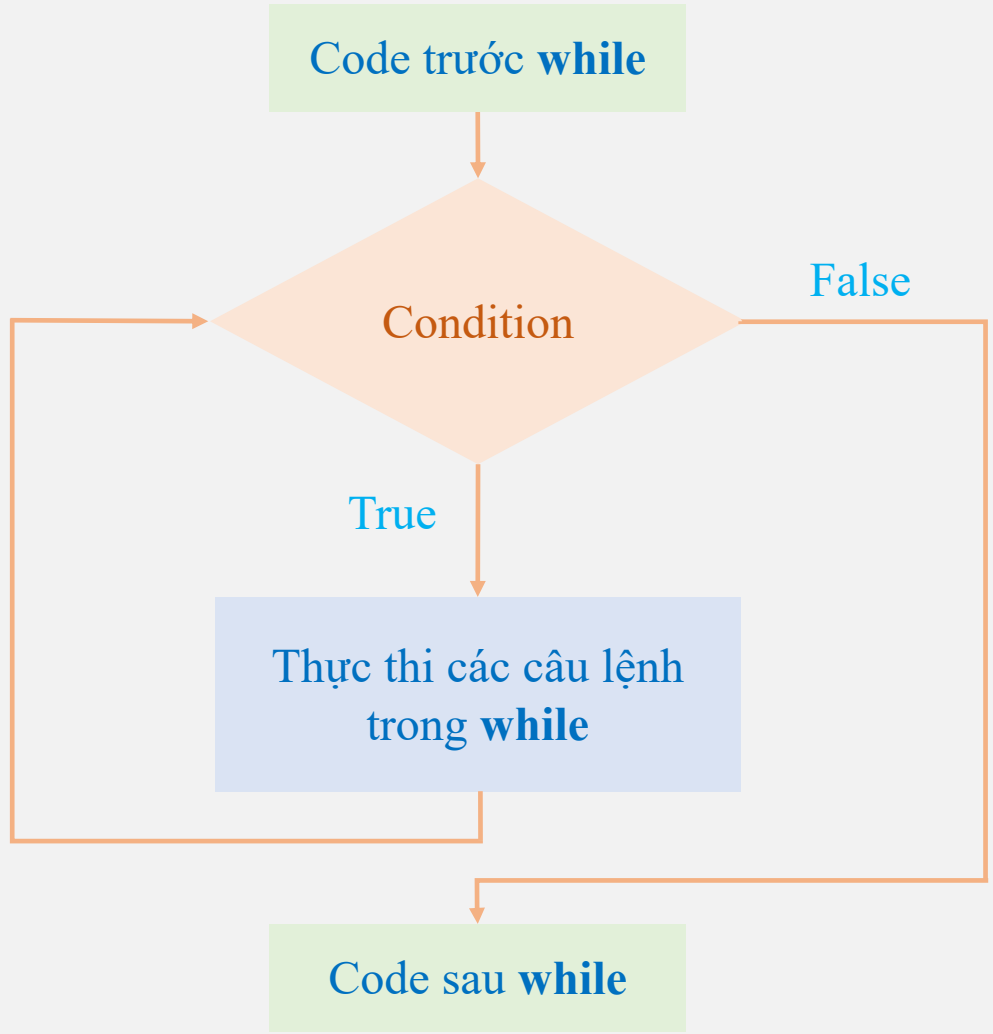
While Loop

keyword

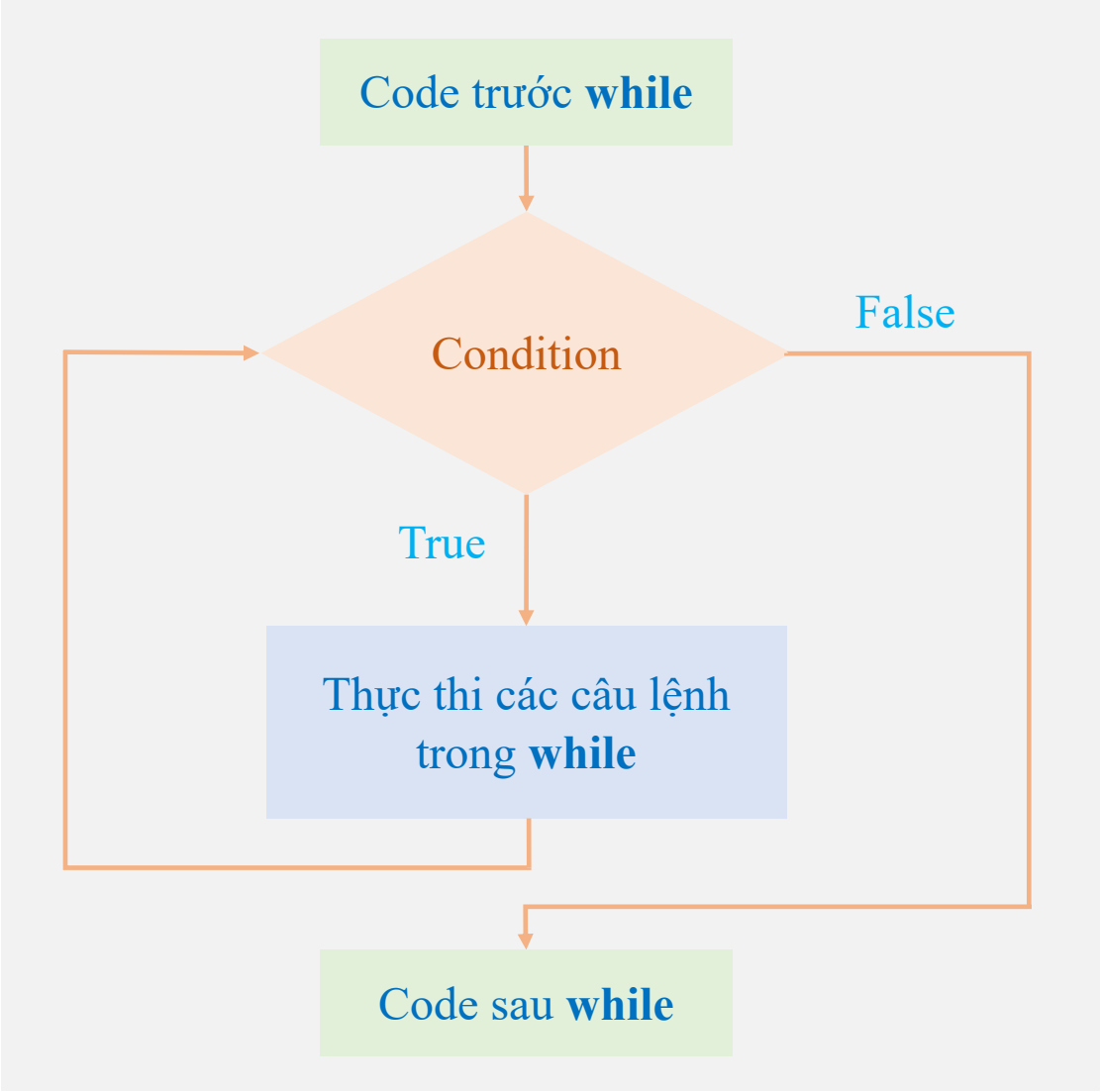
colon

```
# code trước while

while condition:
    # khối code trong while

# code sau while
```

indentation



While Loop

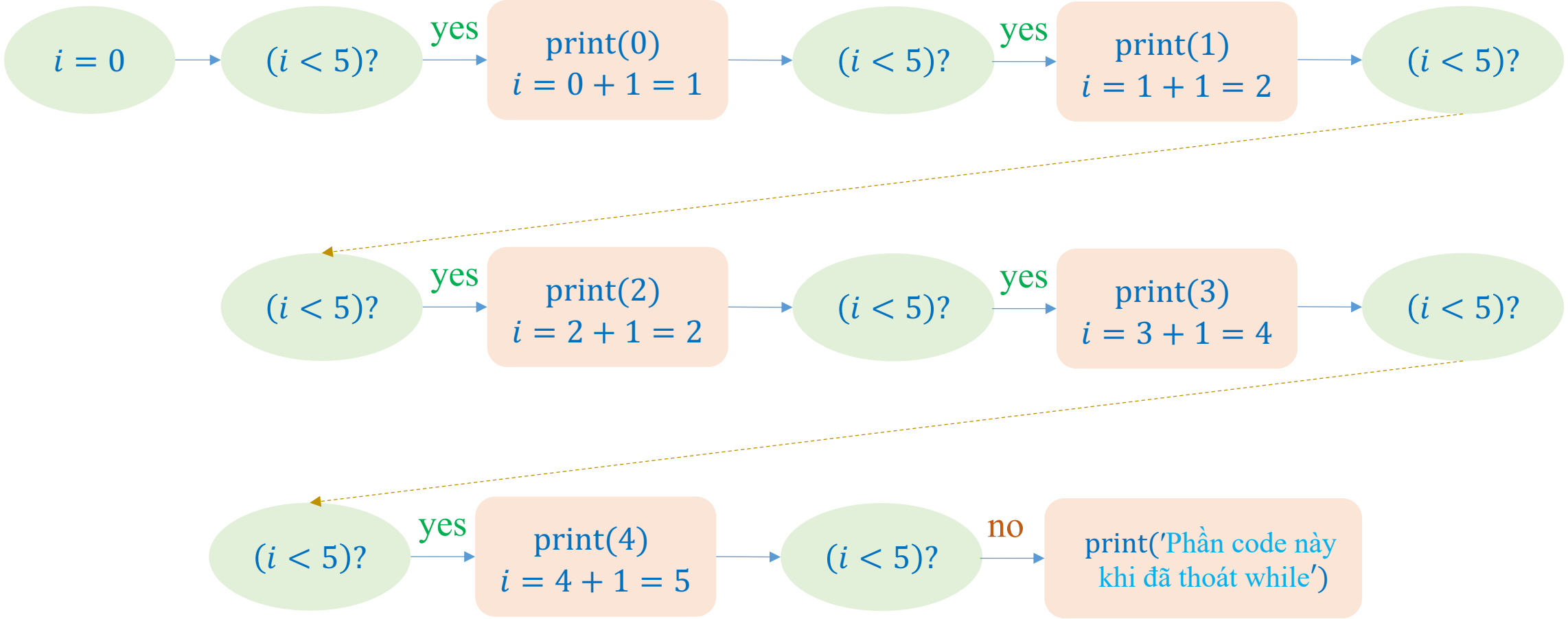



```python
# tạo biến i
i = 0

# bắt đầu vòng lặp while
while i<5:
    # code inside while
    print(i)
    i = i + 1

print('Phần code này khi đã thoát while')
```

```
0
1
2
3
4
Phần code này khi đã thoát while
```

While Loop

```python
# tạo biến i
i = 0

# bắt đầu vòng lặp while
while i<5:
    # code inside while
    print(i)
    i = i + 1

print('Phần code này khi đã thoát while')
```



While Loop

while-True-break

```python
import random

# cho vòng lặp chạy vô tận
while True:
    # sinh số ngẫu nhiêu
    num = random.randint(0,10)
    print('Số sinh ra có giá trị là', num)

    # kiểm tra num có bằng 5 hay không?
    if num == 5:
        # nếu có thì thoát khỏi while
        break;
print('Đã thoát khỏi while')
```

```
Số sinh ra có giá trị là 4
Số sinh ra có giá trị là 3
Số sinh ra có giá trị là 8
Số sinh ra có giá trị là 1
Số sinh ra có giá trị là 0
Số sinh ra có giá trị là 5
Đã thoát khỏi while
```

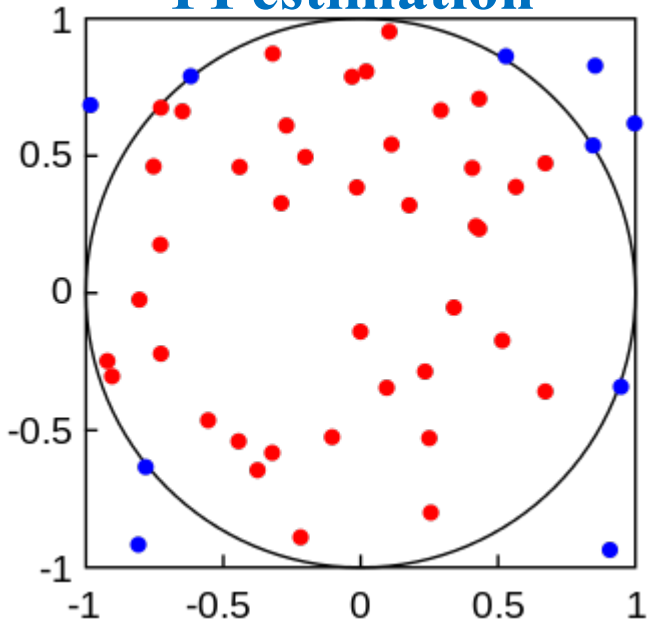
Exercises

E estimation

$$e \approx 1 + \frac{1}{1!} + \frac{1}{2!} + \ldots + \frac{1}{n!}$$

Simulation of coin tossing

PI estimation



Compute quadratic root for the number a

Newton Method

Given **a** and n = 0
Set a value for x_0 (x_0 = **a**/2)

$$x_{n+1} = \frac{x_n + \frac{a}{x_n}}{2}$$

n = n + 1


QUIZ TIME

Outline

SECTION 1

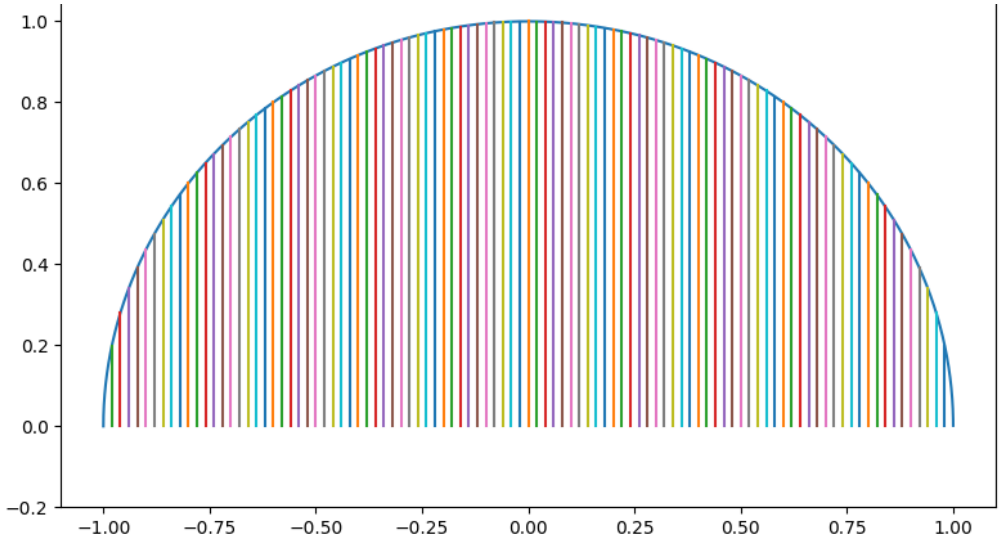
FOR Loop

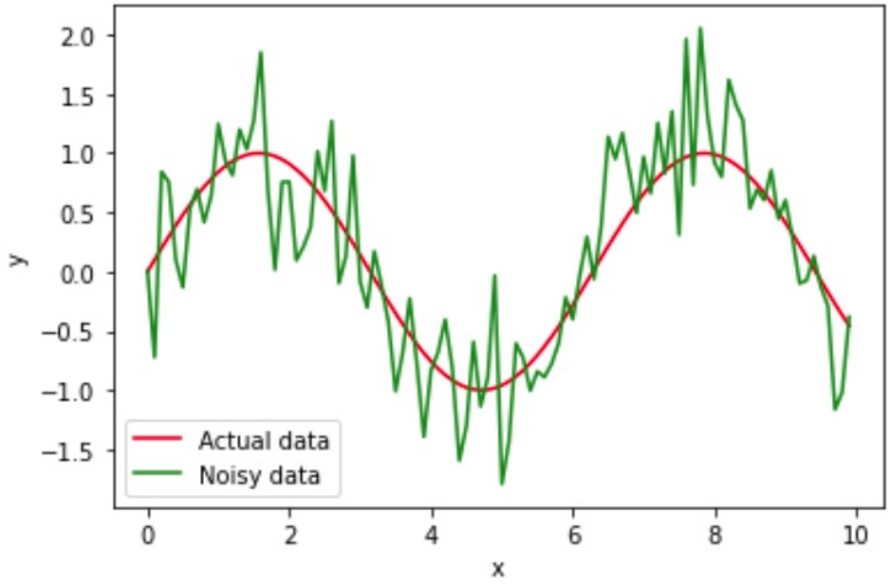
SECTION 2

WHILE Loop

SECTION 3

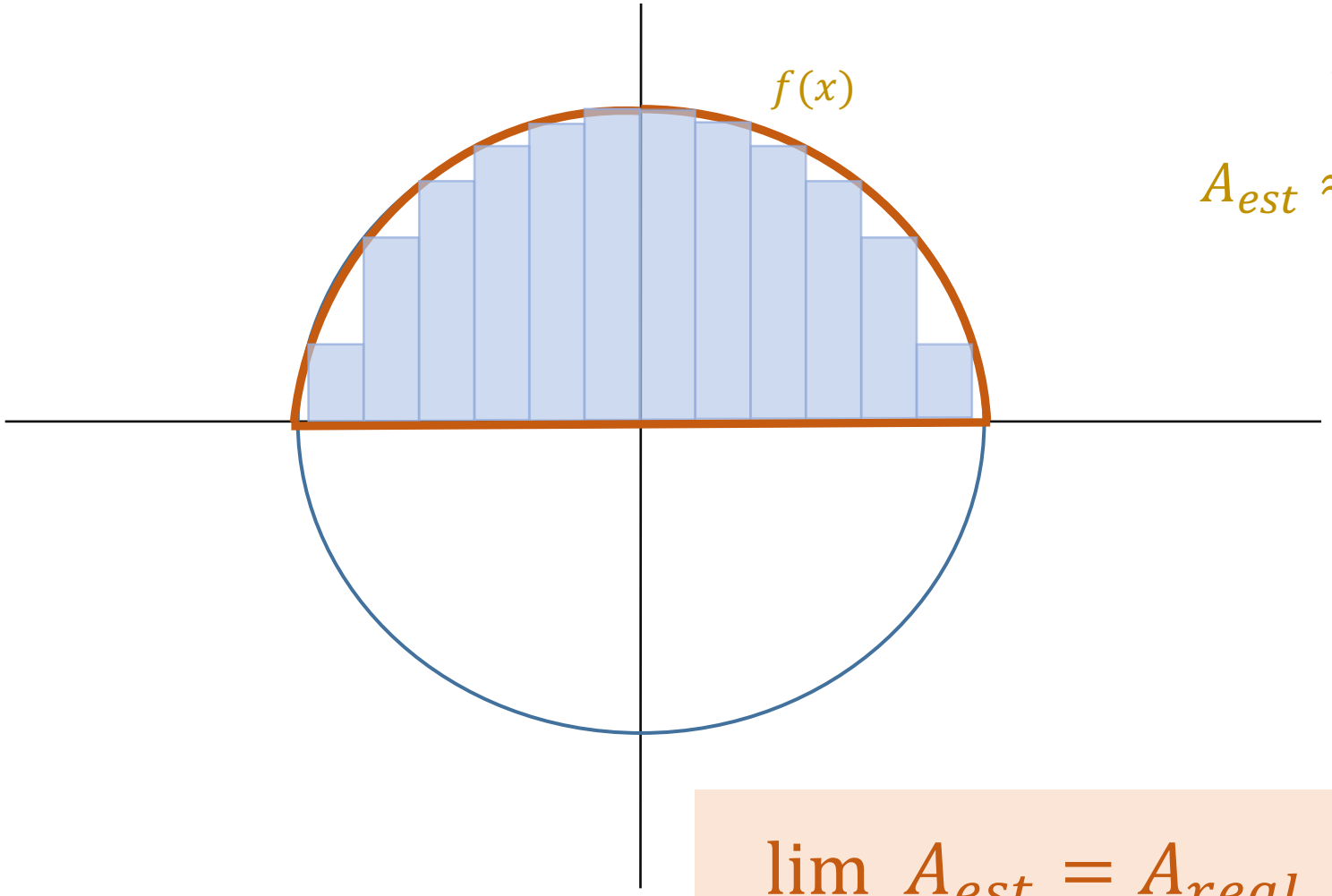
Examples





Example 1

❖ Compute the area of a unit circle



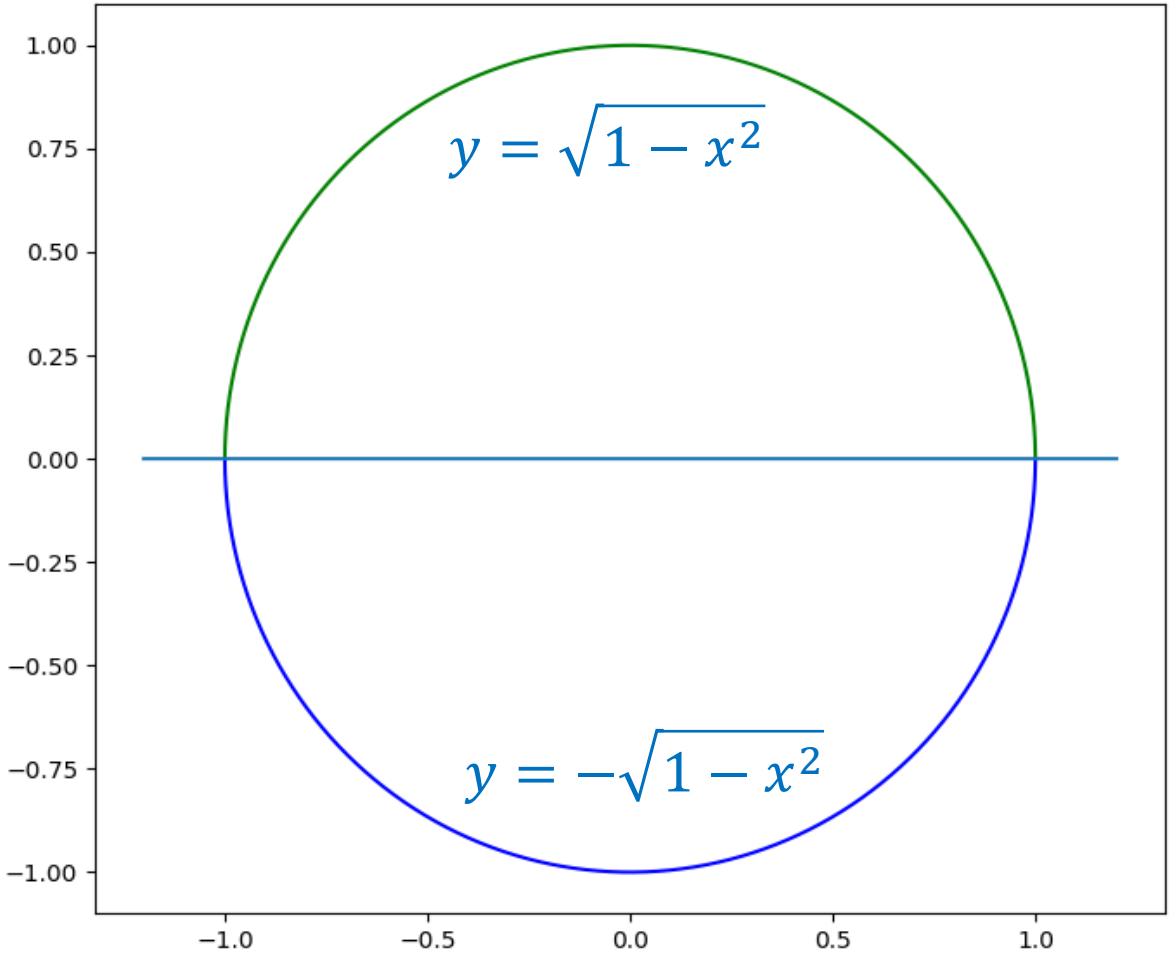
$$A_{est} \approx f(x_1)\Delta x_1 + f(x_2)\Delta x_2 + \dots + f(x_n)\Delta x_n$$

$$A_{est} \approx \sum_{i=1}^{n} f(x_i)\Delta x_i$$

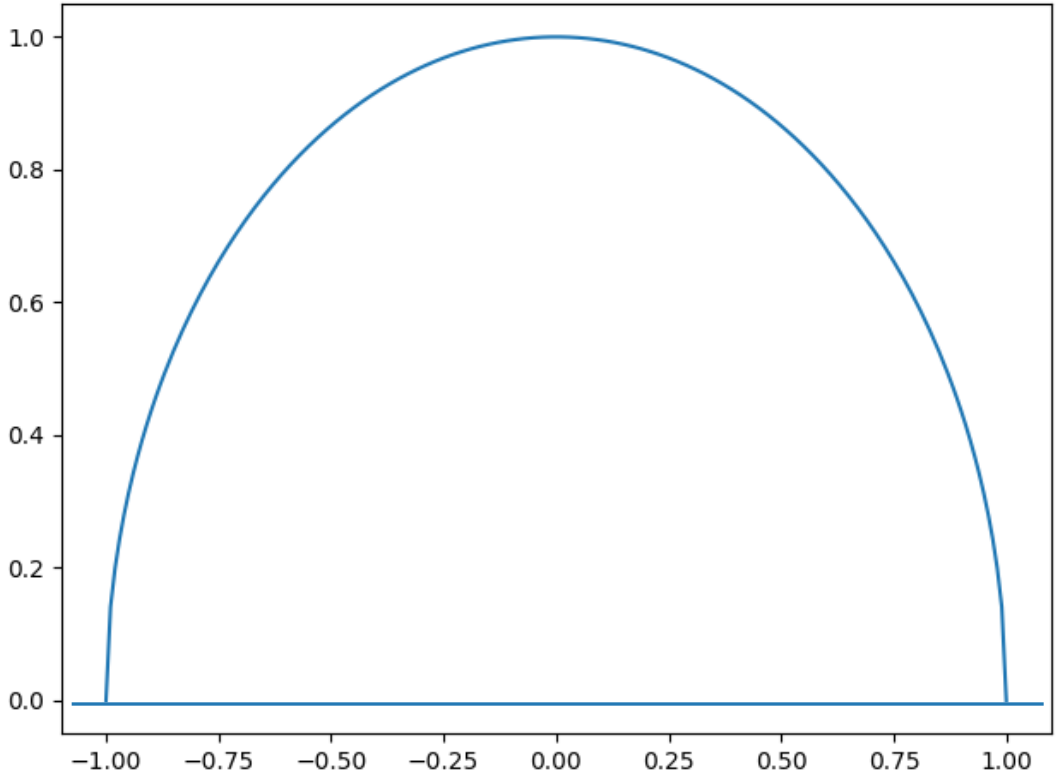
$$\lim_{\Delta x \to 0} A_{est} = A_{real}$$

Example 1

❖ Compute the area of a unit circle




```python
import math

def compute_y(x):
    return math.sqrt(1 - x*x)
```



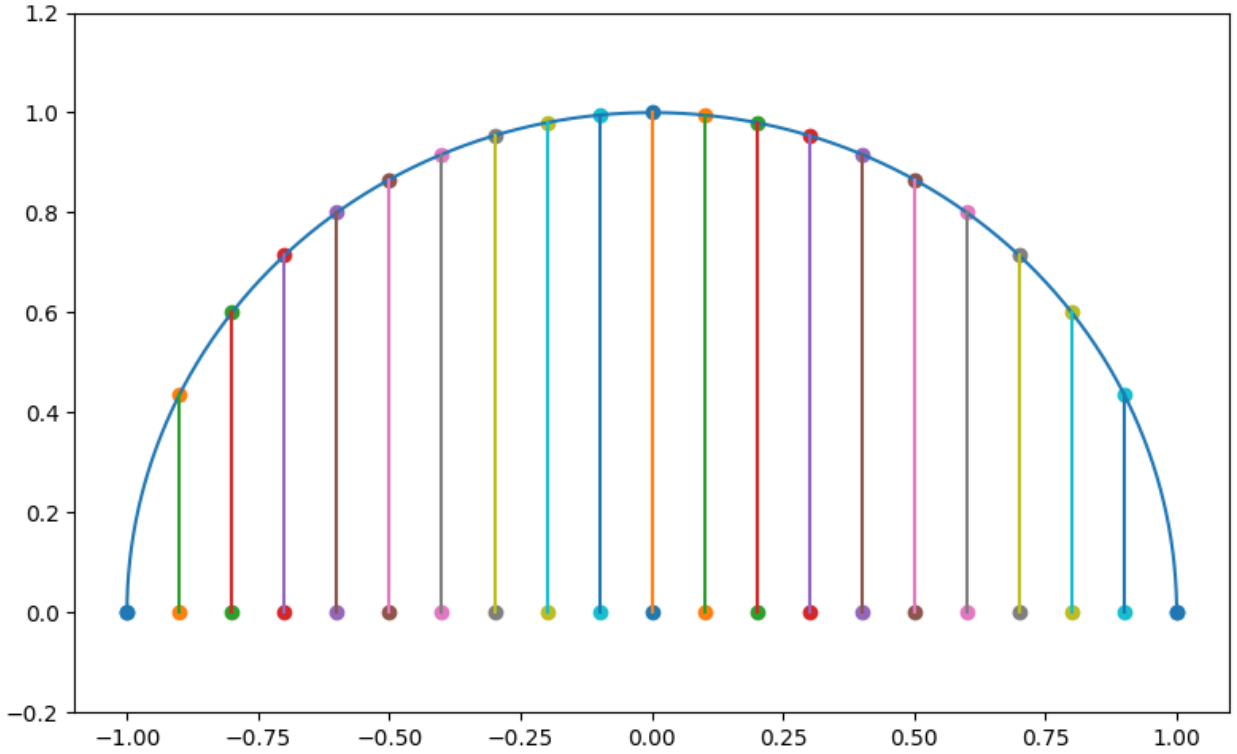
Example 1

❖ Compute the area of a unit circle

math.pi=3.141592

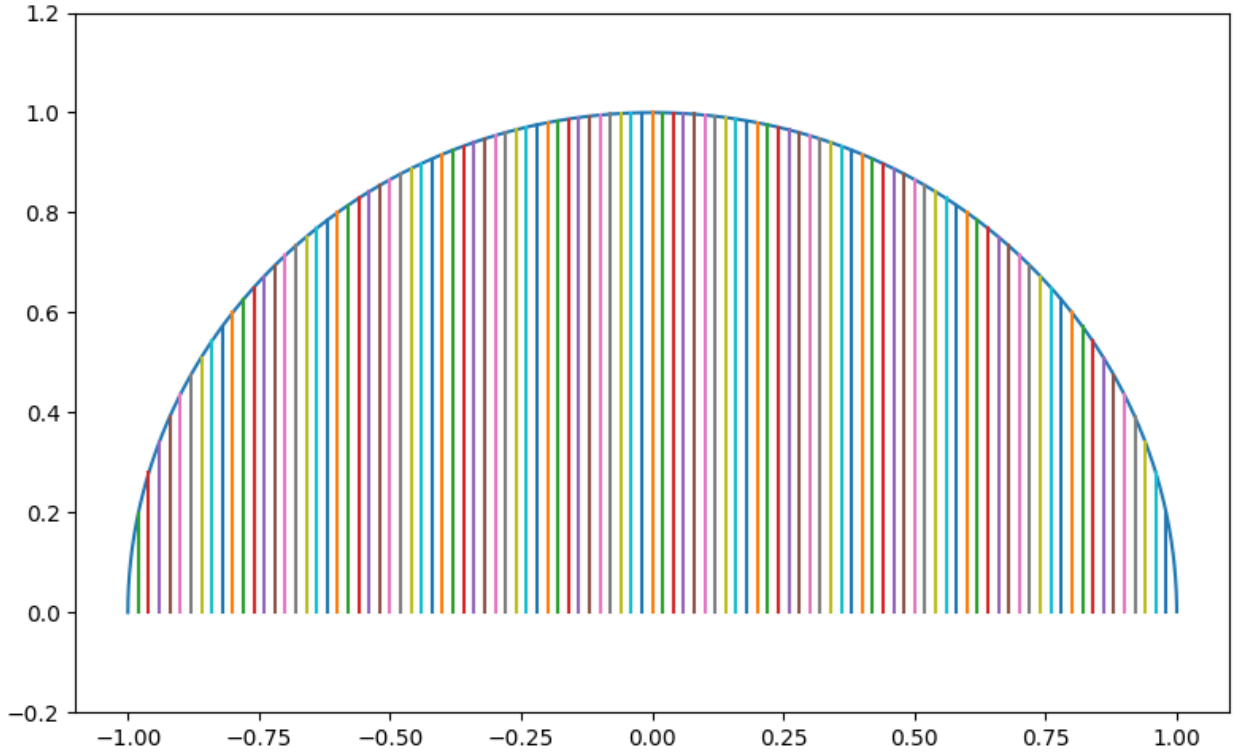
$n = 20$

$A_{est} = 3.1045$



$n = 200$

$A_{est} = 3.1404$



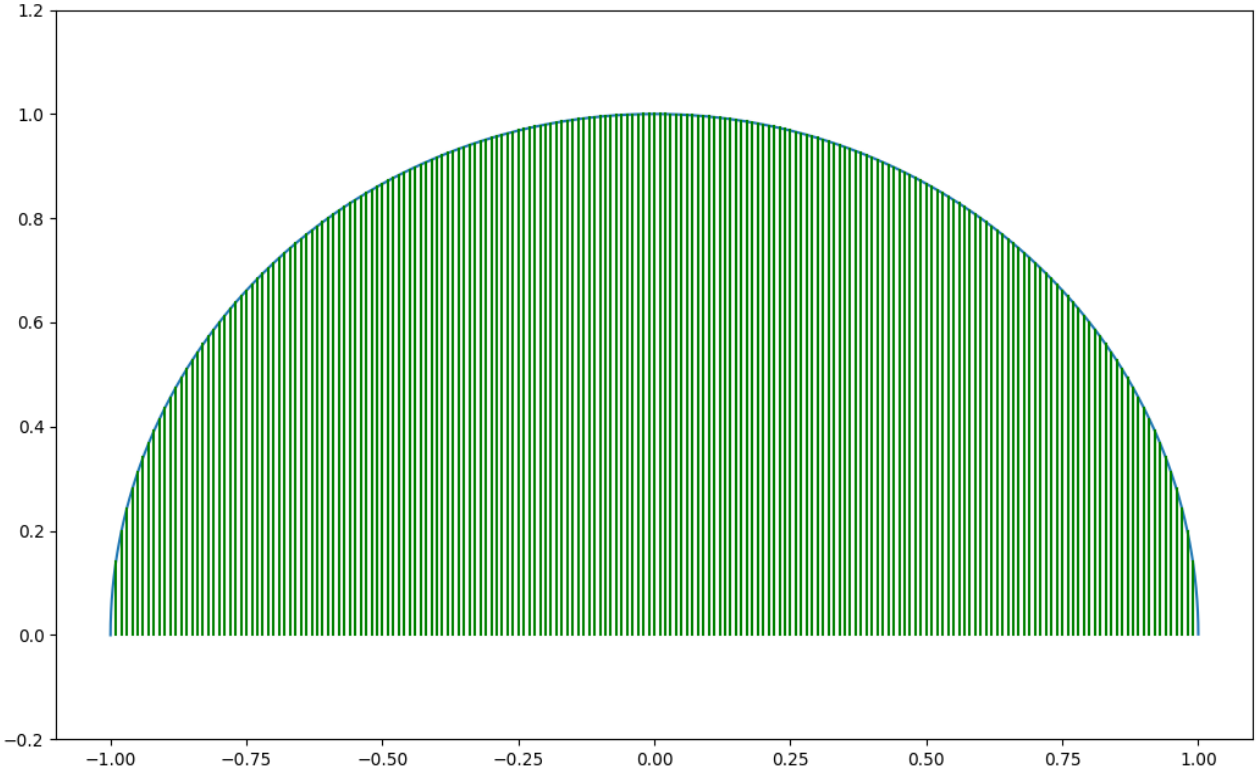
Example 1

❖ Compute the area of a unit circle

math.pi=3.141592

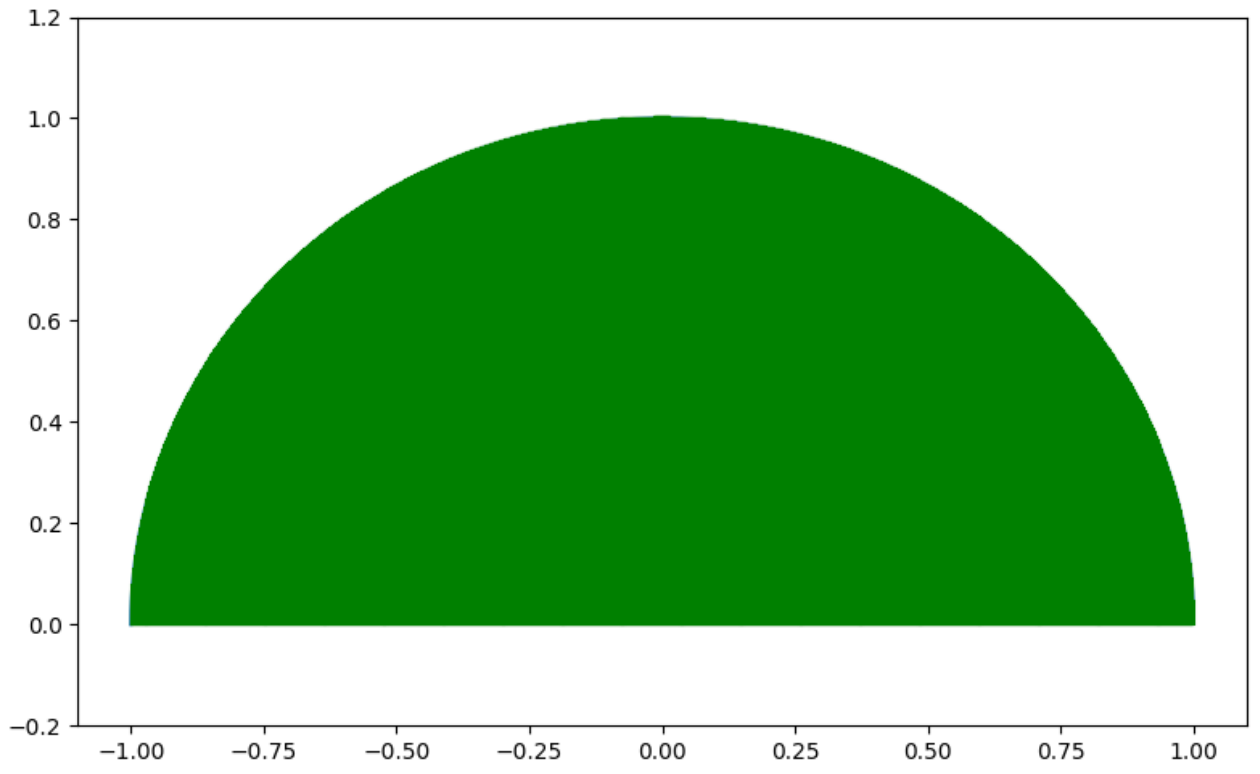
$n = 200$

$A_{est} = 3.1404$



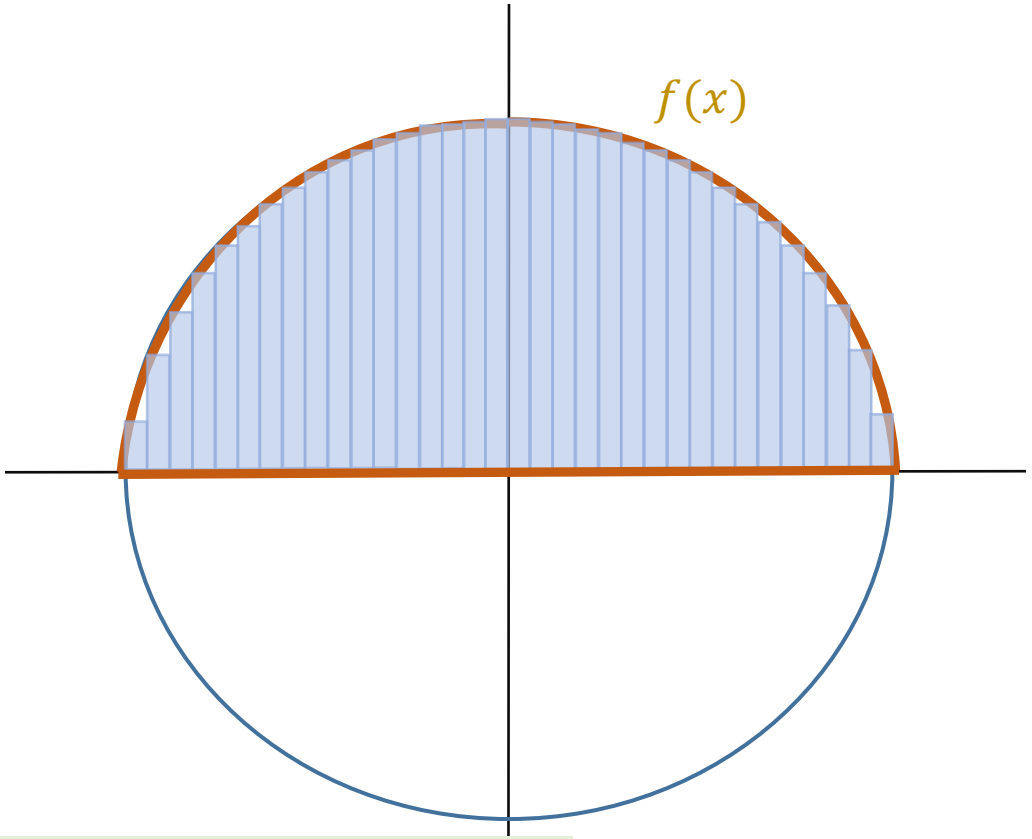
$n = 2000$

$A_{est} = 3.14155$



Example 1

❖ Compute the area of a unit circle



$$A_{est} \approx \sum_{i=1}^{n} f(x_i)\Delta x_i$$

```python
import math

def compute_y(x):
    return math.sqrt(1 - x*x)
```

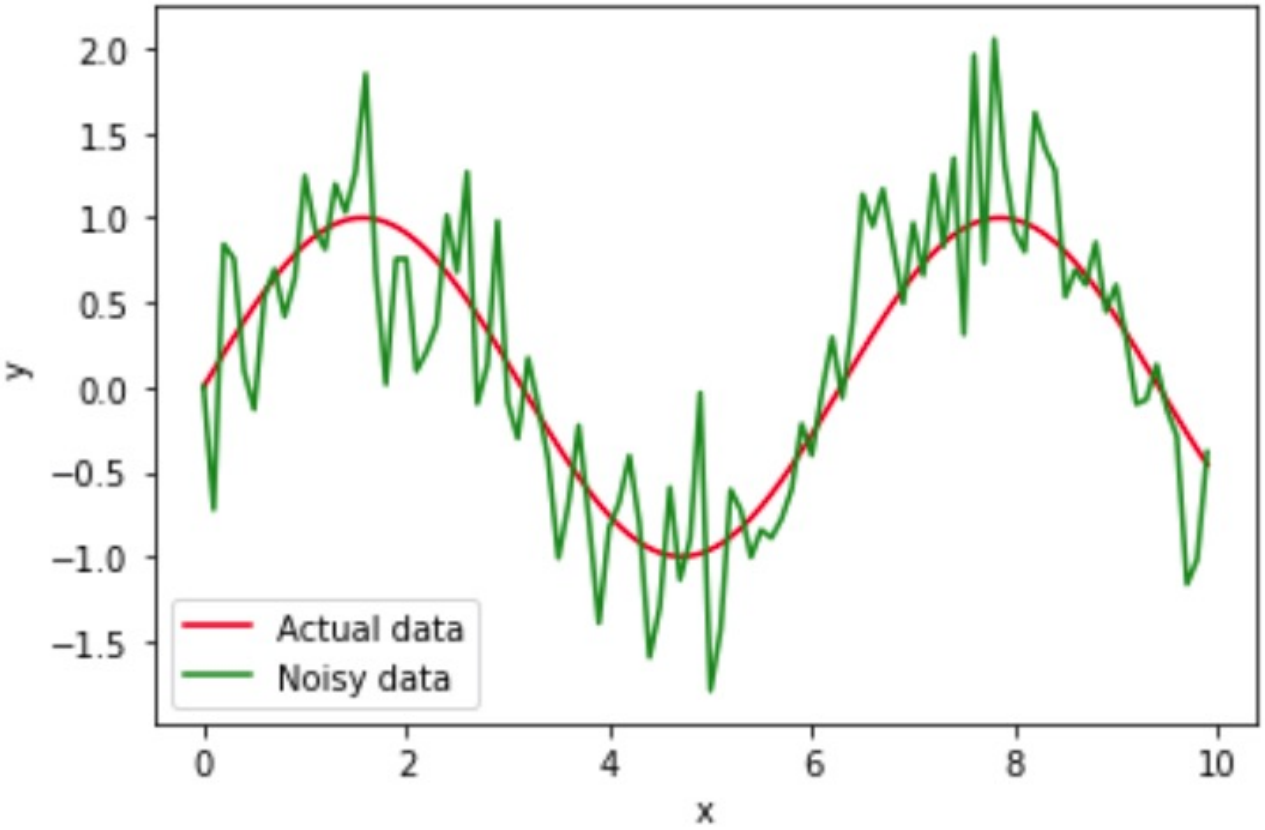
```python
delta_x = 0.01
n = int(2 / delta_x)

x = -1.0
half_area = 0.0
for _ in range(n):
    y = compute_y(x)
    half_area = half_area + delta_x*y
    x = x + delta_x
```

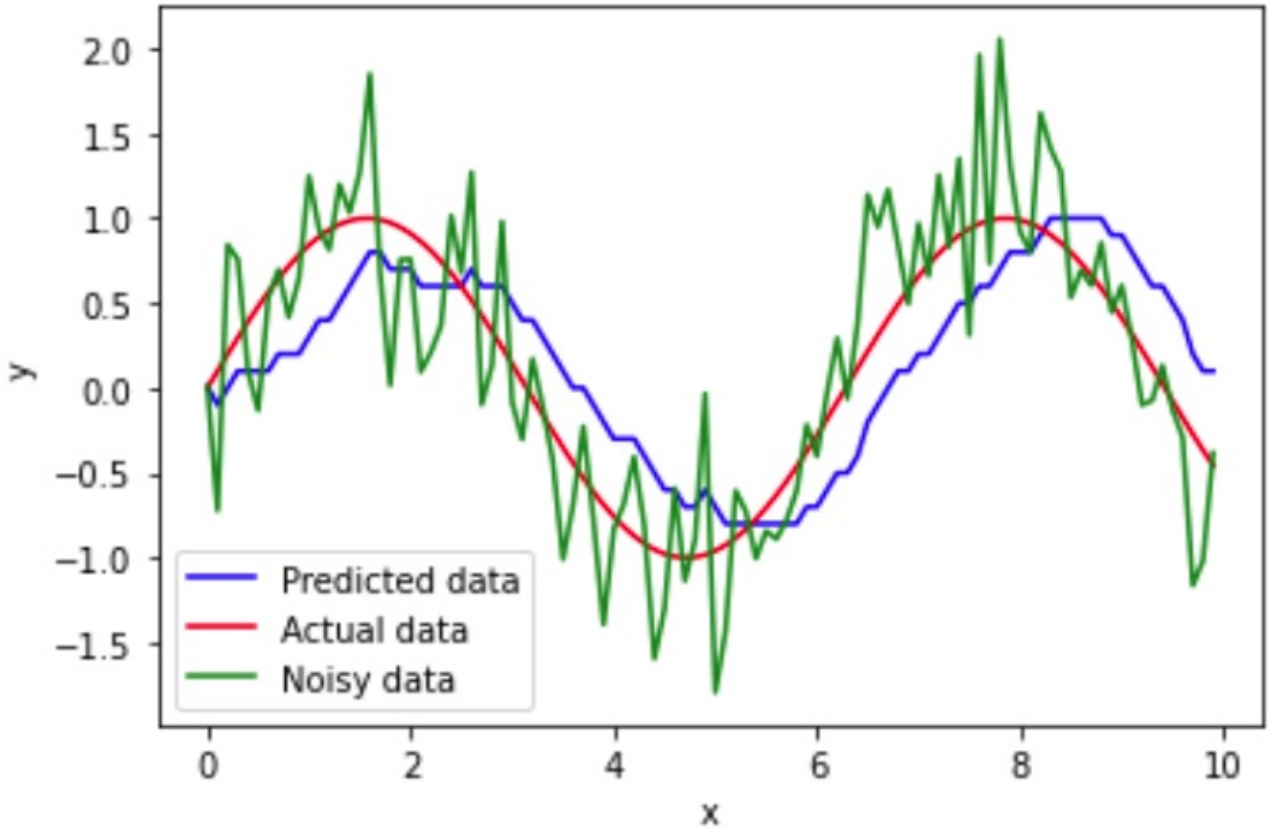
```python
print(half_area*2)
```

```
3.1404170317790423
```

Example 2

❖ Context





Example 2

❖ Moving average

$k = 2$

3	8	6	5	1	7	9	0	8	4
5.5	7.0	5.5	3.0	4.0	8.0	4.5	4.0	6.0	

$$SMA_t = \frac{s_{t-1} + s_{t-2} + \cdots + s_{t-k}}{k}$$

3	8	6	5	1	7	9	0	8	4
3.0	5.5	5.8	5.4	3.2	5.1	7.0	3.5	5.8	4.9

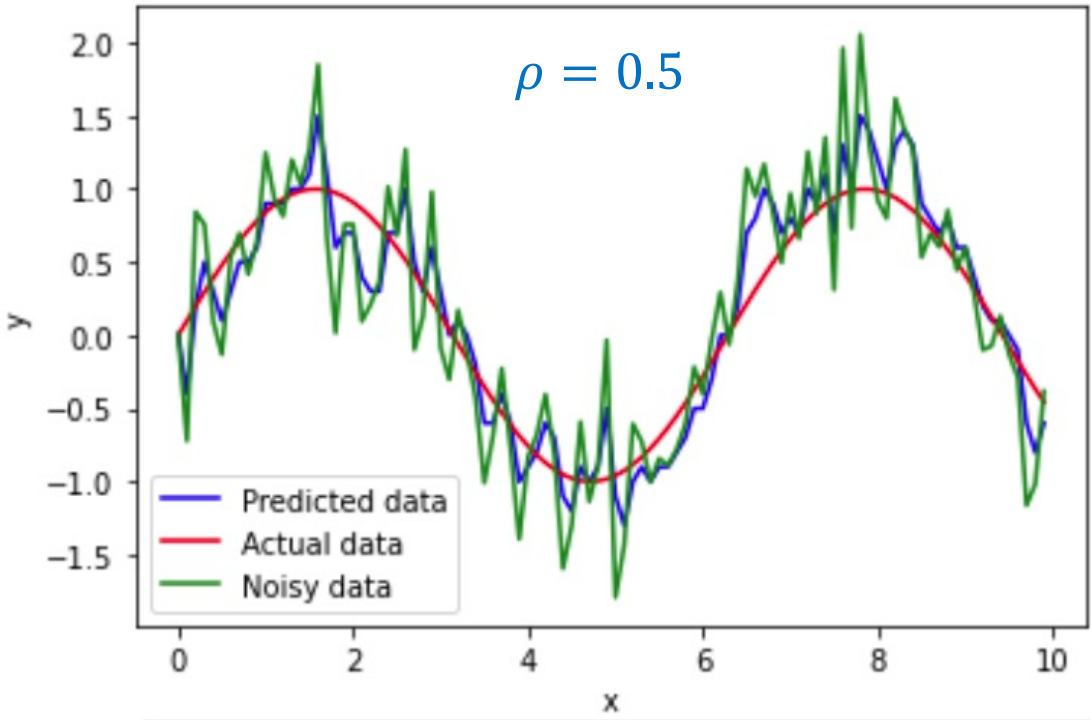
$$EMA_t = \rho EMA_{t-1} + (1-\rho)s_t$$

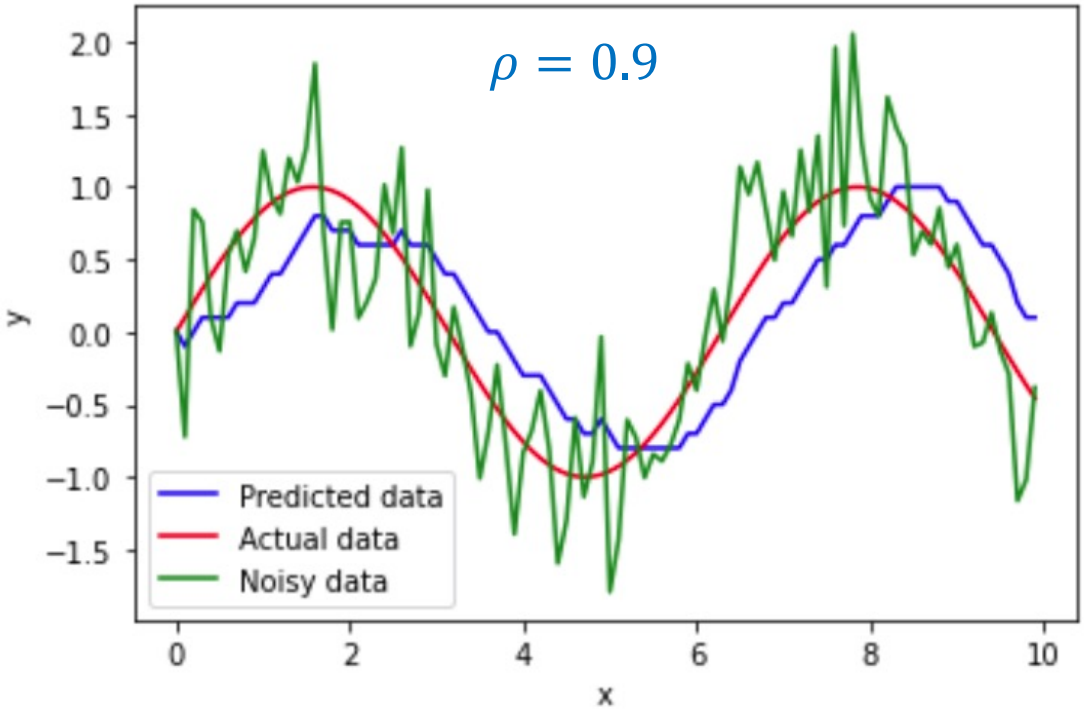
$\rho = 0.5$

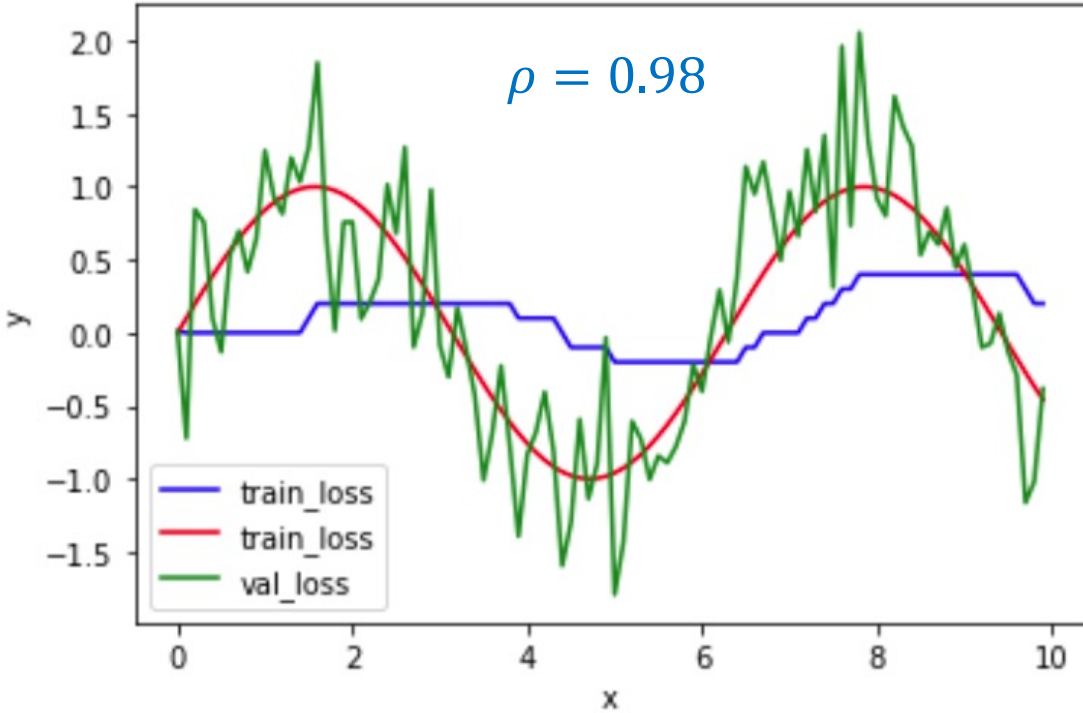
Example 2

❖ **Exponentially weighted averages**

$$V_t = \rho V_{t-1} + (1-\rho)s_t$$

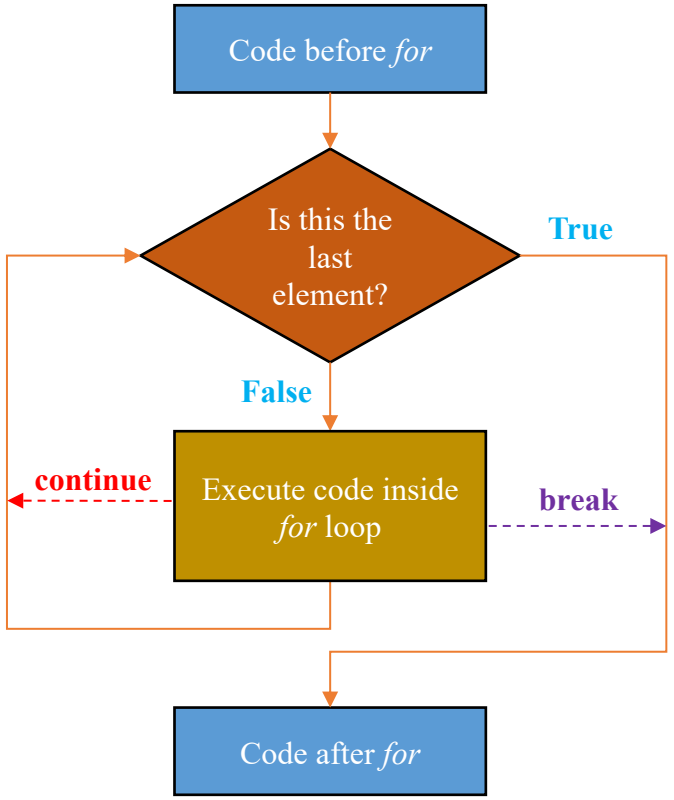




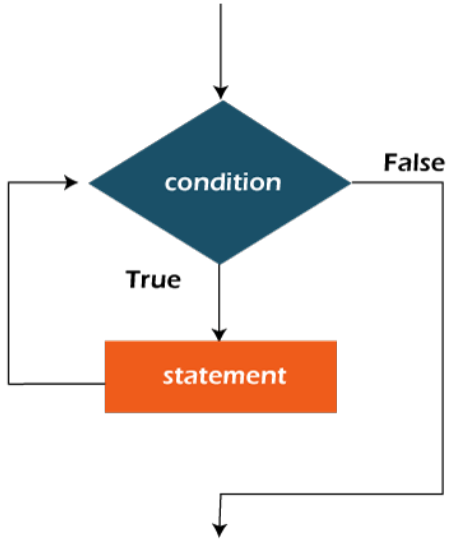


Summary

FOR Loop



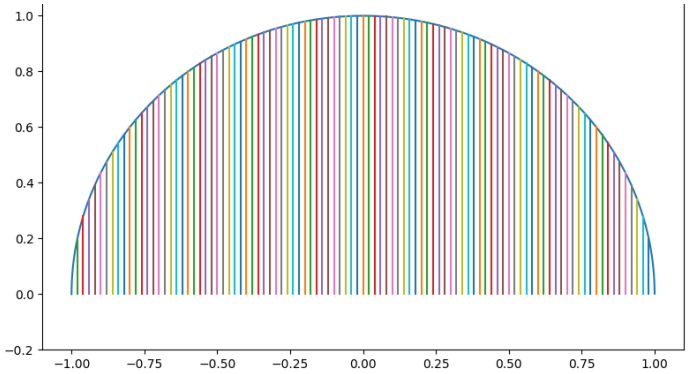
WHILE Loop

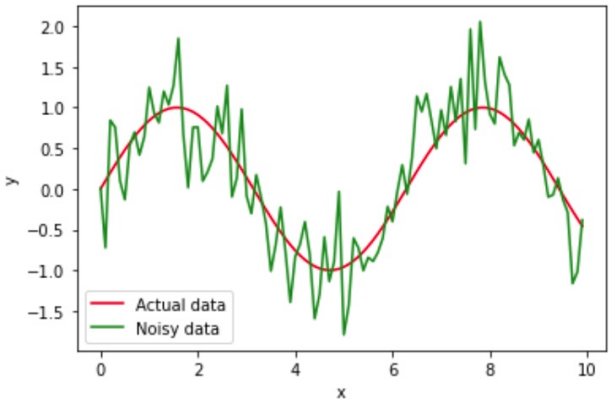



```
# …
while condition:
  # code inside while
# …
```

```
True/False ← condition
```

Examples





AI VIET NAM
All-in-One 2025

Cheat Sheet – For Loop

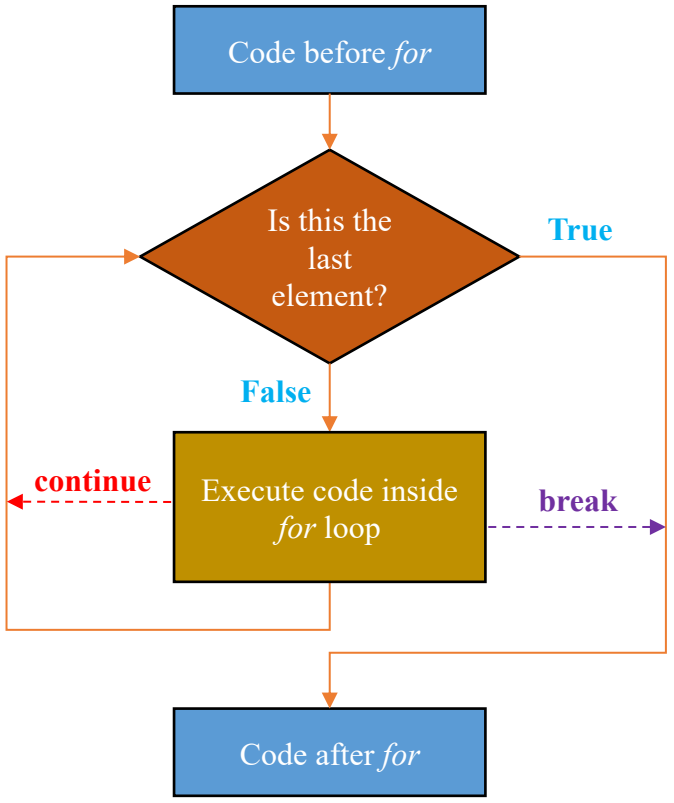
for syntax

indentation

```
# code before for
for element in iterable:
    # code inside for
# code after for
```

_ **Definition:**

+ *for, in:* python ***keywords***
+ *element*: iterable ***element***
+ ***iterable****:* range(), list, string, tuple, and dictionary
+ ***colon***: "**:**"



Common Iterables

_ **String:**

```
greeting = 'Hello AIVIETNAM'
for character in greeting:
    print(character)
```

_ **List:**

```
odds = [1, 3, 5, 7]
for odd in odds:
    print(odd)
```

_ **Tuple:**

```
fruits = ('apple', 'banana'
'melon', 'peach')
for fruit in fruits:
    print(fruit)
```

_ **Dictionary:**

```
parameters = {'lr': 0.1,
    'optimizer': 'Adam',
    'metric': 'Accuracy'}
for key in parameters:
    print(key,
          parameters(key))
```

_ **range(start, end, step):**

```
range(start=0, end=5, step=1) ~ range(5)
```

```
[0, 1, 2, 3, 4]
```

```
# usage of range()
# just like using a list
for i in range(5):
    print(i)
```

Special keywords

_ ***continue*:**

```
for i in range(10):
    if i == 5:
        # code after continue
          will not be executed
        continue

    print(i)
#output: 0,1,2,3,4,6,7,8,9
```

_ ***break*:**

```
for i in range(10):
    if i == 5:
        # if true then the
          loop will be end
        break

    print(i)
#output: 0,1,2,3,4
```

for loop applications

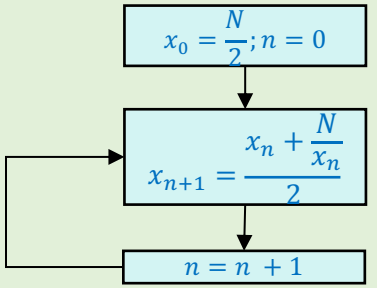
Coin tossing

$$P(event) = \frac{|event|}{|S|}$$

Euler's number

$$e \approx \left(1 + \frac{1}{n}\right)^n \qquad \lim_{n\to\infty}\left(1 + \frac{1}{n}\right)^n = e$$

Quadratic Root



PI estimation

_ **Monte Carlo Method:**

$$\pi \approx \frac{s^2 N_C}{N_S}$$

_ **Gregory-Leibniz Series:**

$$\pi \approx 4\sum_{i=1}^{n}\frac{(-1)^{i+1}}{2i-1}$$

_ **Nilakantha Series:**

$$\pi \approx 3 + 4\sum_{i=0}^{n}\frac{-1^i}{(2i+2)(2i+3)(2i+4)}$$

AI VIET NAM
All-in-One 2025

Cheat Sheet 2

Random & Math module

_ **Math module's common methods and constants:**

Definition	Syntax	Definition	Syntax
Absolute	`math.fabs(n)`	Factorial	`math.factorial()`
Logarith	`math.log(n)`	*Rounding 1*	`math.round()`
Sine	`math.sine(n)`	*Rounding 2*	`math.ceil()`
Cosine	`math.cosine(n)`	*Rounding 3*	`math.floor()`
Exponential	`math.exp(n)`	Euler (e)	`math.e`
Square root	`math.sqrt(n)`	PI (π)	`math.pi`

_ **Random module:**

+ Generate random floating-point in [0, 1): `random.random()`
+ Generate random integer in [a, b]: `random.randint(a, b)`

Random/Loop Examples

Coin tossing

$$P(event) = \frac{|event|}{|S|}$$

Euler's number

$$e \approx \left(1 + \frac{1}{n}\right)^n$$

Quadratic Root

$$x_0 = \frac{a}{2};\ i = 0 \rightarrow n\_loops;\ x_{n+1} = \frac{x_n + \frac{a}{x_n}}{2}$$

PI estimation

_ **Monte Carlo Method:**

$$\pi \approx \frac{s^2 N_C}{N_S}$$

_ **Gregory-Leibniz Series:**

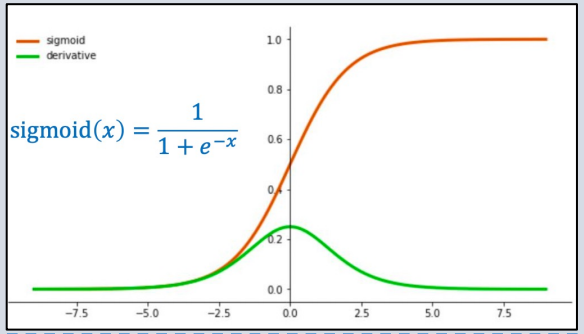
$$\pi \approx 4 \sum_{i=1}^{n} \frac{(-1)^{i+1}}{2i-1}$$

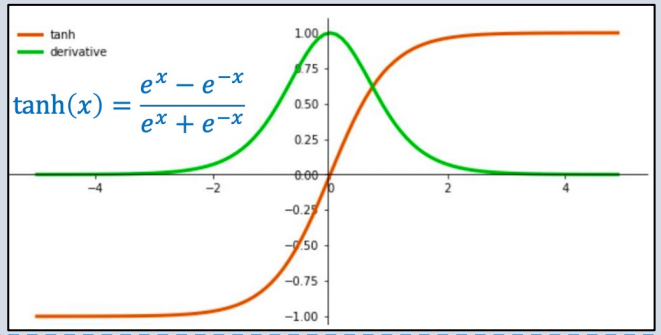
_ **Nilakantha Series:**

$$\pi \approx 3 + 4 \sum_{i=0}^{n} \frac{-1^i}{(2i+2)(2i+3)(2i+4)}$$

Activation Functions

Map x values into smaller ranges

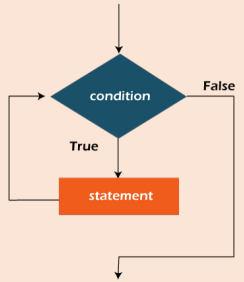




While Loop

```
# …
while condition:
  # code inside while
# …
```

True/False ← condition



_ **while condition:**

```
i = 0
while i < 5:
  print(i)
  i = i + 1
print("done")
```

_ **while-True-break:**

```
i = 0
while True:
  print(i)
  i = i + 1
  if i == 5:
    break
print("done")
```

Common Errors

_ **NameError:**

```
a = 5
c = a + b
print(c) # b not defined
Print(a) # Print not defined
```

_ **ValueError:**

```
print(int("aivietnam"))
```

_ **RecursionError:**

```
def a_func(n):
  return a_func(n)
a_func(5) # infinite calls
```

_ **SyntaxError:**

```
print('aivietnam")
```

_ **ZeroDivisionError:**

```
print(5 / 0)
```

_ **TypeError:**

```
print(5 + "aivietnam")
```

_ **IndetationError:**

```
a = 1
 b = 2 # identation
print(a + b)
```

_ **ModuleNotFoundError:**

```
import mymodule
```

_ **IndexError:**

```
print("aivietnam"[50])
```