AI VIET NAM
@aivietnam.edu.vn

Basic Python for AI

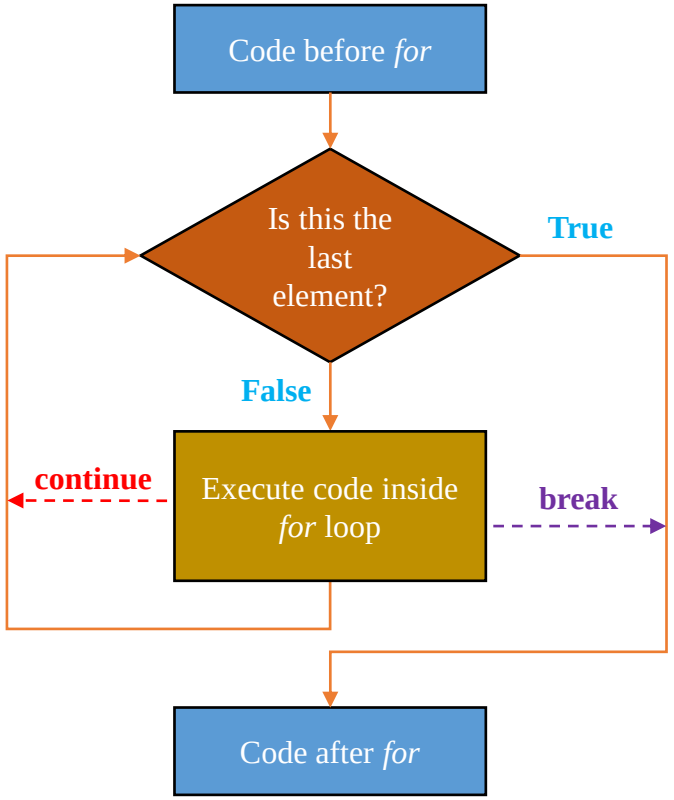
Loops and Files

Quang-Vinh Dinh
PhD in Computer Science

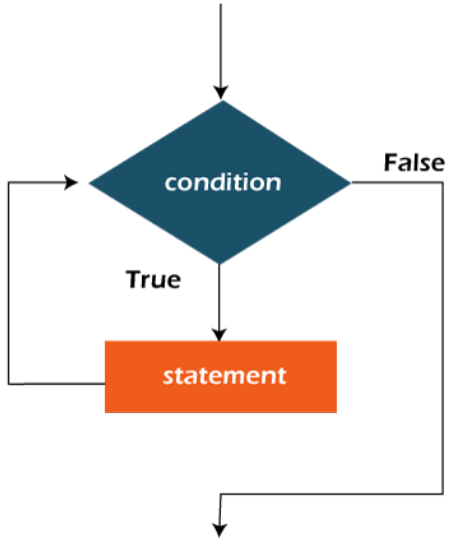
Friday (07/06/2024)

Objectives

FOR Loop



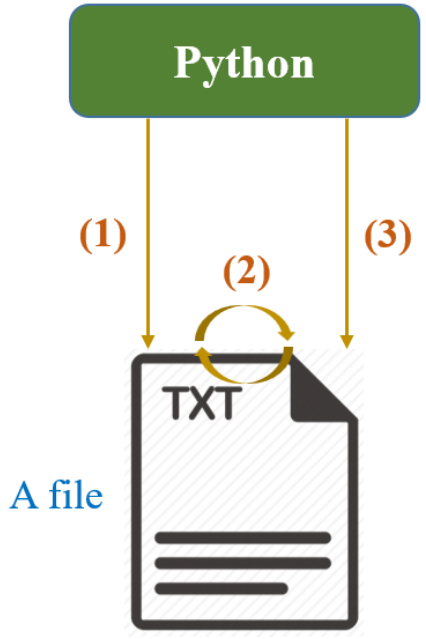
WHILE Loop




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

`True/False ← condition`

Files



(1)	Connect to file
(2)	Read from/write to file
(3)	Disconnect to file

Outline

SECTION 1

FOR Loop

SECTION 2

WHILE Loop

SECTION 3

Files

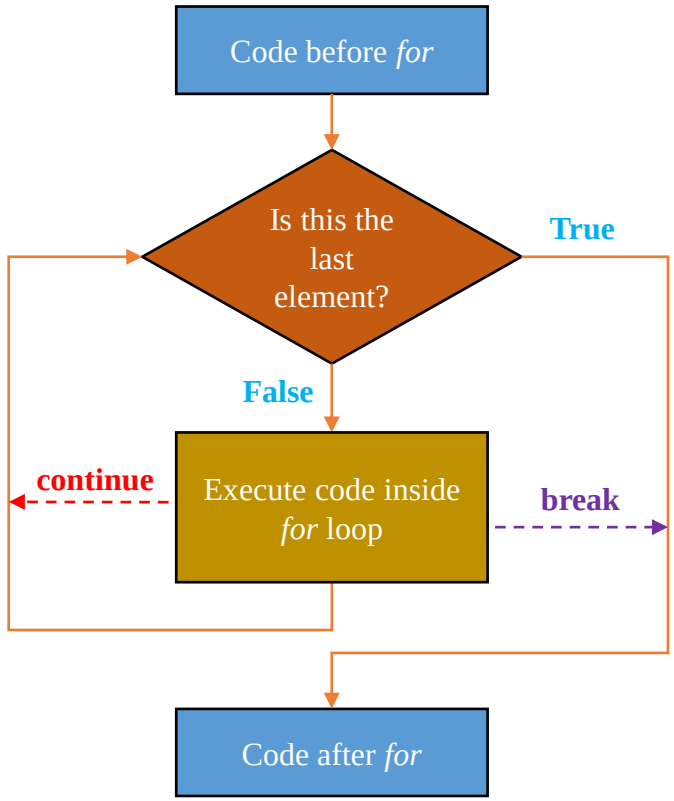
SECTION 4

Examples

for syntax

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

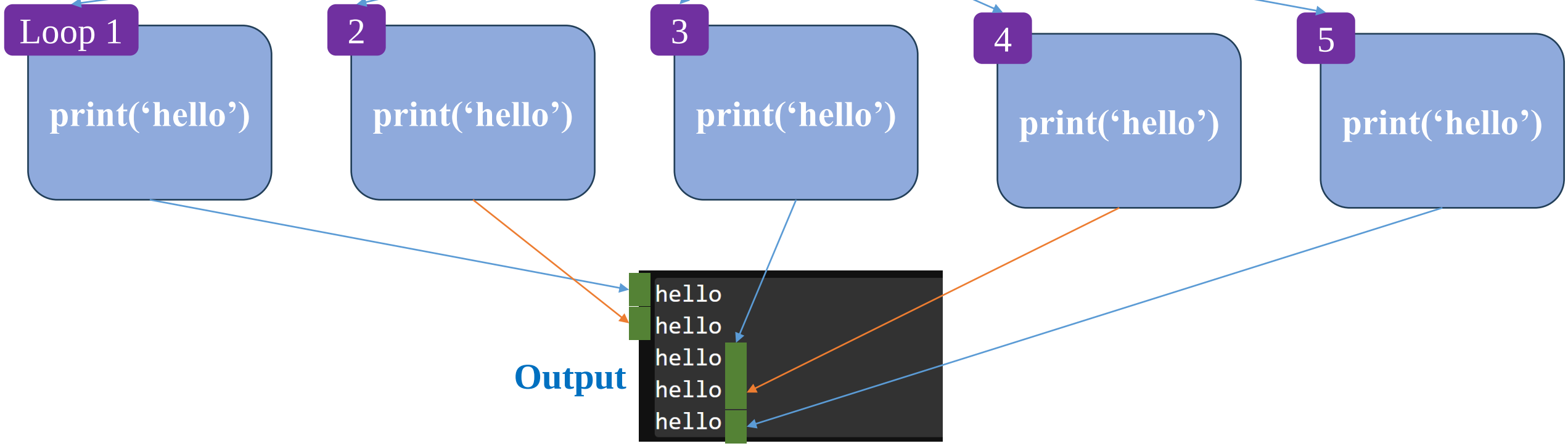
indentation



For Loop

❖ Repeat action in Python

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



For Loop

keyword

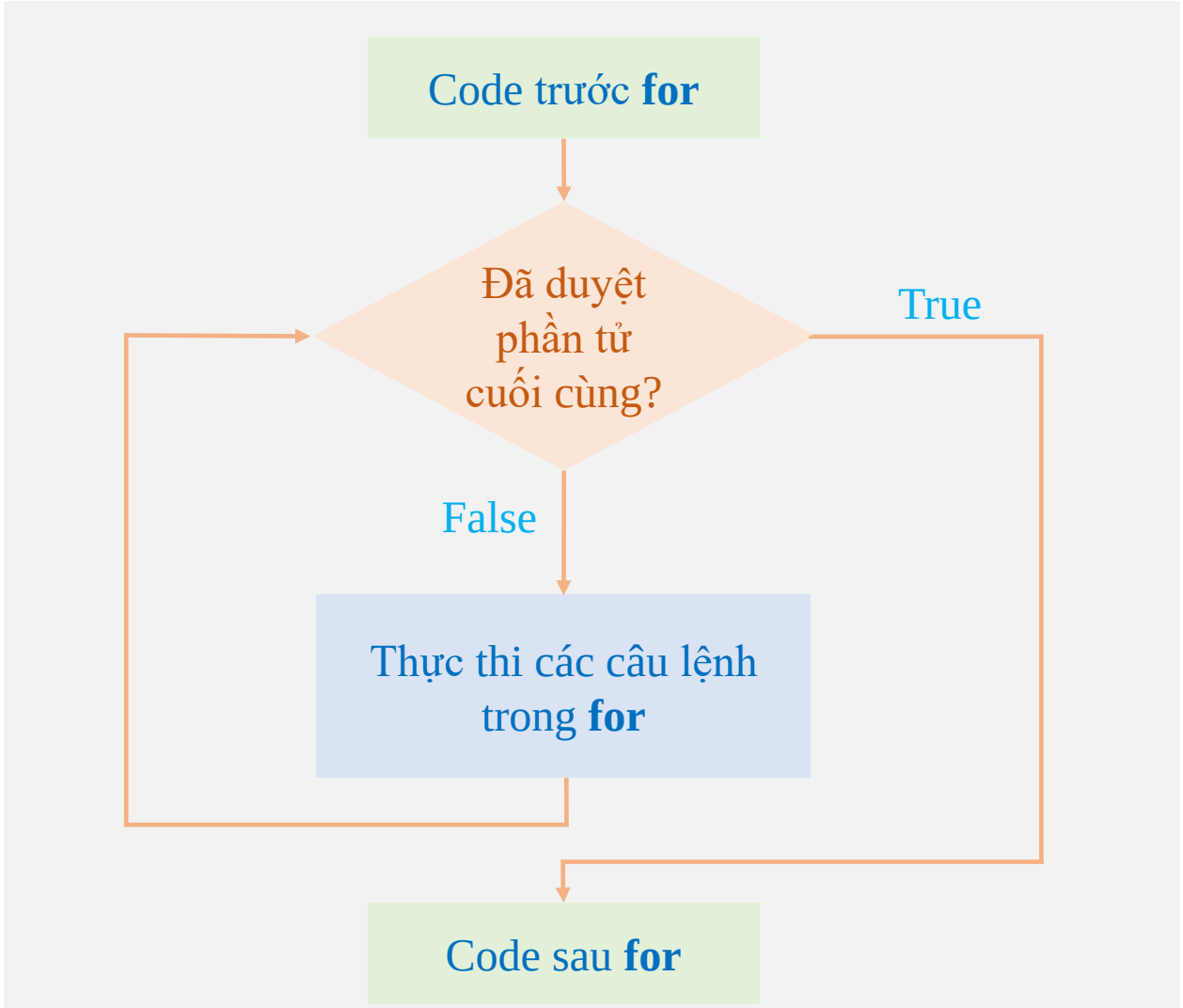
colon

```
# code trước for

for element in iterable:
    # code trong for

# code sau for
```

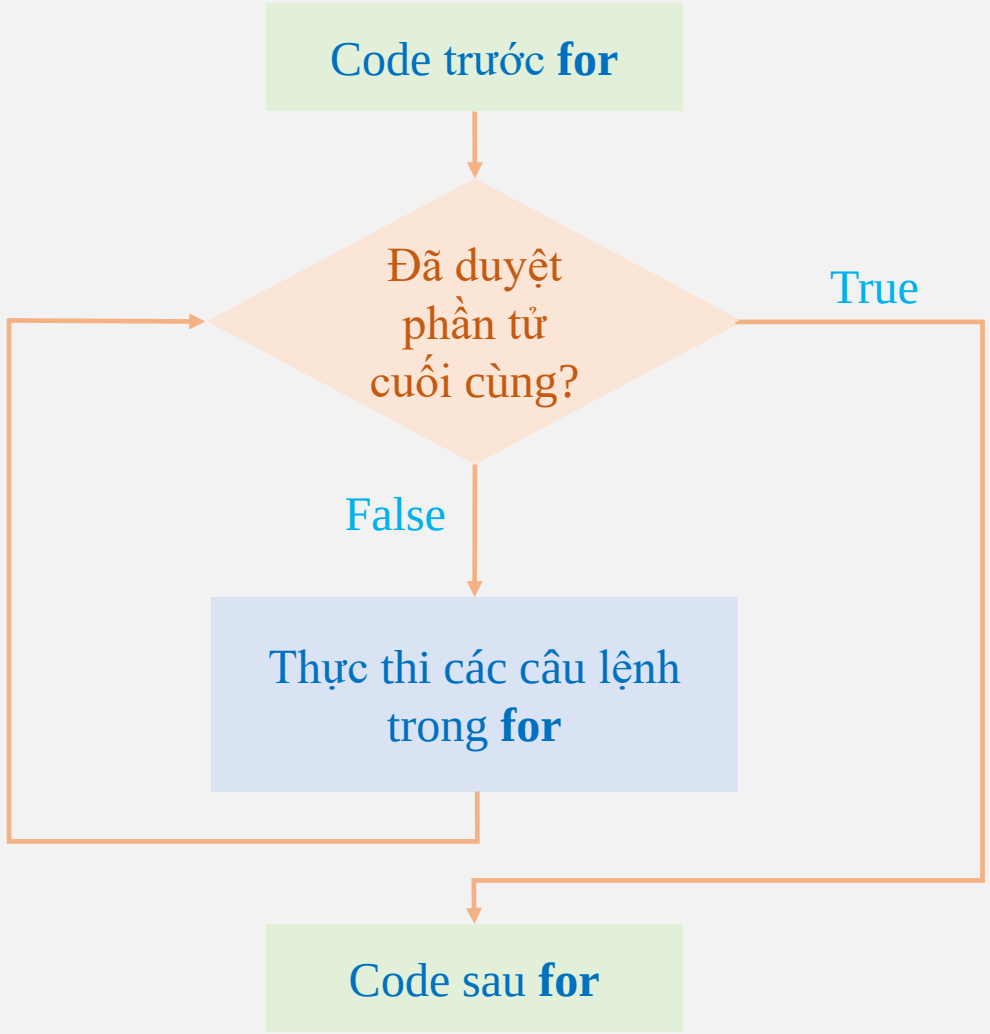
indentation

Iterables

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



For Loop



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

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

⬇

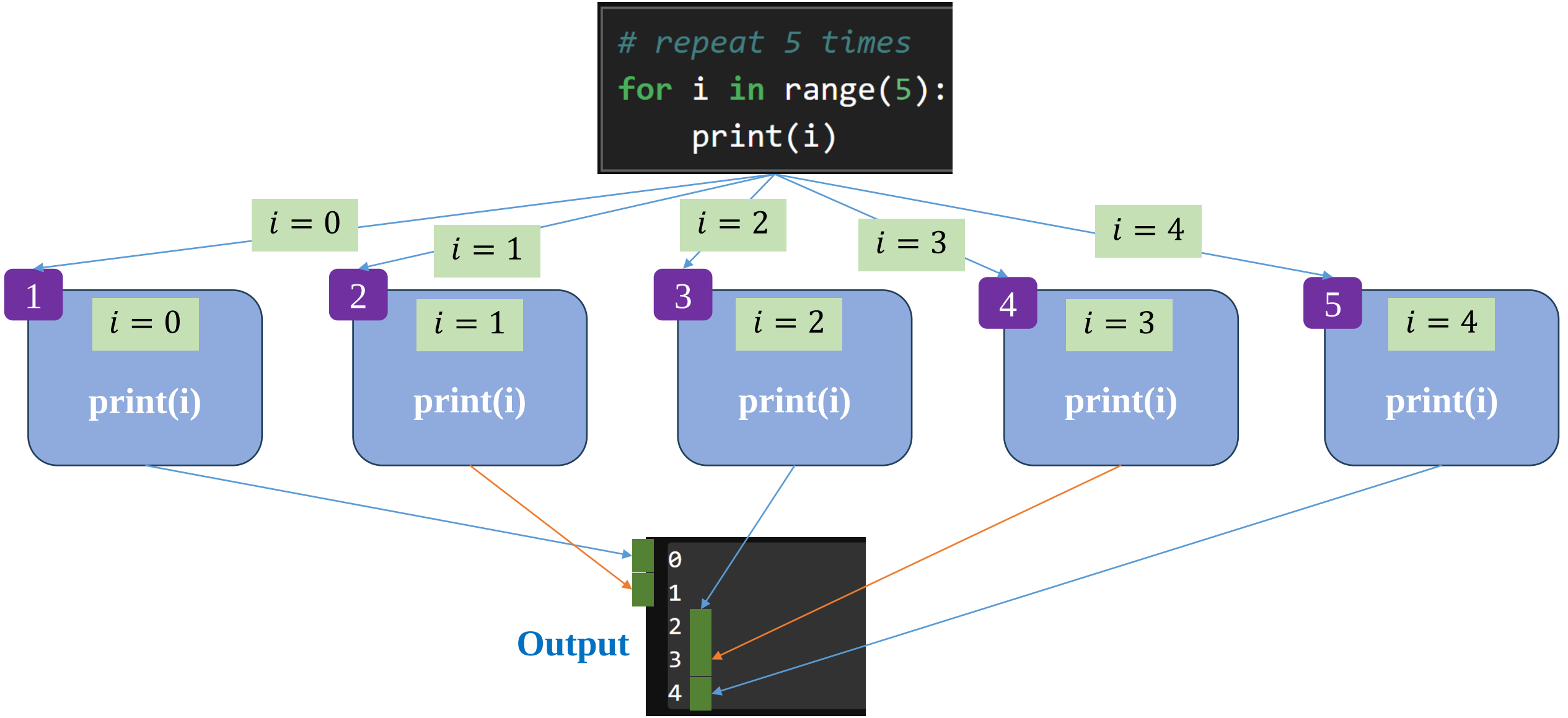
0, 1, 2, 3, 4

range(5)

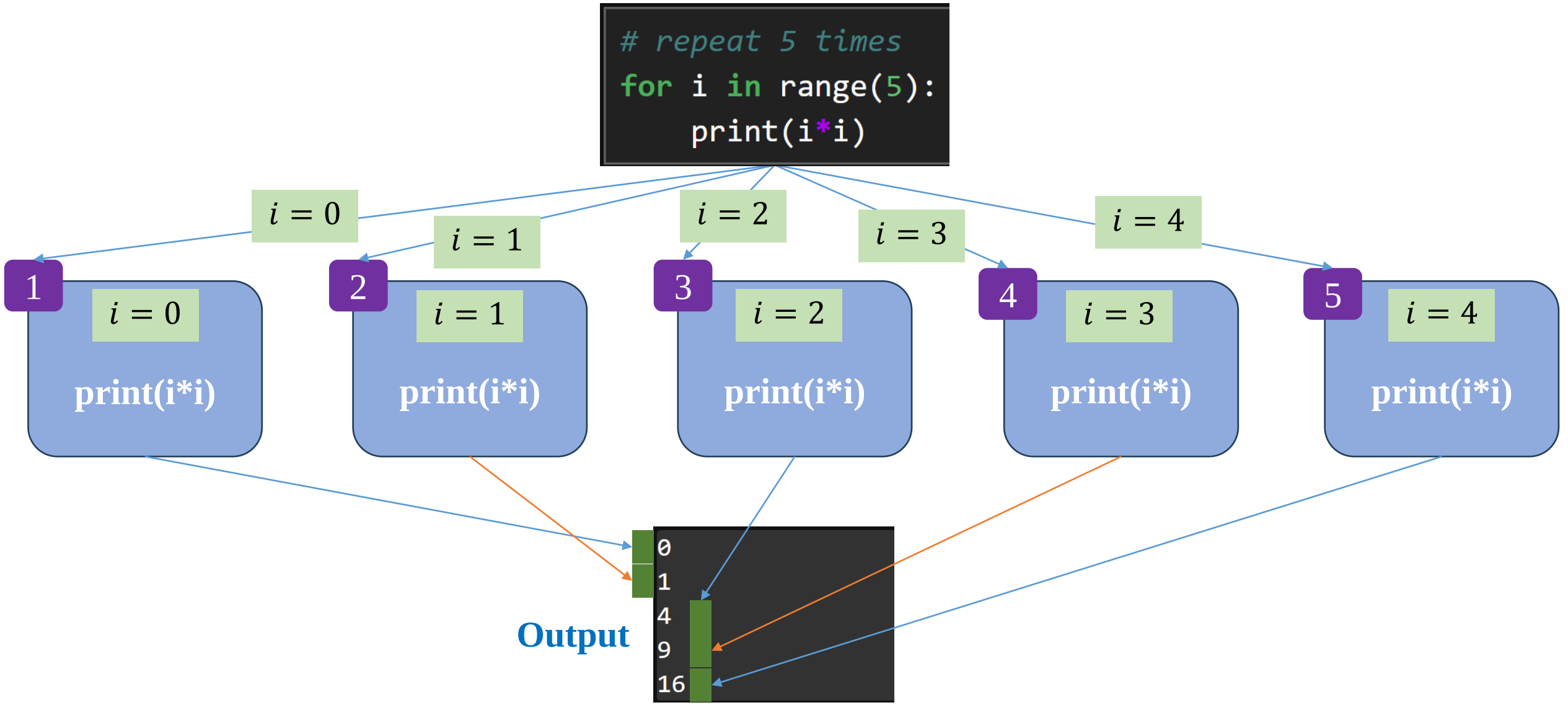
⬇

0, 1, 2, 3, 4

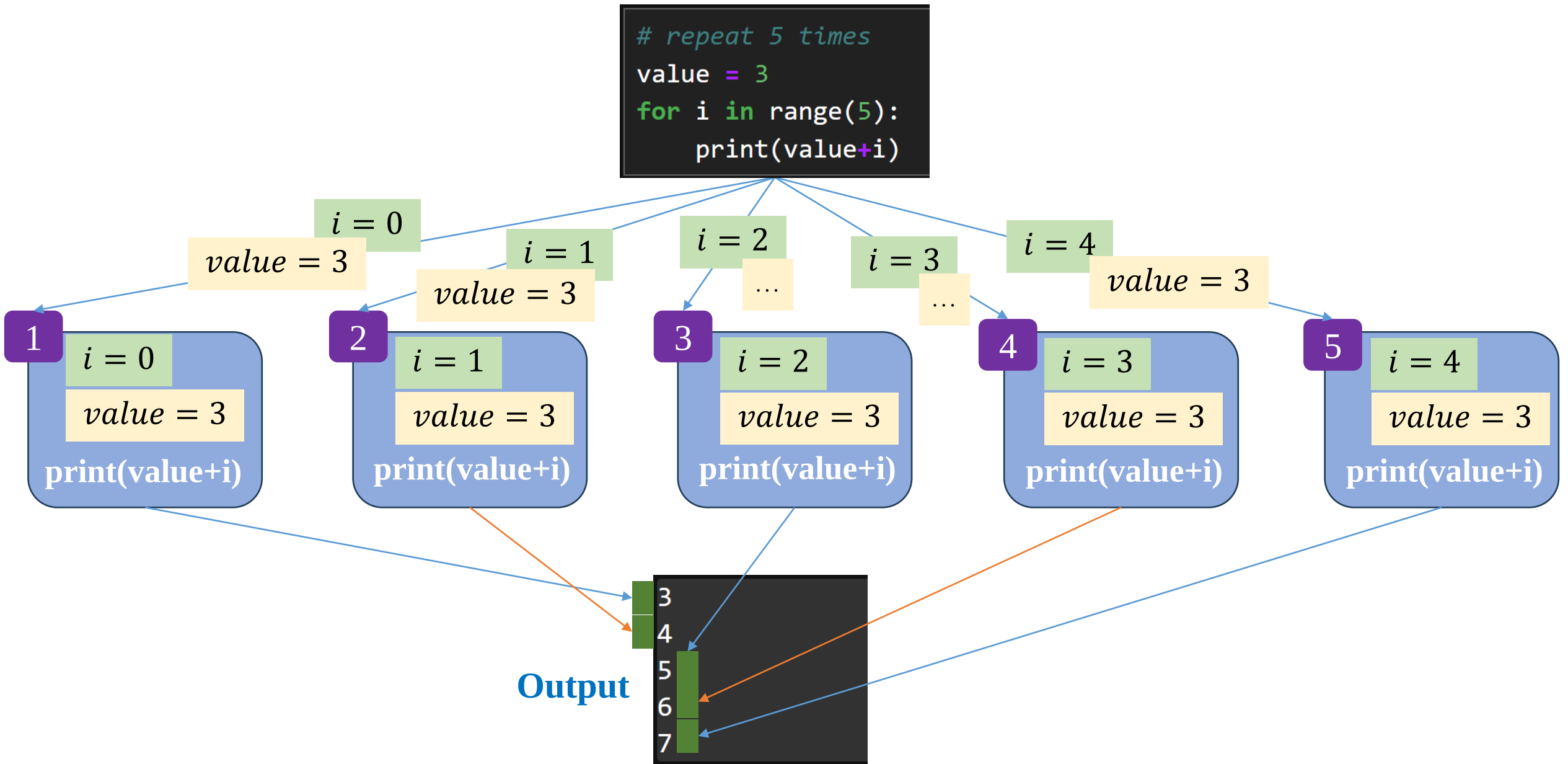
For Loop


# 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

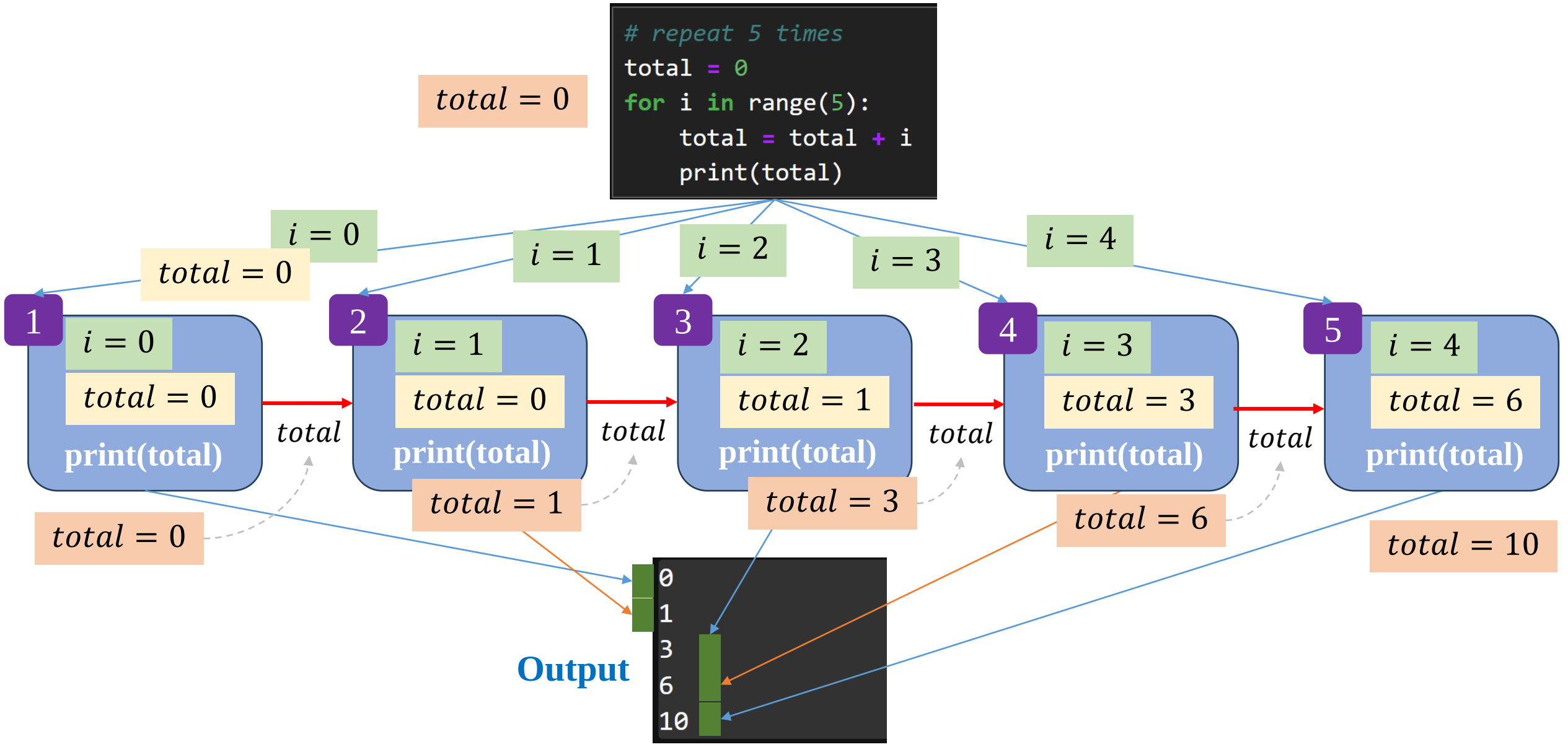
For Loop


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

For Loop

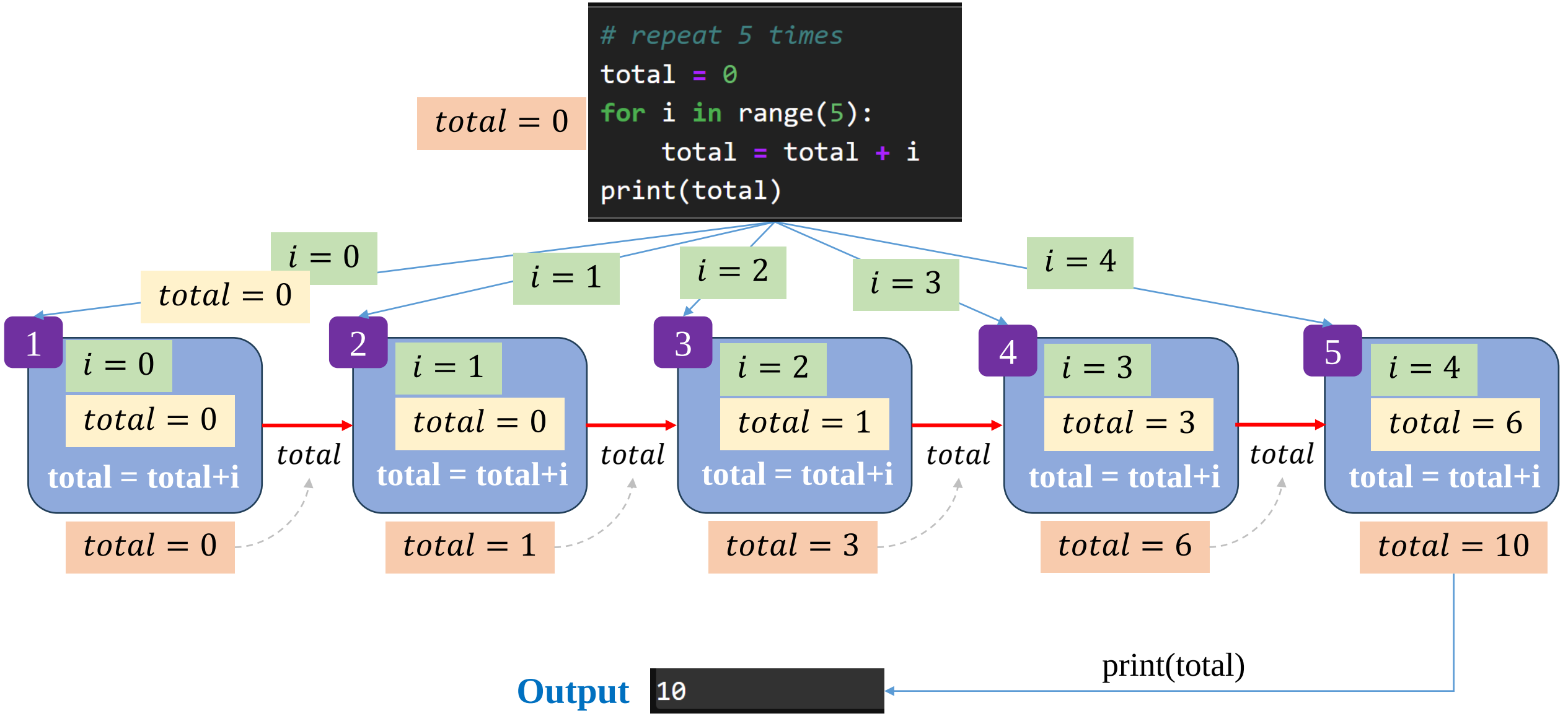

# repeat 5 times
value = 3
for i in range(5):
print(value+i)
i = 0
value = 3
i = 1
value = 3
i = 2
...
i = 3
...
i = 4
value = 3
1
i = 0
value = 3
print(value+i)
2
i = 1
value = 3
print(value+i)
3
i = 2
value = 3
print(value+i)
4
i = 3
value = 3
print(value+i)
5
i = 4
value = 3
print(value+i)
Output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

For Loop


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

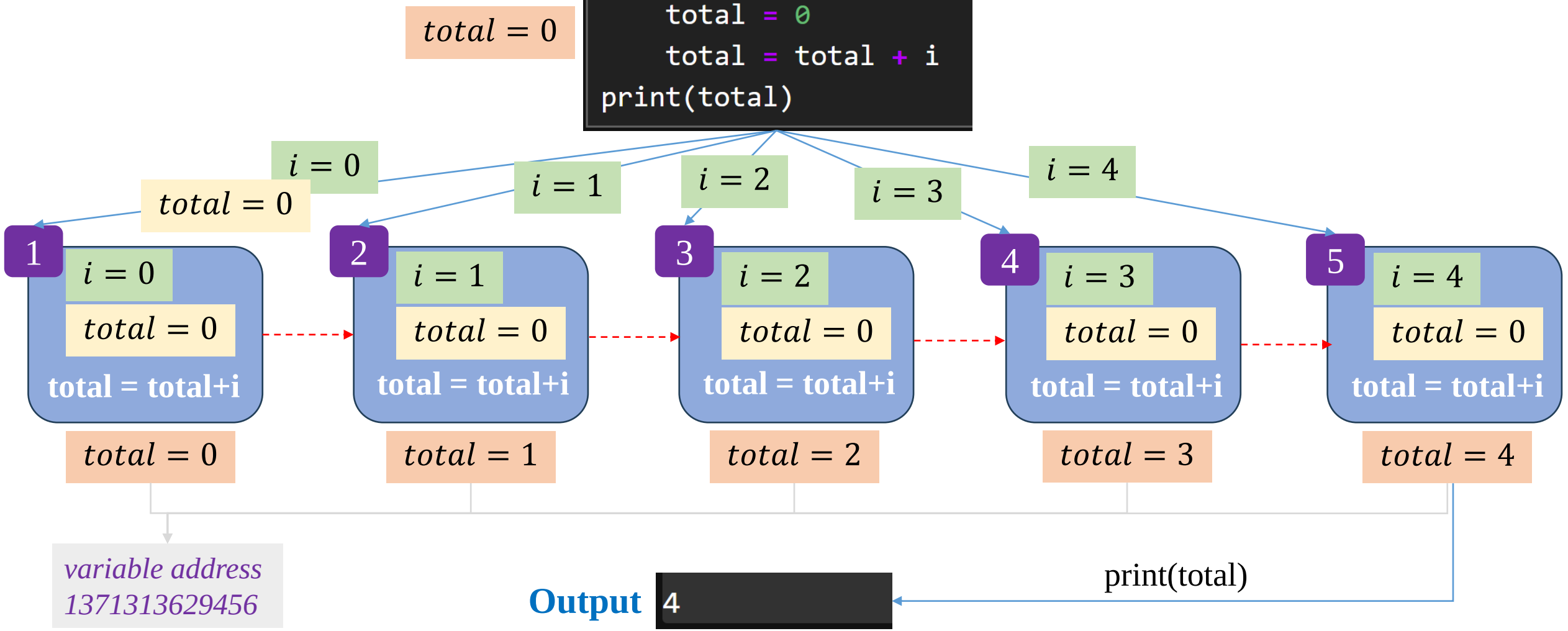
For Loop


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



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)
```

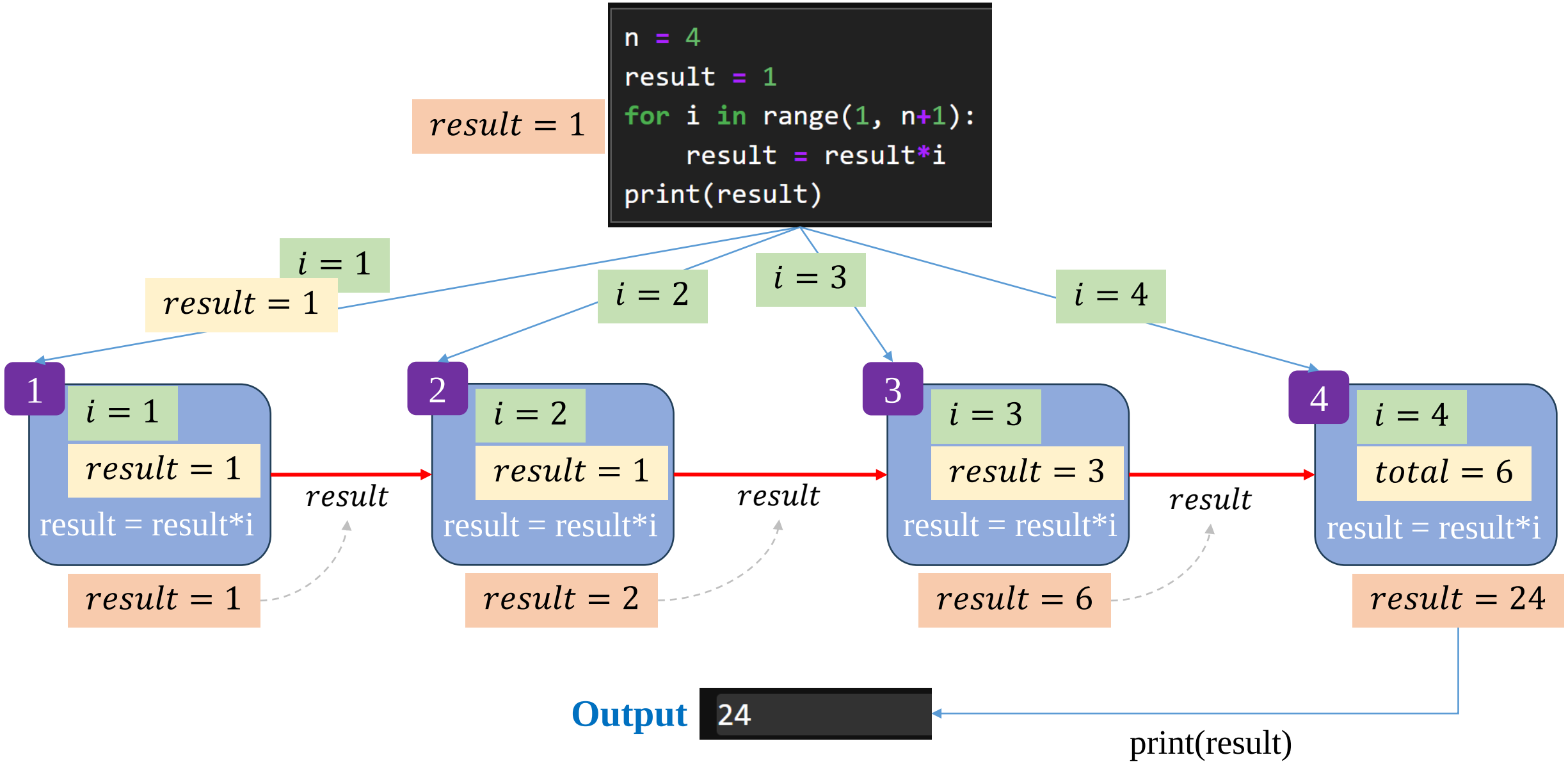
```
1
3
5
```

6

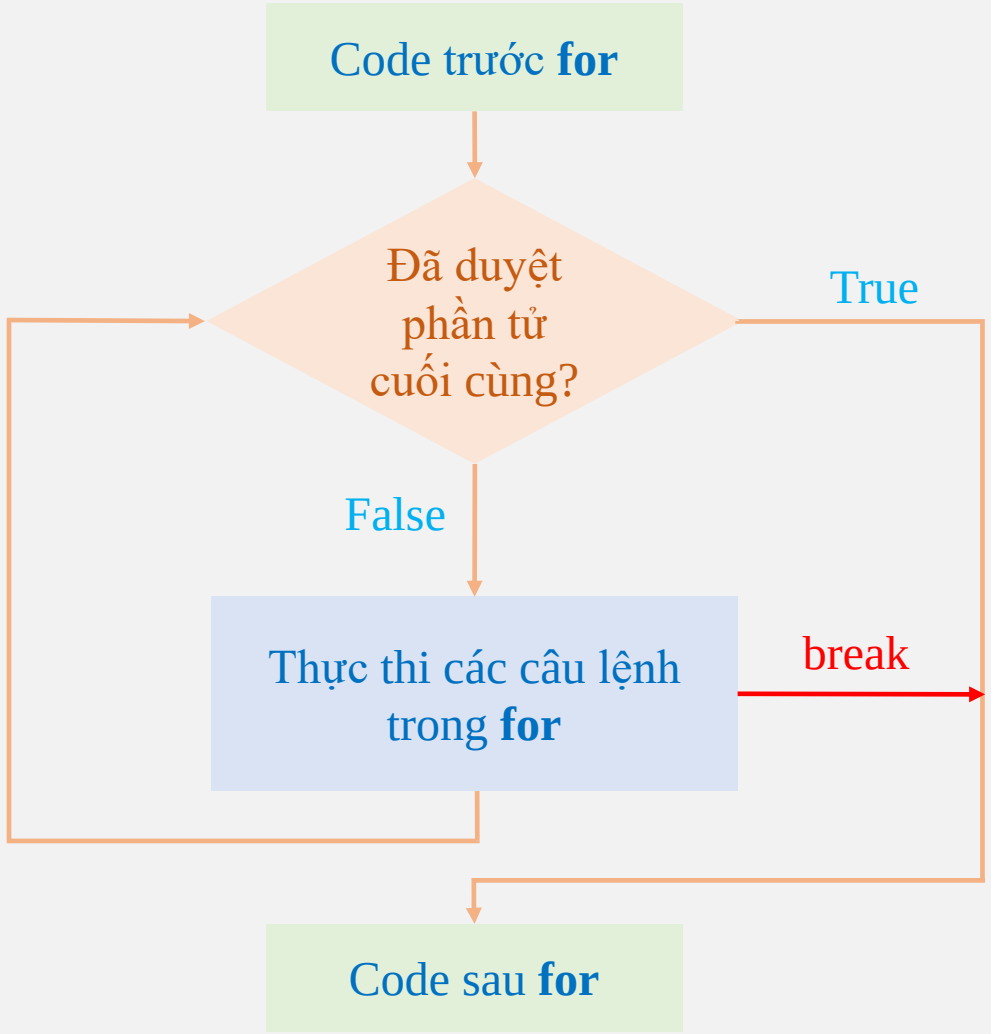
```
n = 3
for i in range(1, n+1):
    print(i)
```

```
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
2
i = 2
result = 1
result = result*i
result = 2
3
i = 3
result = 3
result = result*i
result = 6
4
i = 4
total = 6
result = result*i
result = 24
result
result
result
Output
24
print(result)

For Loop

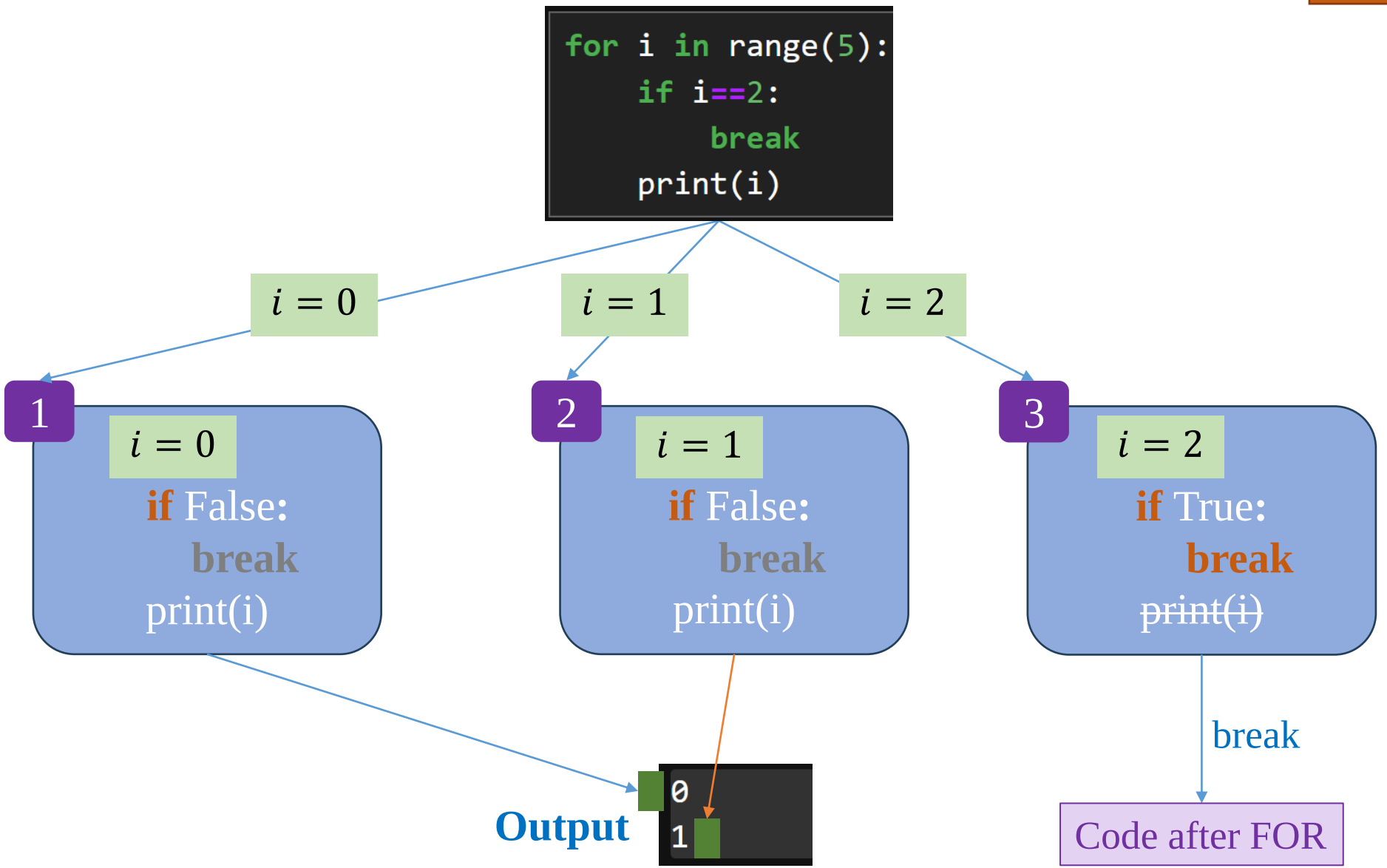


break keyword

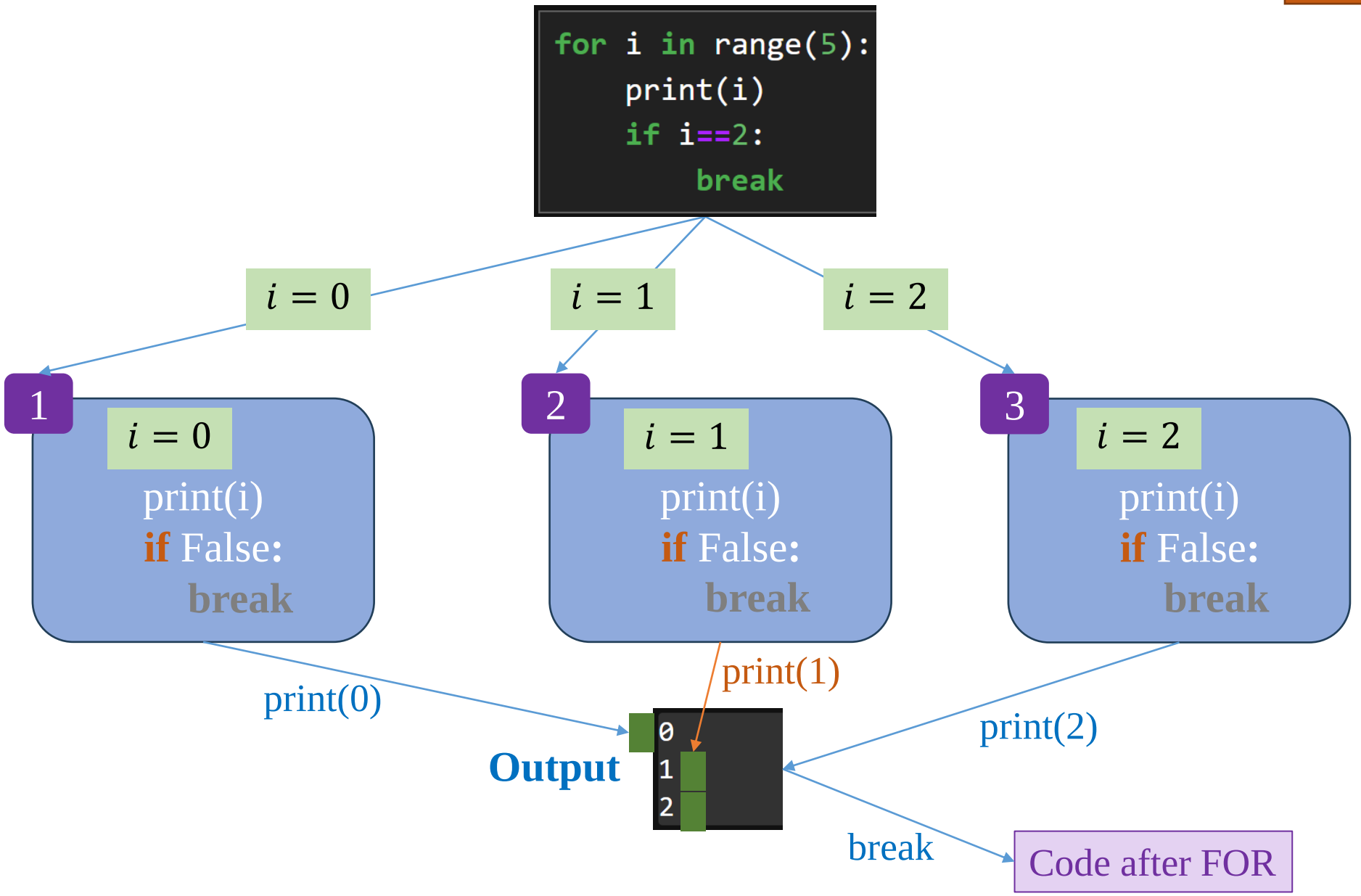
```python
# duyệt phần tử trong range(10)
for i in range(10):
    # hỏi phần tử i có bằng 5 không?
    if i == 5:
        # nếu bằng thì thoát vòng lặp for này
        break

    # 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
```

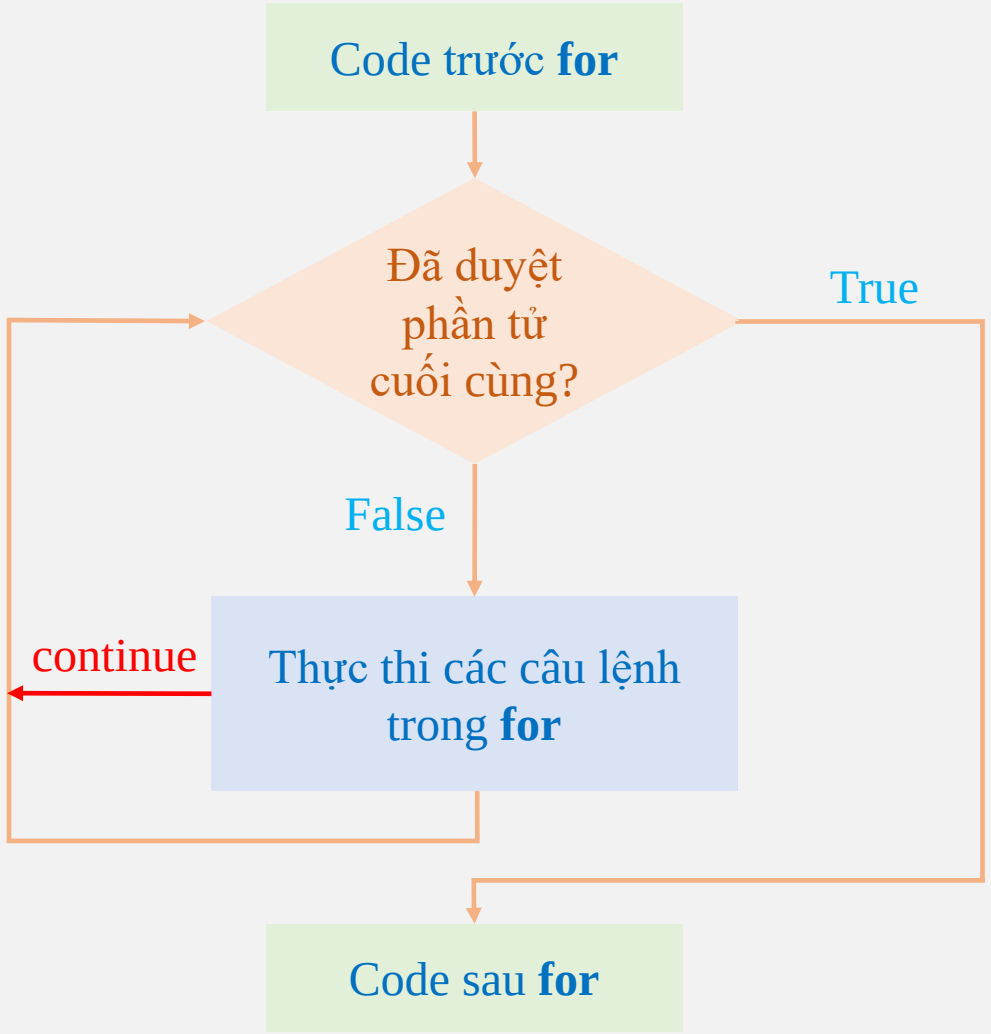
For Loop


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 False:
break
print(0)
print(1)
print(2)
Output
0
1
2
break
Code after FOR

For Loop

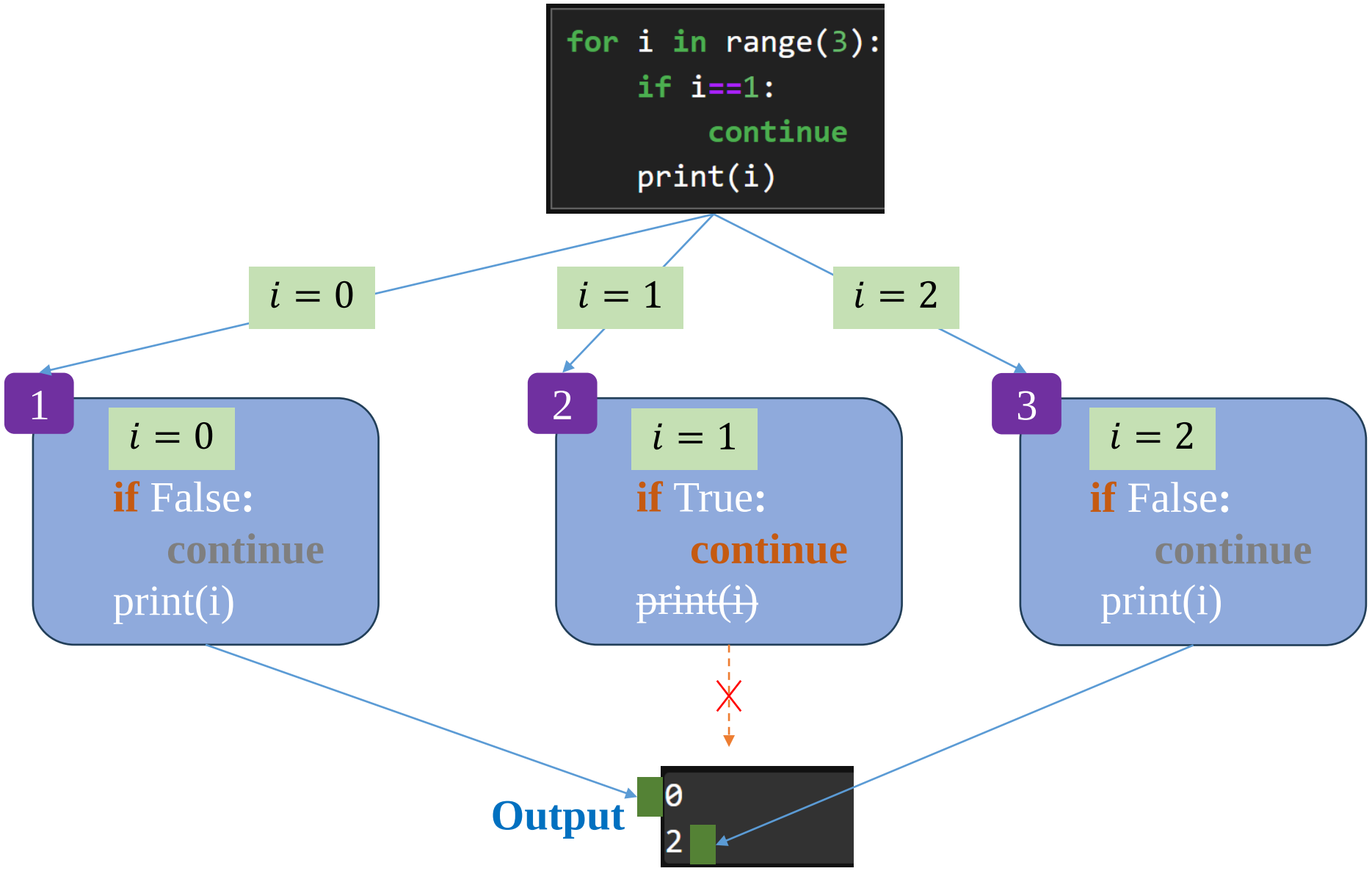


continue keyword

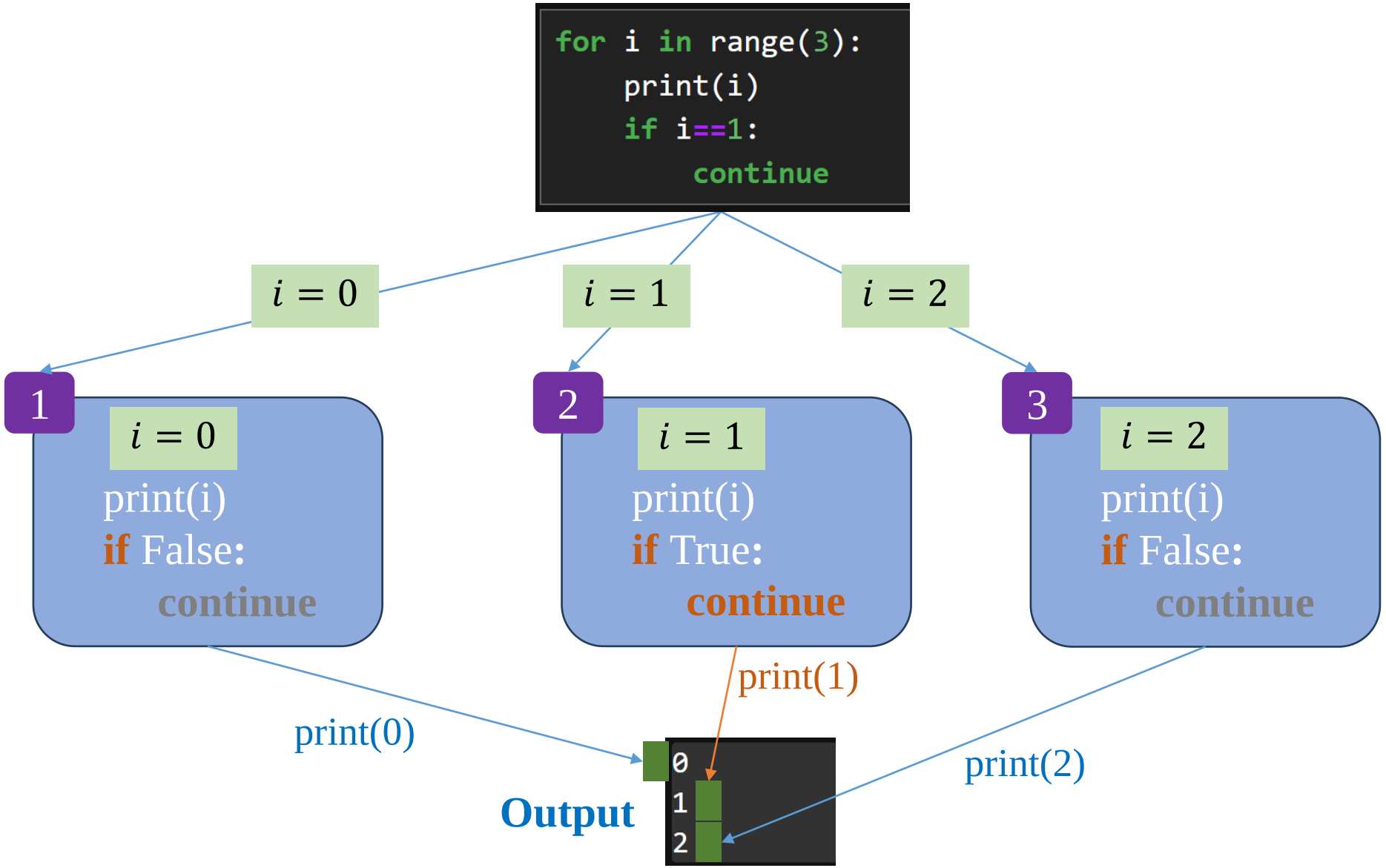
```
1.  # duyệt phần tử trong range(10)
2.  for i in range(10):
3.      # hỏi phần tử i có bằng 5 không?
4.      if i == 5:
5.          # nếu bằng thì gọi continue
6.          # phần code sau continue sẽ không
7.          # được thực thi trong lần lặp này
8.          continue
9.
10.     # làm gì đó với i
11.     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
```

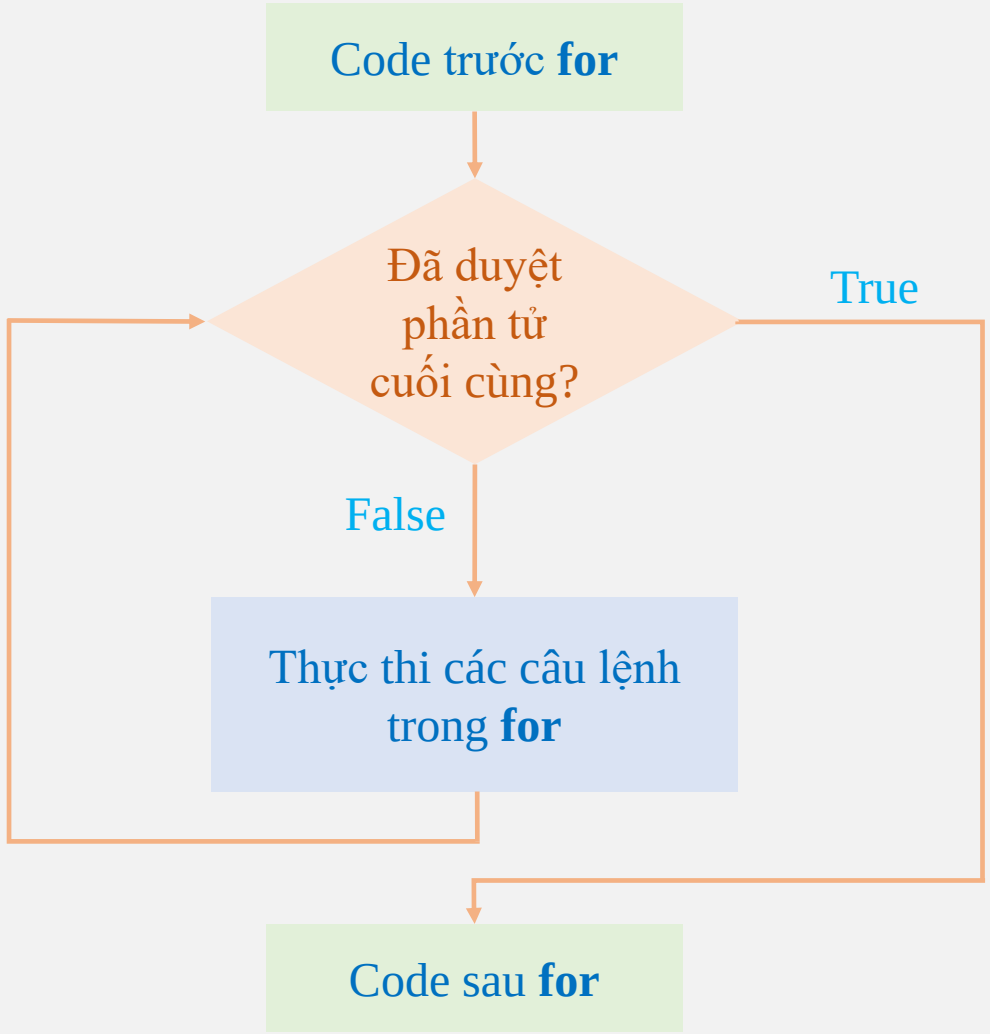
For Loop


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
```

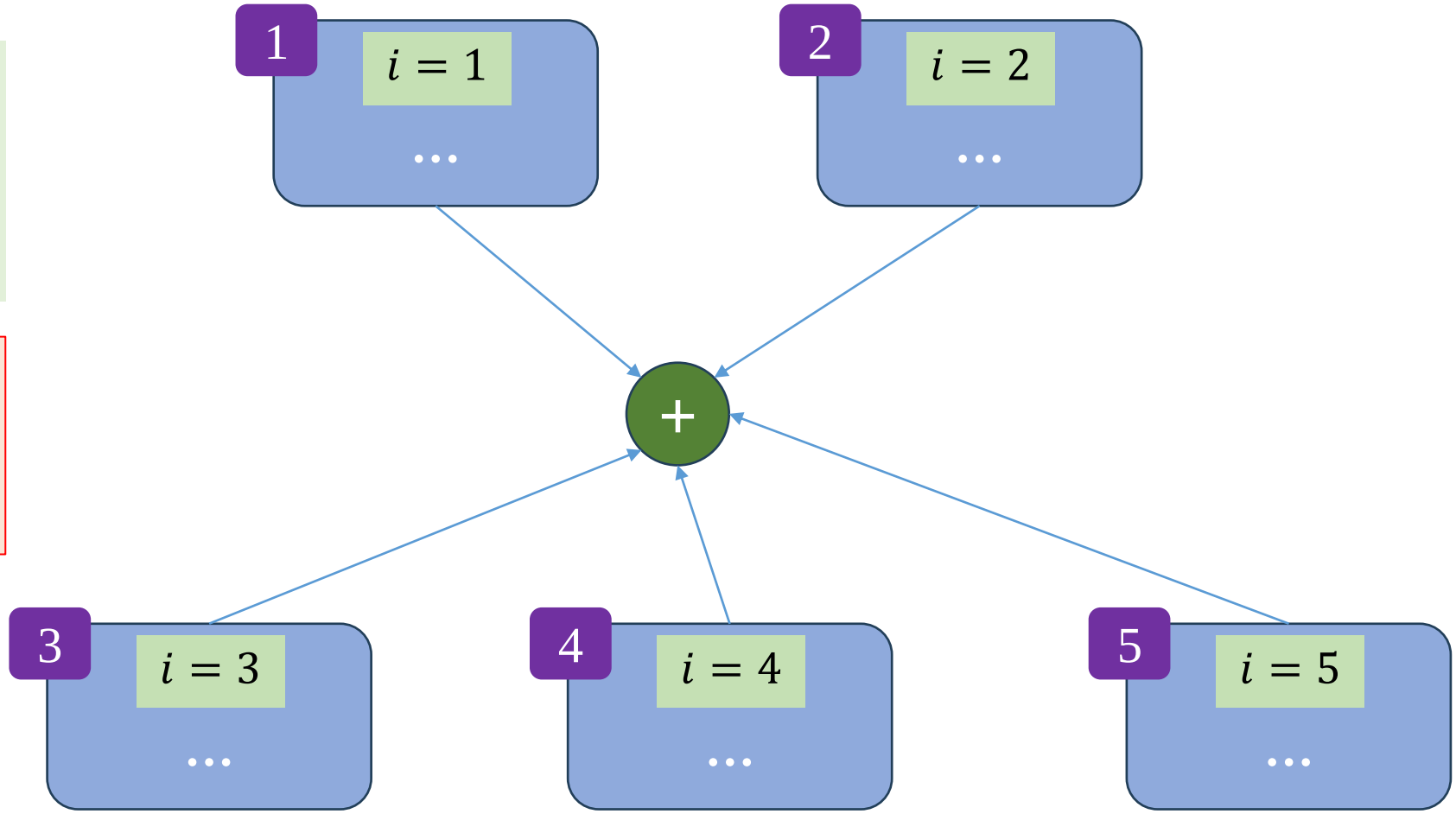
Example

❖ 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

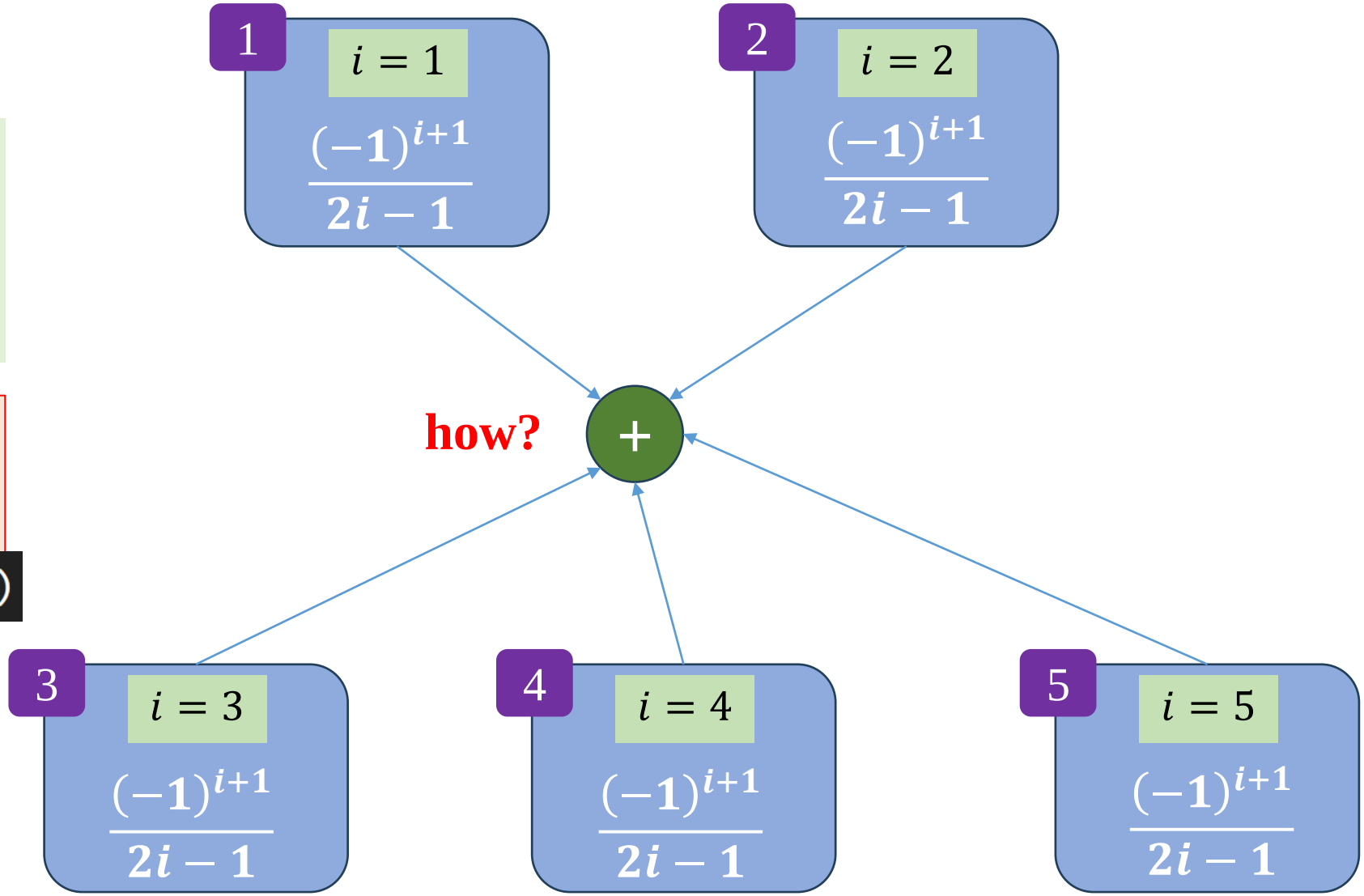
❖ PI estimation

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

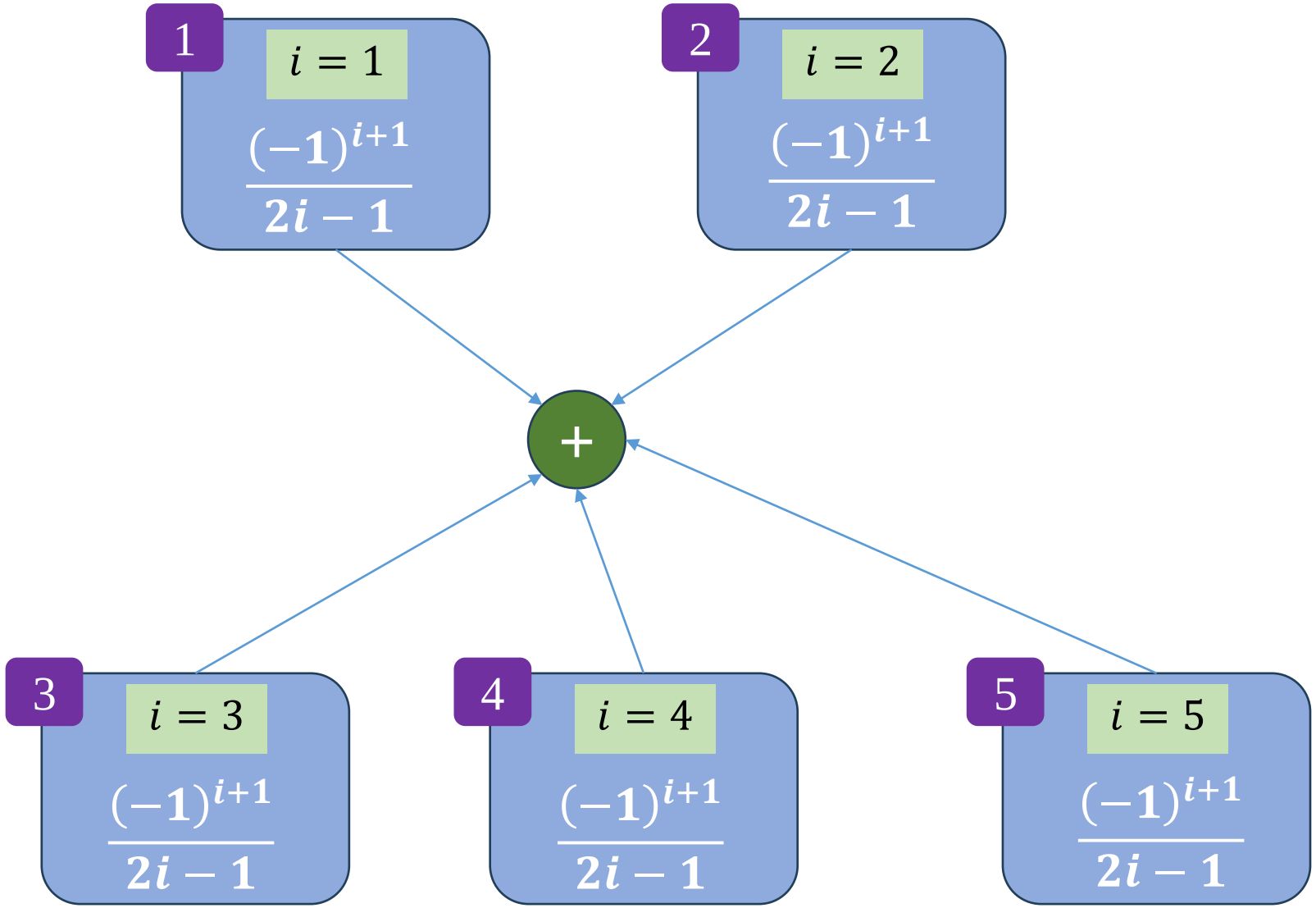
Gregory-Leibniz Series

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

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

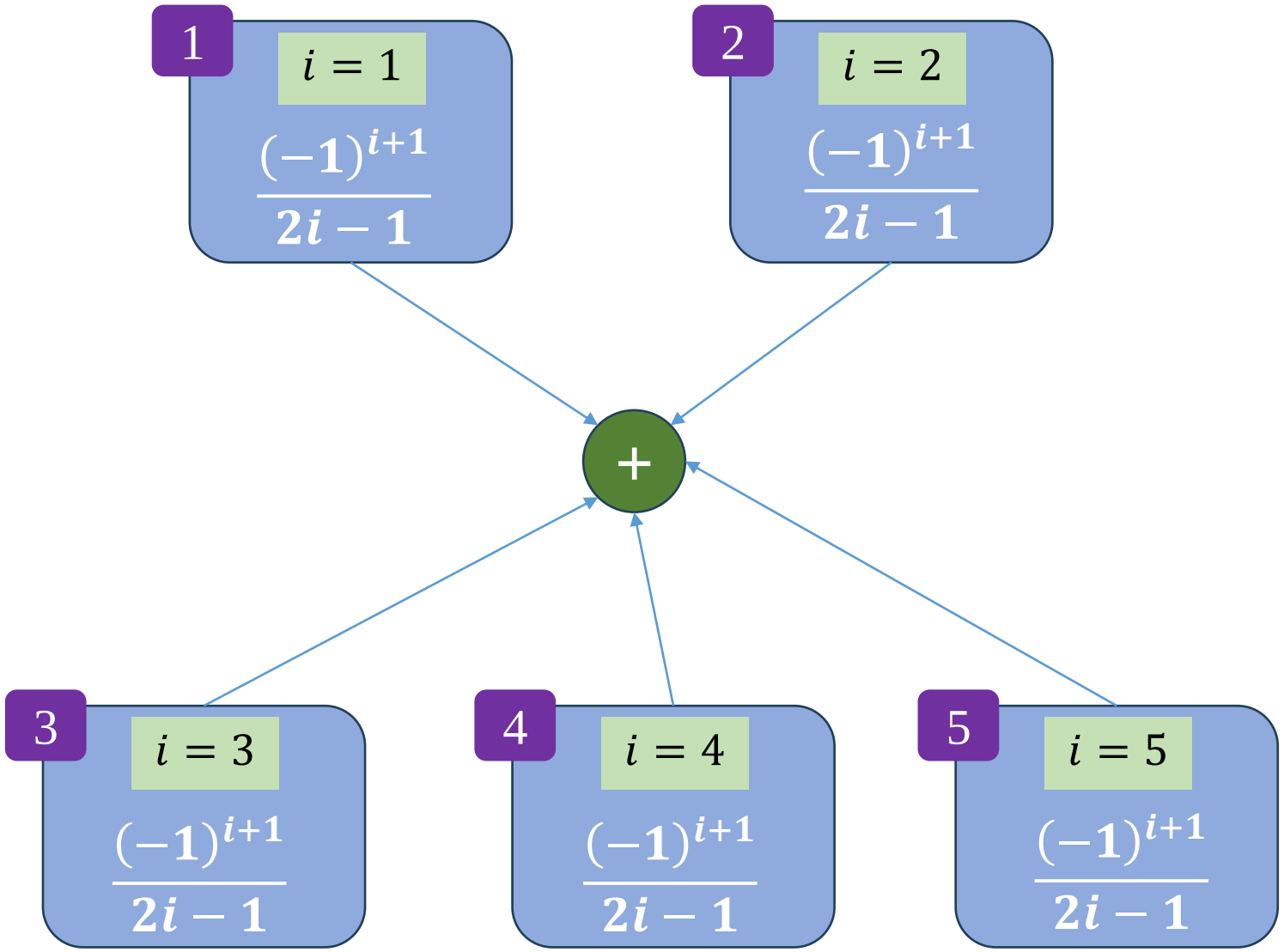
❖ PI estimation

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

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

...

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_1 x_2 x_3

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

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

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

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

...

```
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 r r r r

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

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

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

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

...

```
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
```

Compute the right side first, then assign to r

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

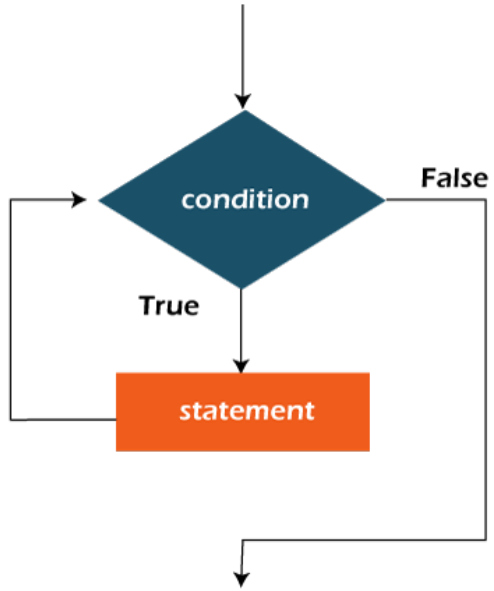
WHILE Loop

SECTION 3

Files

SECTION 4

Examples




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

`True/False ← condition`

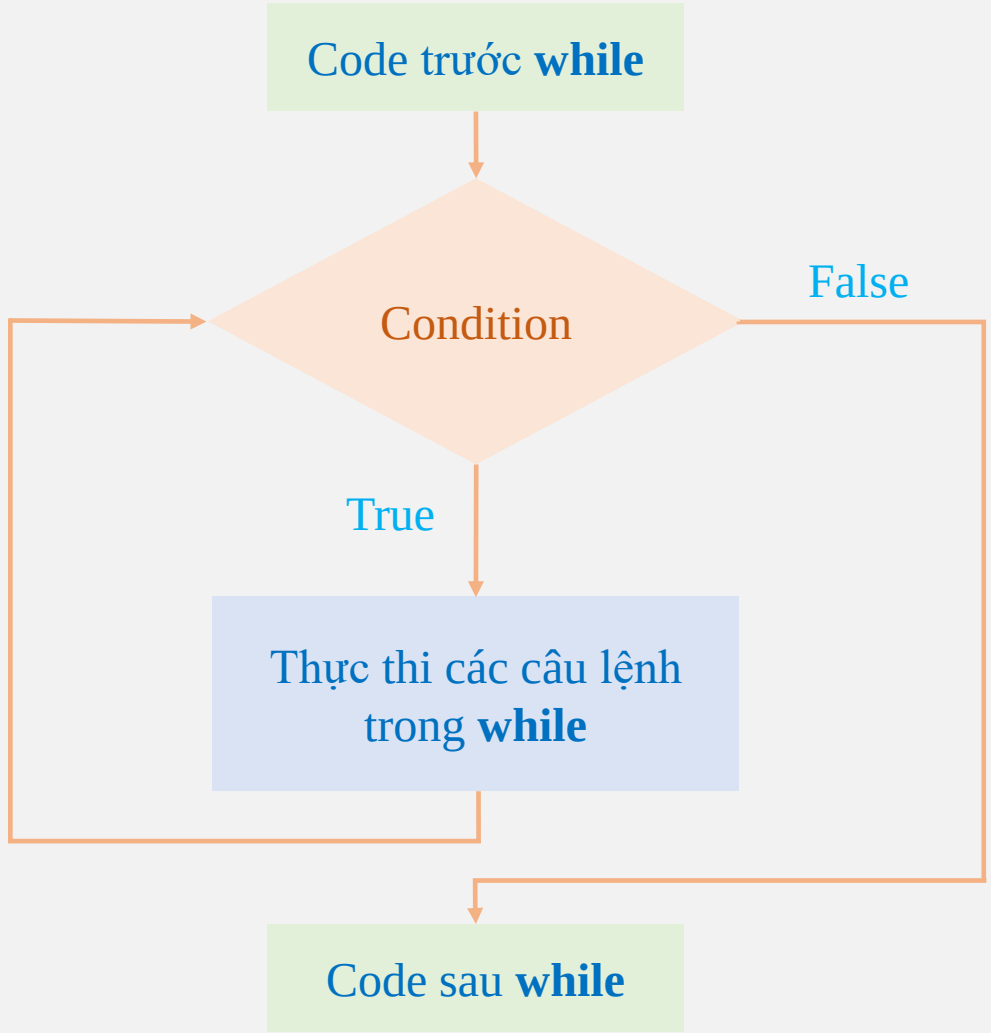
While Loop

keyword

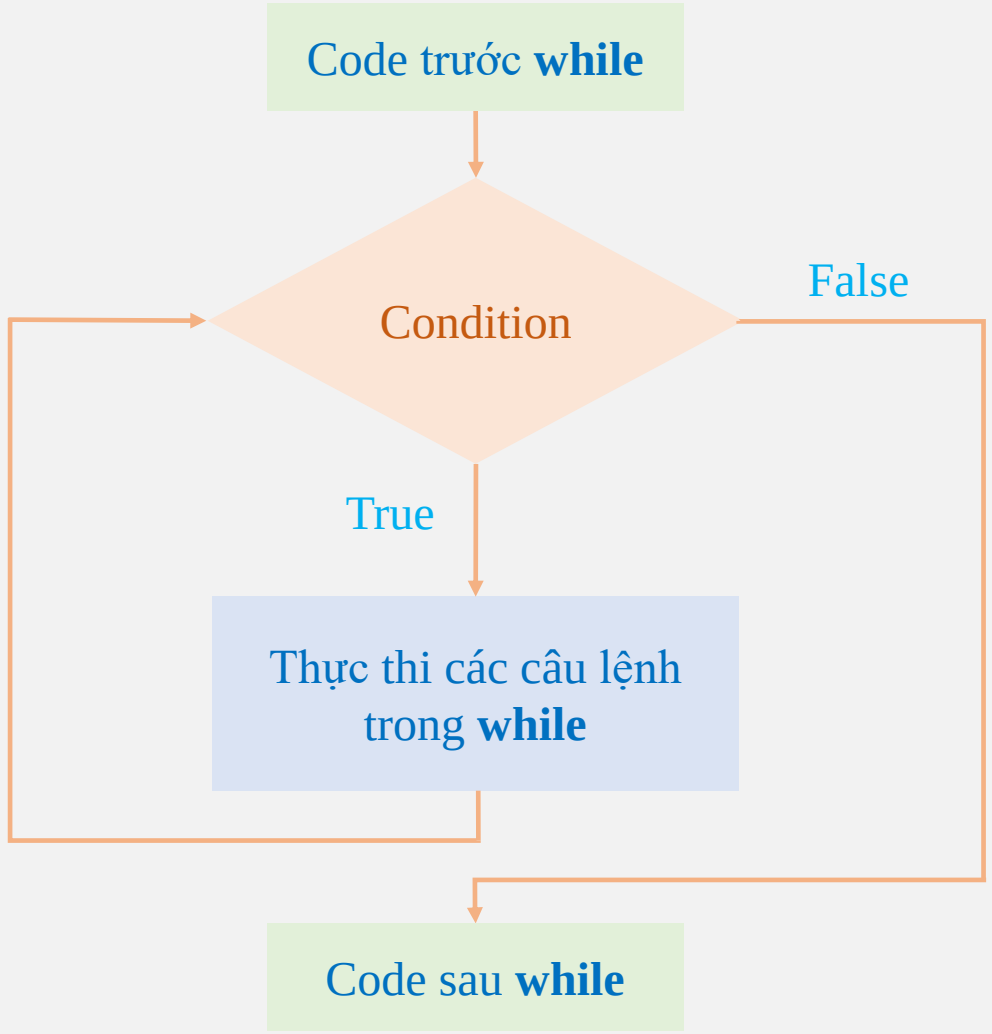
colon

```
# code trước while

while condition:
    # khối code trong while

# code sau while
```

indentation



While Loop

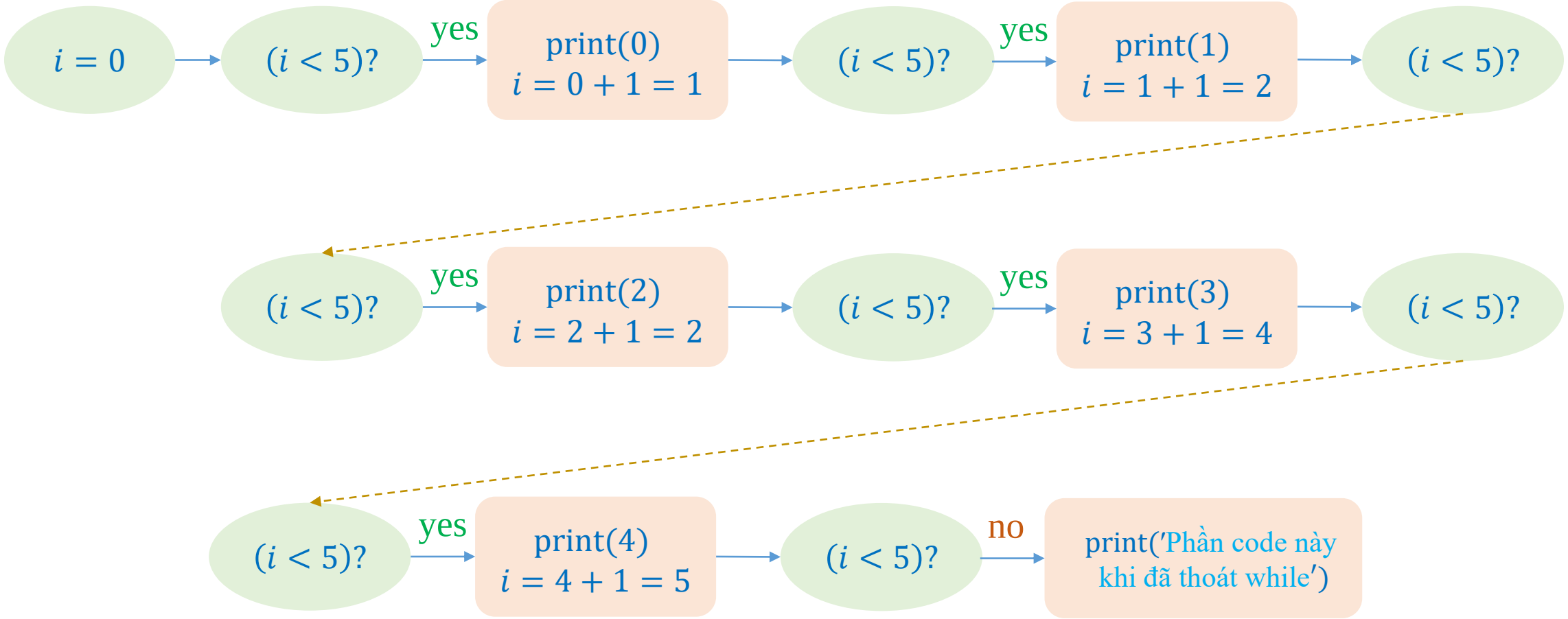



```python
# tạo biến i
i = 0

# 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
```

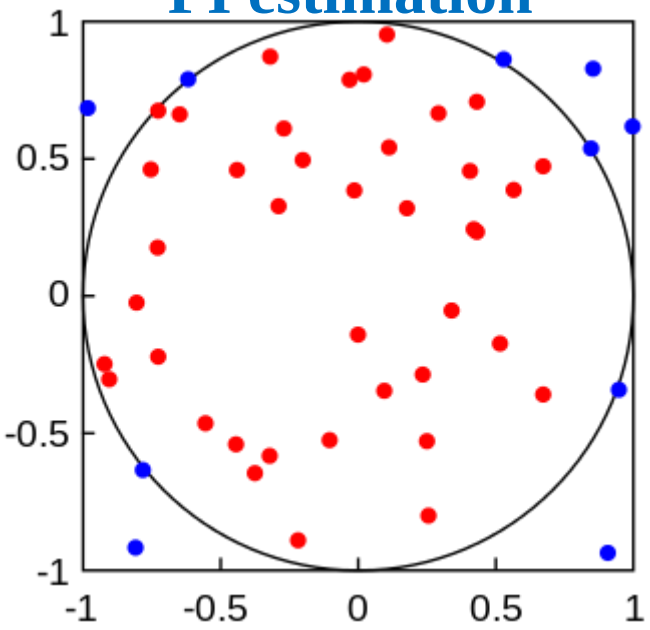
Exercises

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

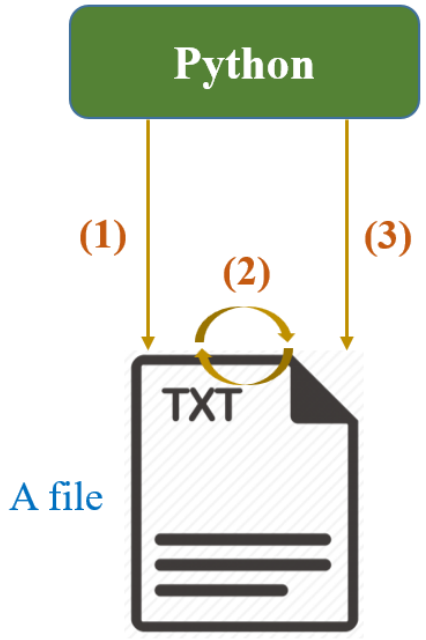
WHILE Loop

SECTION 3

Files

SECTION 4

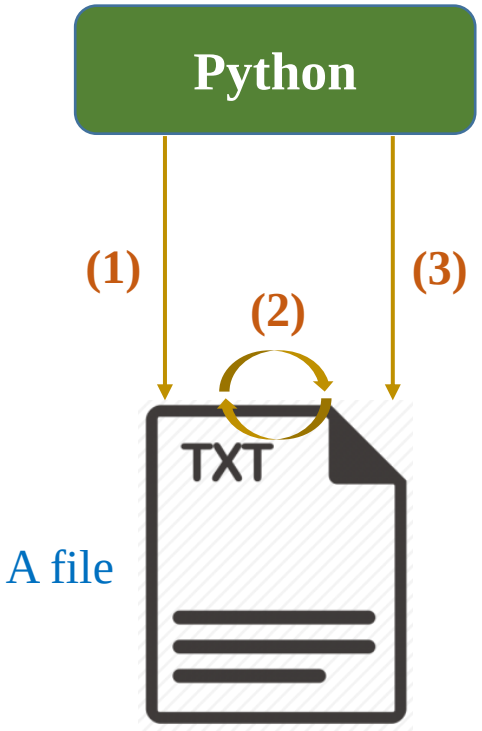
Examples



(1)	Connect to file
(2)	Read from/write to file
(3)	Disconnect to file

File

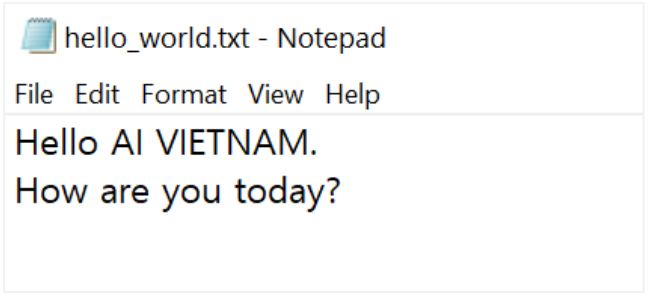
❖ Typical procedure



(1)	Connect to file
(2)	Read from/write to file
(3)	Disconnect to file

Read from a file (already exist)

(1)	open(file_path, 'r')
(2)	read()
(3)	close()

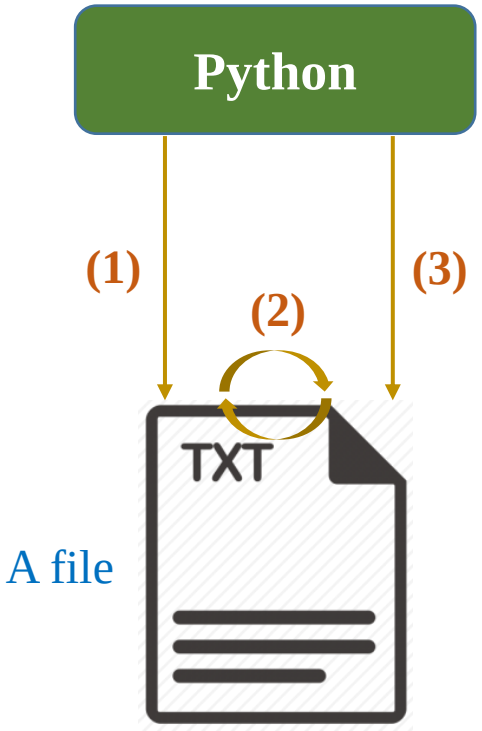



```python
# kết nối với file
a_file = open('hello_world.txt','r')

# read content as string
data = a_file.read()

print(type(data))
print(data)

# Đóng kết nối với file
a_file.close()
```

```
<class 'str'>
Hello AI VIETNAM.
How are you today?
```

File

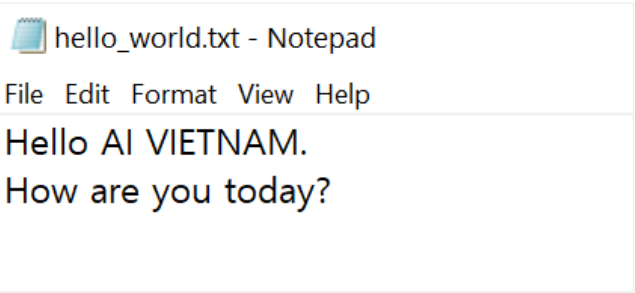
❖ Typical procedure



(1)	Connect to file
(2)	Read from/write to file
(3)	Disconnect to file

Read content from a file as lines

(1)	open(file_path, 'r')
(2)	readlines()
(3)	close()

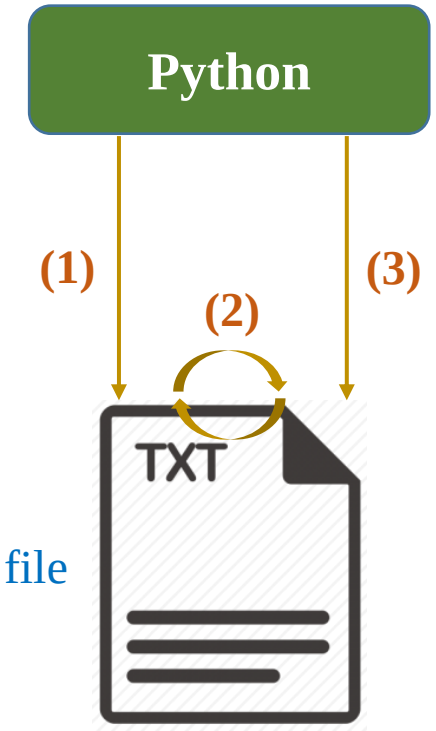



```python
# kết nối với file
a_file = open('hello_world.txt','r')

# read content as string
lines = a_file.readlines()
for line in lines:
    print(line)

# Đóng kết nối với file
a_file.close()
```

```
Hello AI VIETNAM.

How are you today?
```

File

❖ Typical procedure

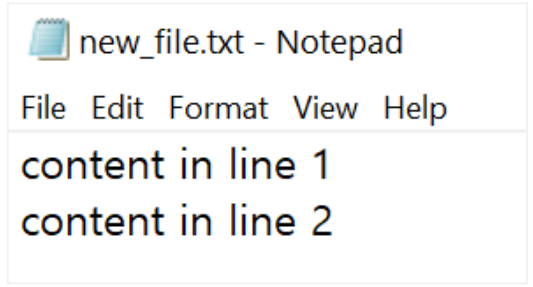


(1)	Connect to file
(2)	Read from/write to file
(3)	Disconnect to file

Write to a file (not exist)

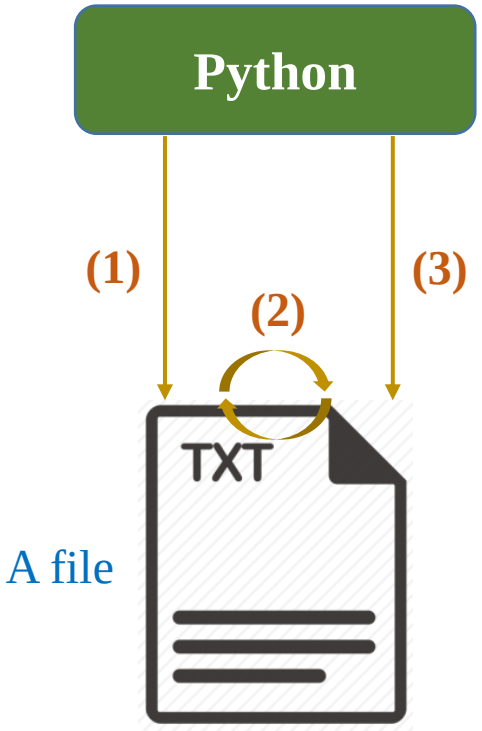
(1)	open(file_path, 'w')
(2)	write()
(3)	close()

```
1  # kết nối với file
2  a_file = open('new_file.txt', 'w')
3
4  text1 = 'content in line 1 \n'
5  a_file.write(text1)
6
7  text2 = 'content in line 2 \n'
8  a_file.write(text2)
9
10 # Đóng kết nối với file
11 a_file.close()
```



File

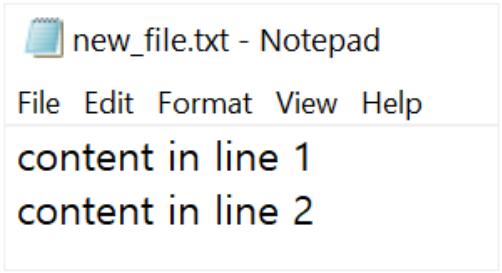
❖ Typical procedure



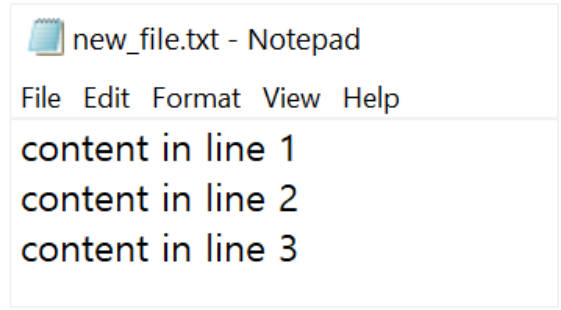
(1)	Connect to file
(2)	Read from/write to file
(3)	Disconnect to file

Write to a file (appending content if the file already exists)

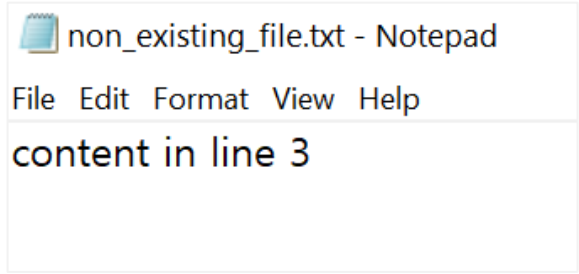
(1)	open(file_path, 'a')
(2)	write()
(3)	close()




```
1  # kết nối với file
2  a_file = open('new_file.txt', 'a')
3
4  text3 = 'content in line 3 \n'
5  a_file.write(text3)
6
7  # Đóng kết nối với file
8  a_file.close()
```




```
1  # kết nối với file
2  a_file = open('non_existing_file.txt', 'a')
3
4  text3 = 'content in line 3 \n'
5  a_file.write(text3)
6
7  # Đóng kết nối với file
8  a_file.close()
```



❖ Common Error

```
1.  # aivietnam.ai
2.  # Lỗi đọc file không tồn tại
3.
4.  my_file = open("file.txt","r")
5.  print(my_file)
```

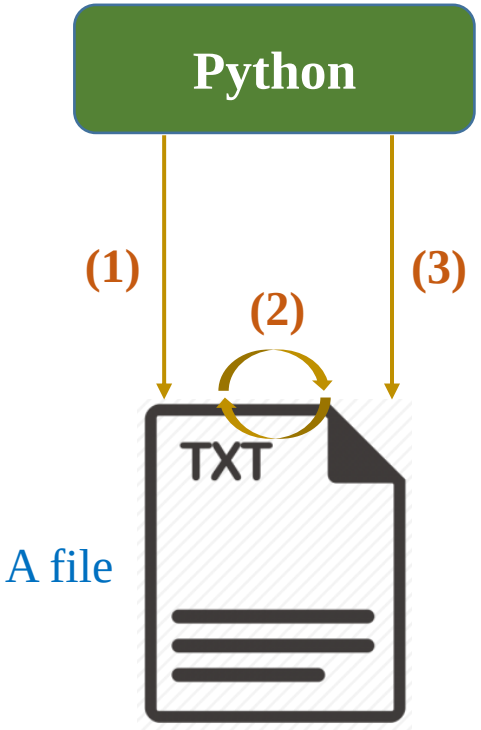
```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-13-73d8e6dda2db> in <module>
      2 # Lỗi đọc file không tồn tại
      3
----> 4 my_file = open("file.txt","r")
      5 print(my_file)

FileNotFoundError: [Errno 2] No such file or directory: 'file.txt'
```

File

❖ Example

```python
# open a file
a_file = open('hello_world.txt', 'w')

# write data to file
text3 = 'writing line \n'
a_file.write(text3)
```

```python
# open a file
a_file = open('hello_world.txt', 'w')

# write data to file
text3 = 'writing line \n'
a_file.write(text3)

# close the file
a_file.close()
```



❖ with keyword

```python
# using with

with open('hello_world.txt', 'w') as file:
    file.write('writing line \n')
```

Outline

SECTION 1

FOR Loop

SECTION 2

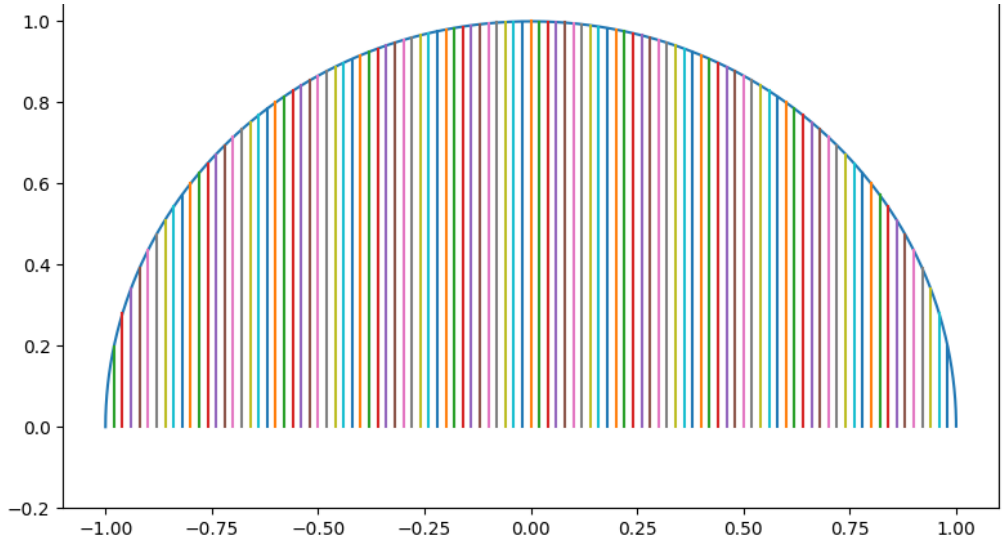
WHILE Loop

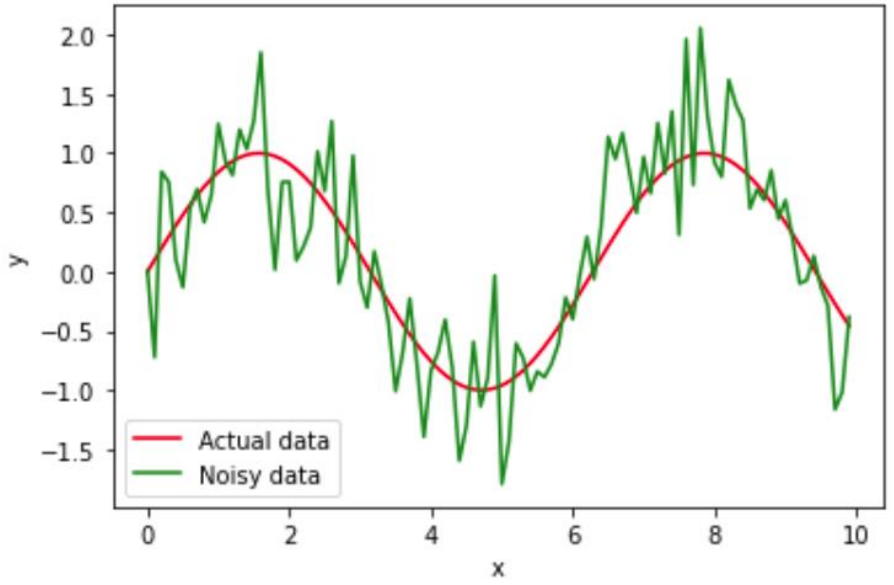
SECTION 3

Files

SECTION 4

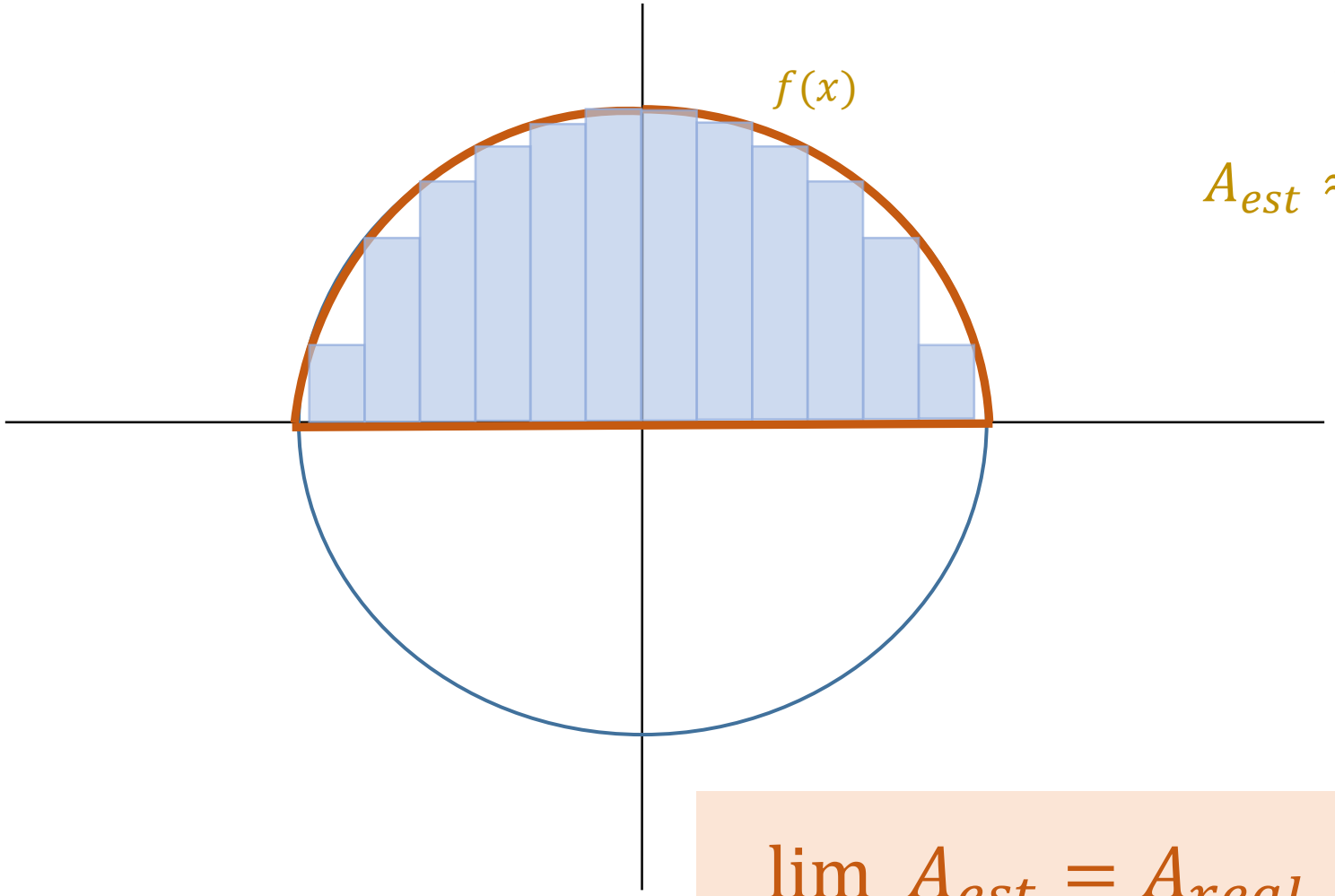
Examples





Example 1

❖ Compute the area of a unit circle



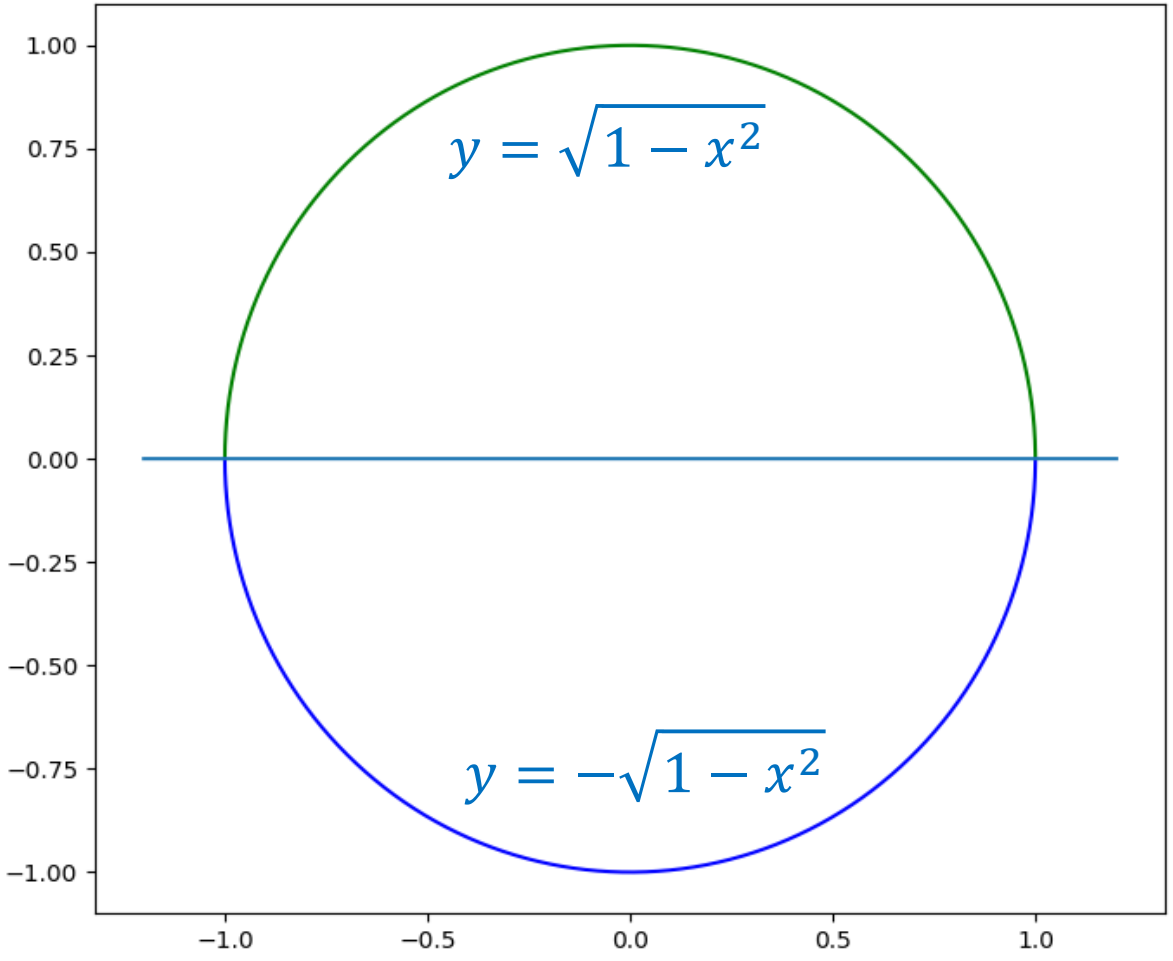
$$A_{est} \approx f(x_1)\Delta x_1 + f(x_2)\Delta x_2 + \dots + f(x_n)\Delta x_n$$

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

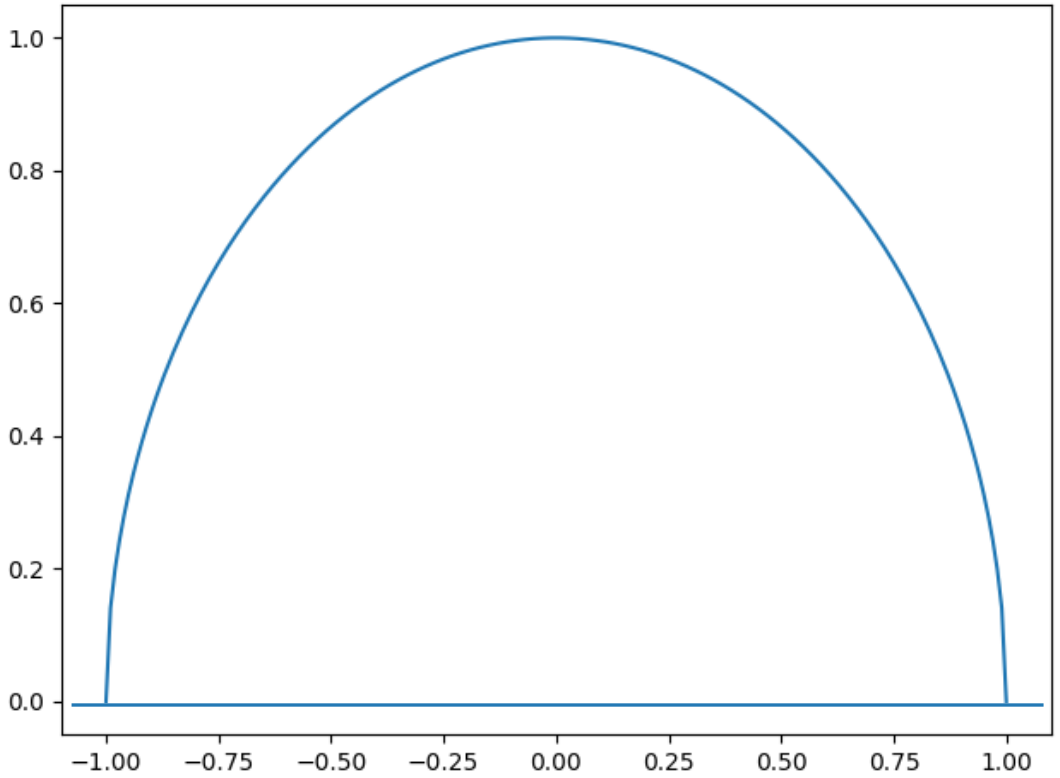
$$\lim_{\Delta x \to 0} A_{est} = A_{real}$$

Example 1

❖ Compute the area of a unit circle




```python
import math

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



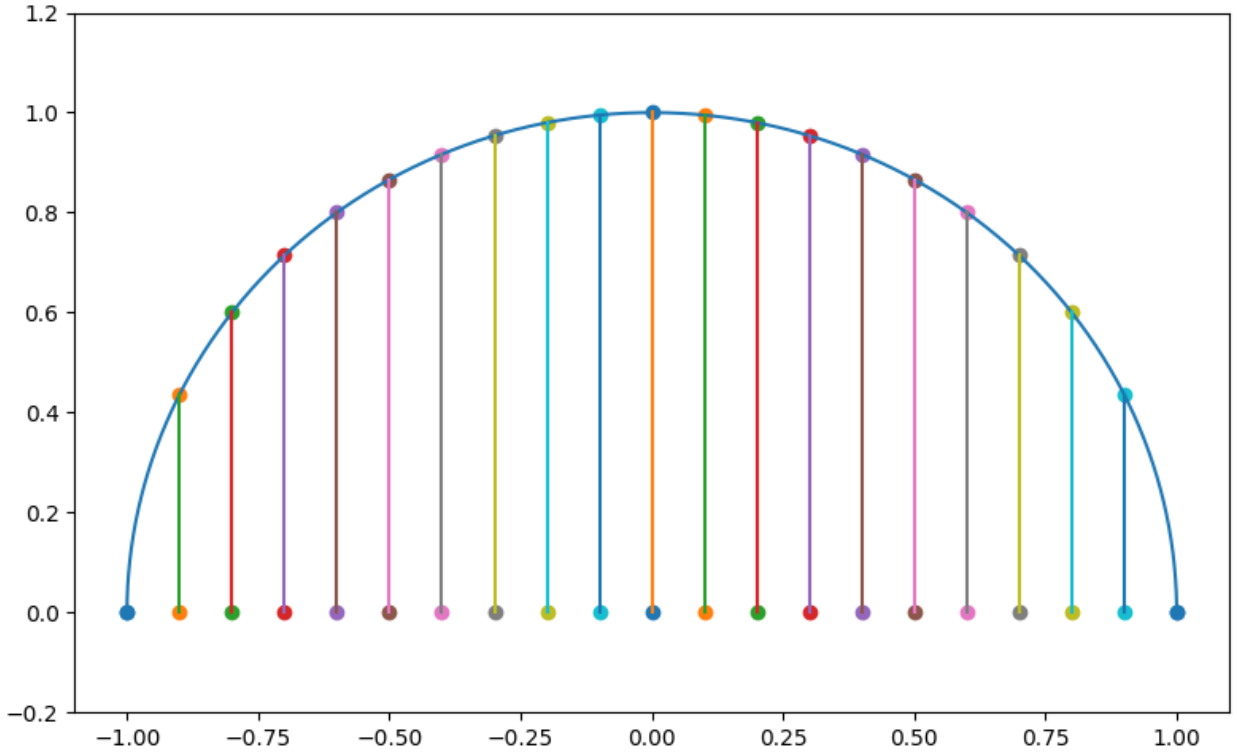
Example 1

❖ Compute the area of a unit circle

math.pi=3.141592

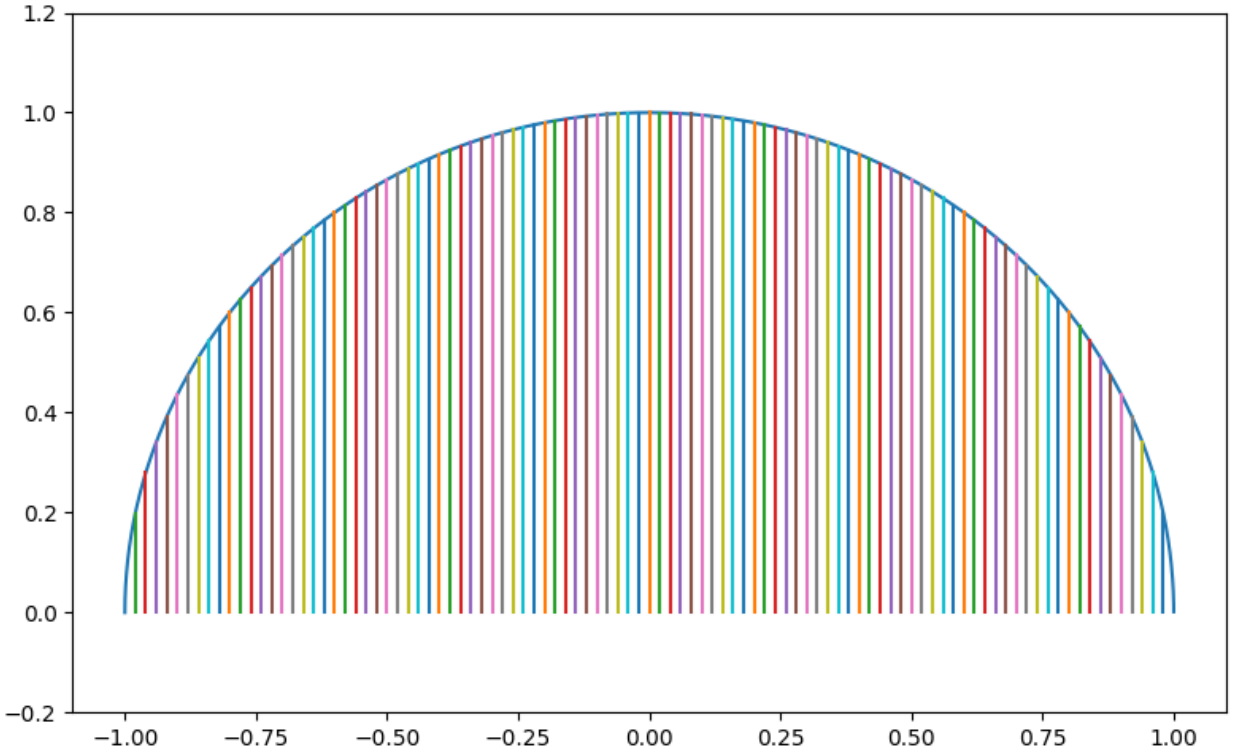
$n = 20$

$A_{est} = 3.1045$



$n = 200$

$A_{est} = 3.1404$



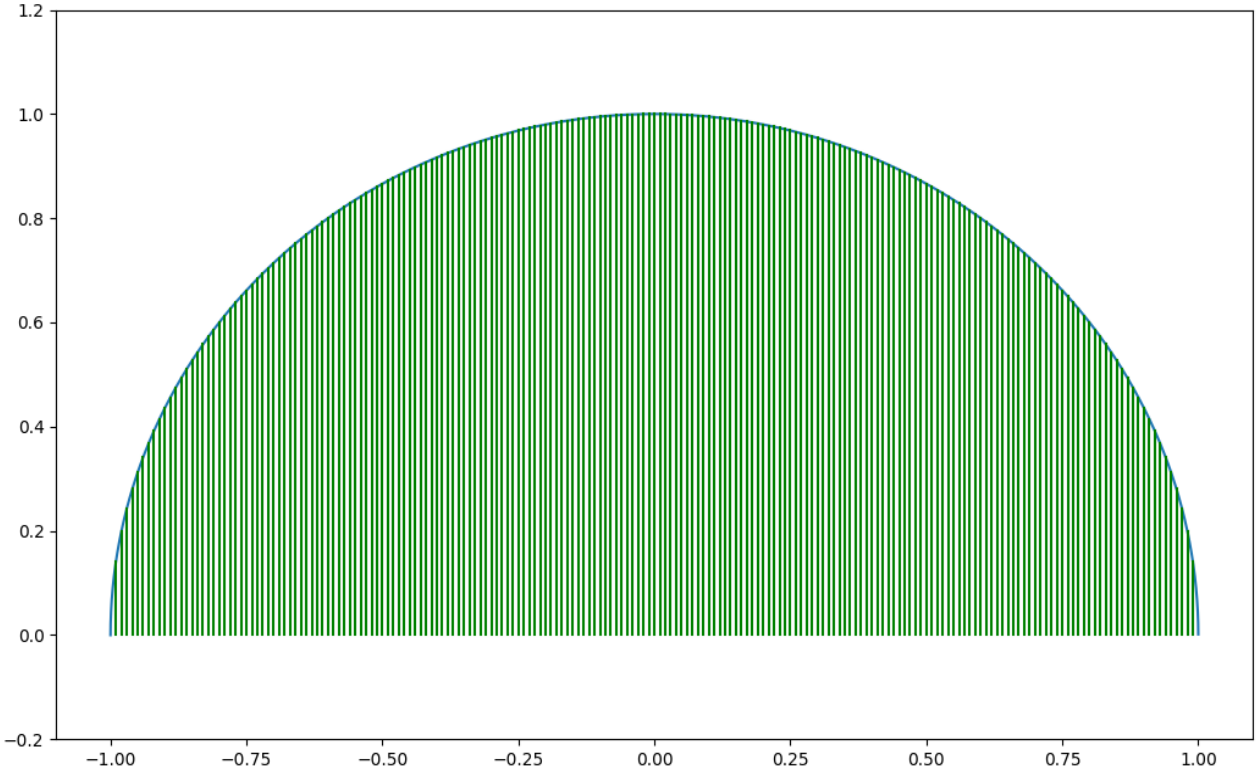
Example 1

❖ Compute the area of a unit circle

math.pi=3.141592

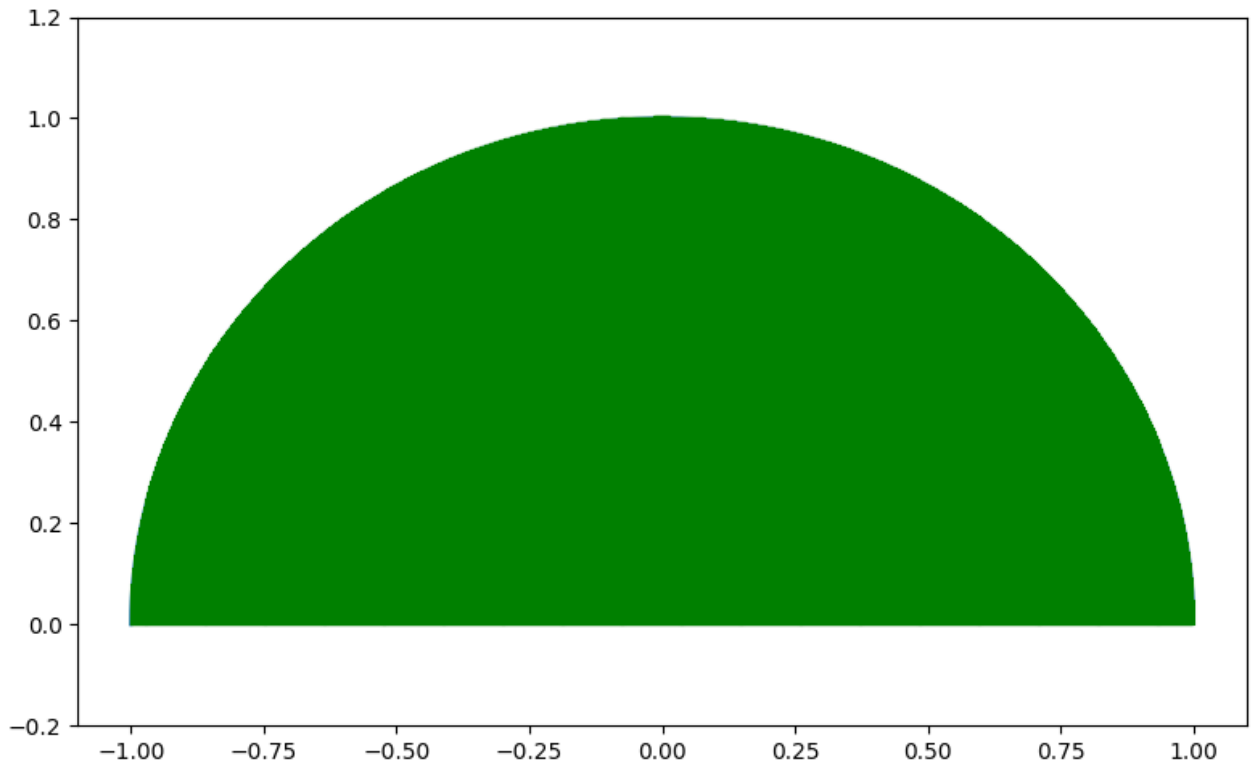
$n = 200$

$A_{est} = 3.1404$



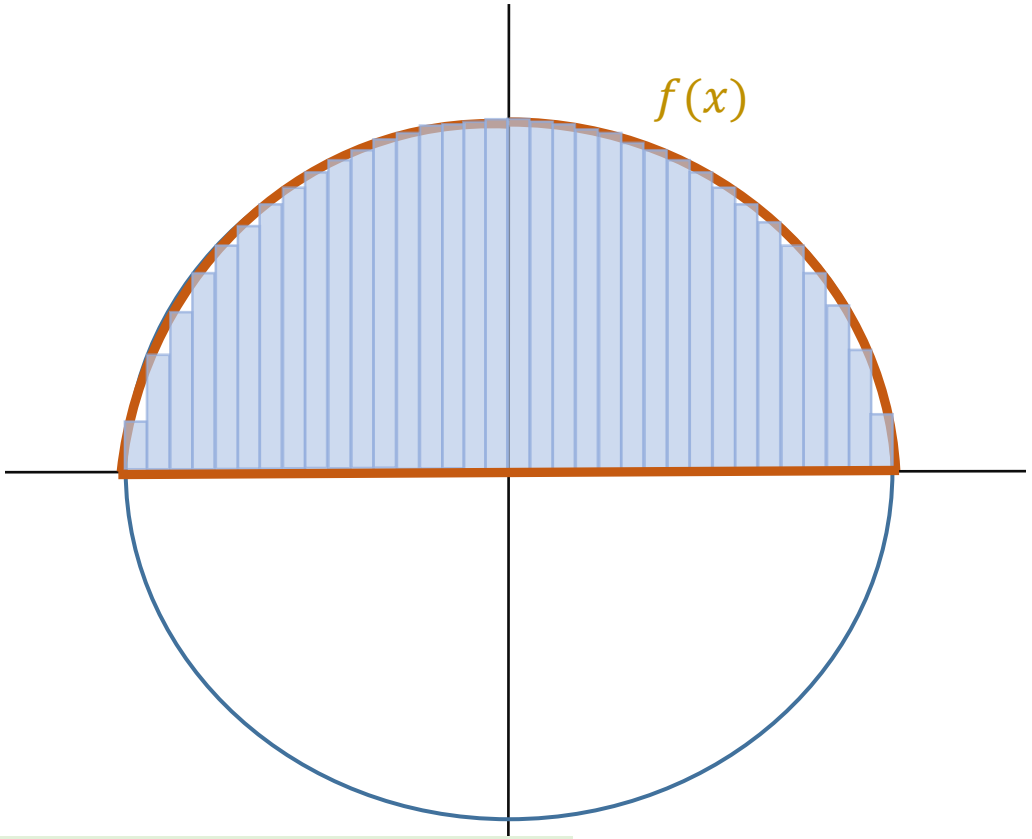
$n = 2000$

$A_{est} = 3.14155$



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)
```

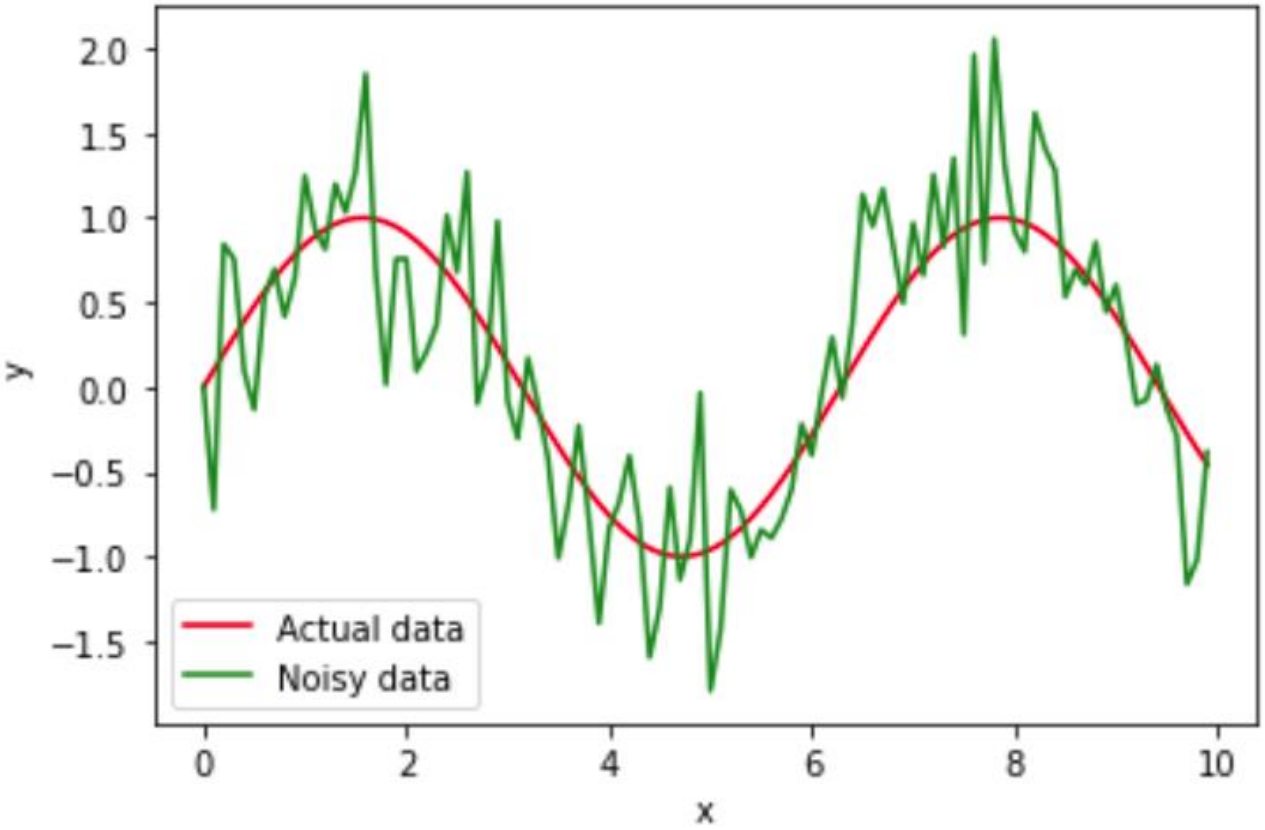
```python
delta_x = 0.01
n = int(2 / delta_x)

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
```

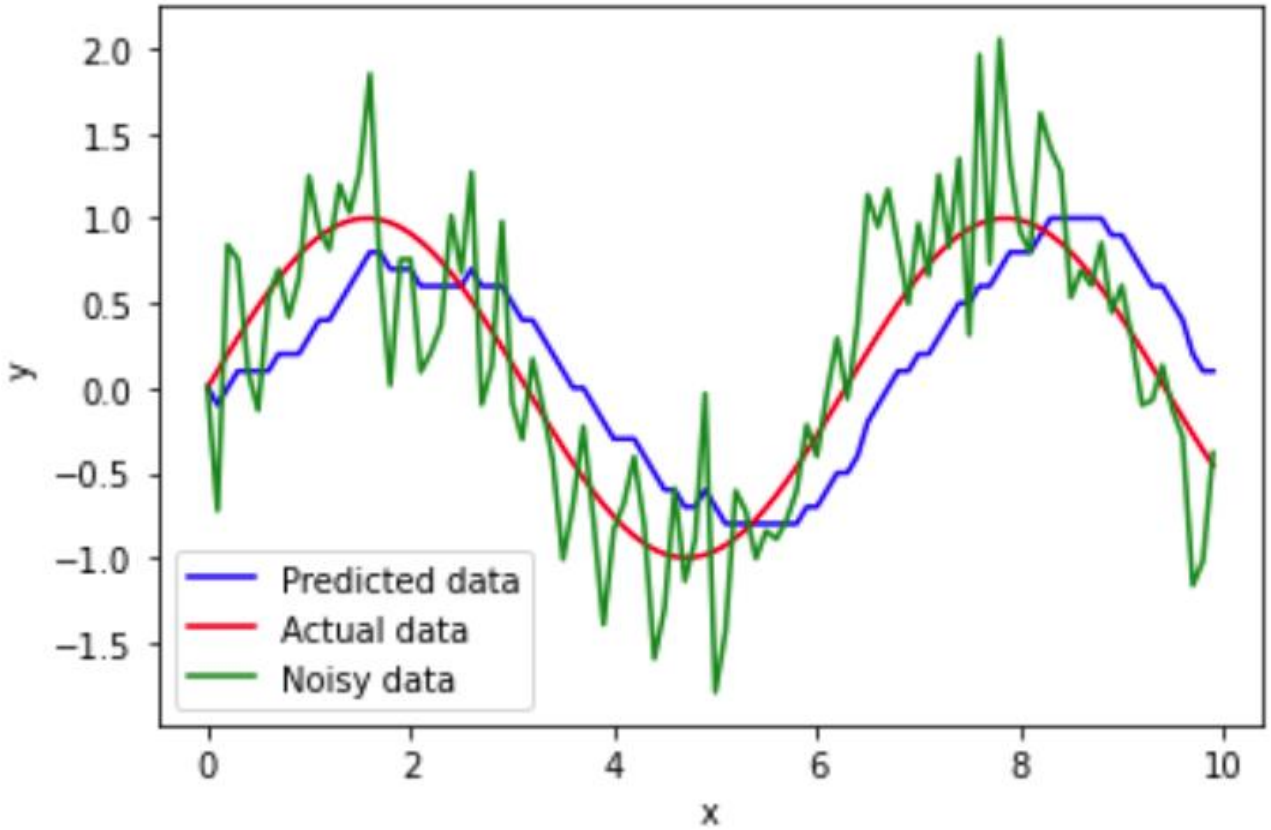
```python
print(half_area*2)
```

```
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

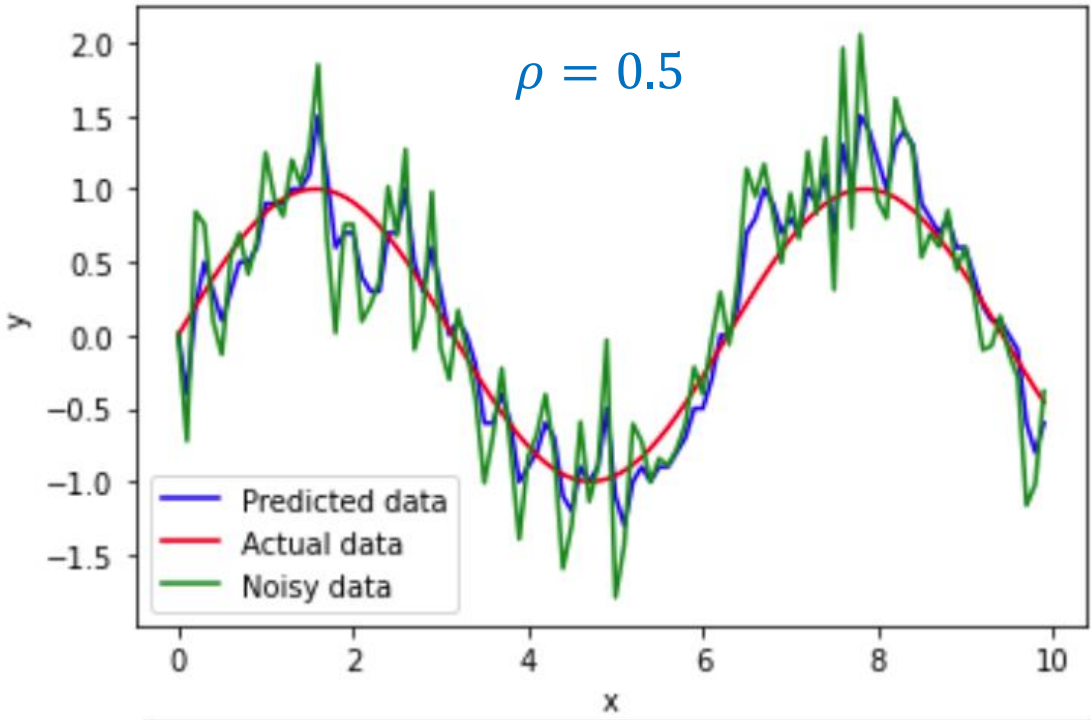
$\rho = 0.5$

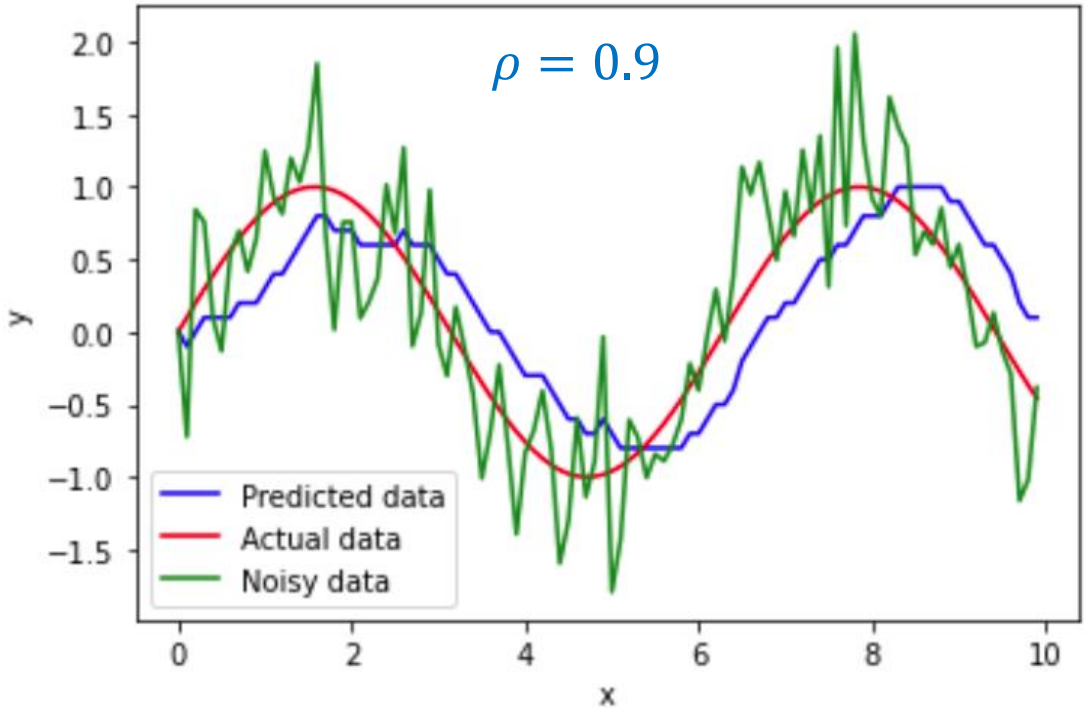
$$EMA_t = \rho EMA_{t-1} + (1 - \rho)s_t$$

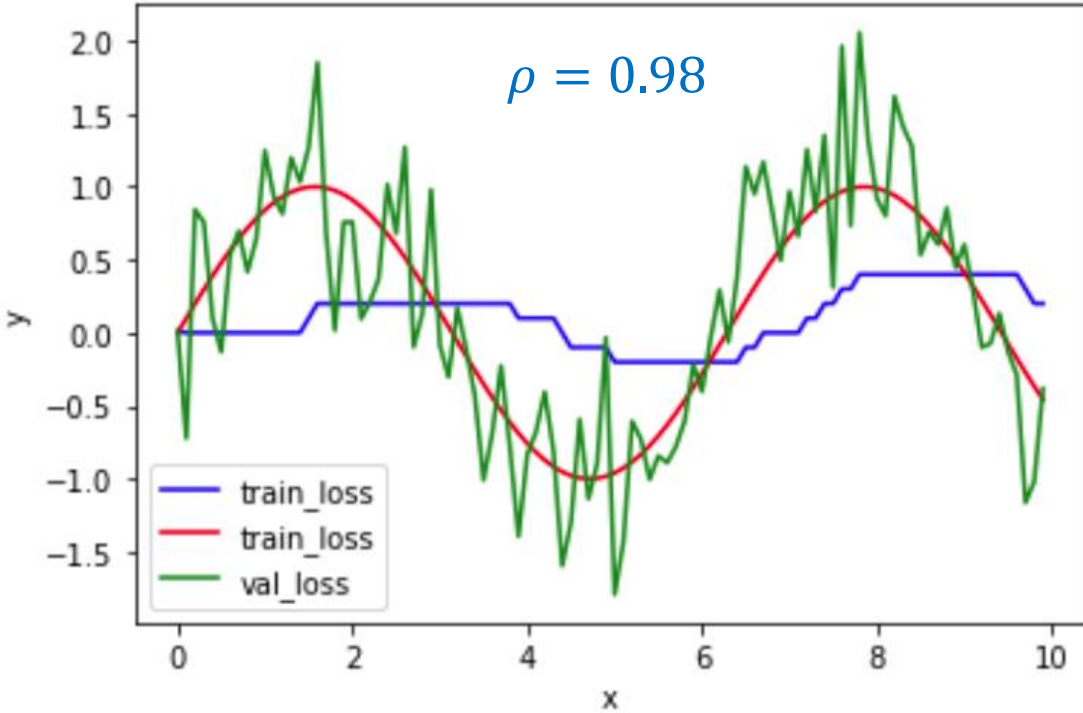
Example 2

❖ **Exponentially weighted averages**

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







AI VIET NAM
All-in-One 2024

Cheat Sheet – For Loop

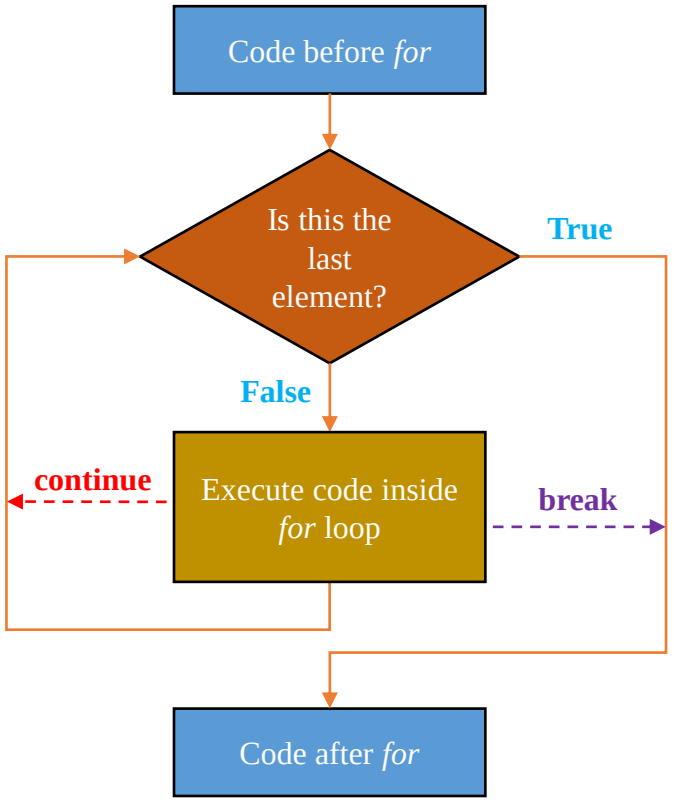
for syntax

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

$$x_0 = \frac{N}{2}; n = 0$$

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

$$n = n + 1$$

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 2024

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:

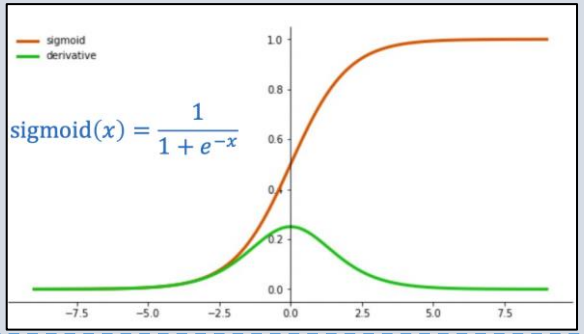
$$\pi \approx 4 \sum_{i=1}^{n} \frac{(-1)^{i+1}}{2i-1}$$

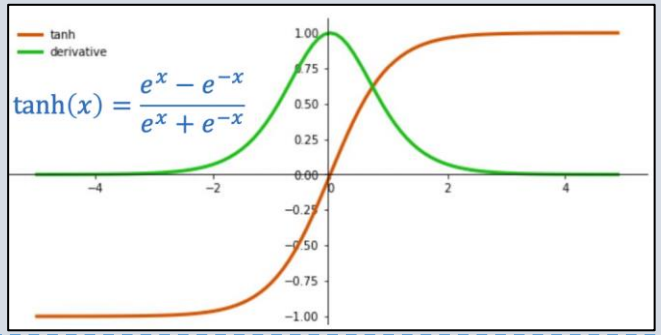
_ Nilakantha Series:

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

Activation Functions

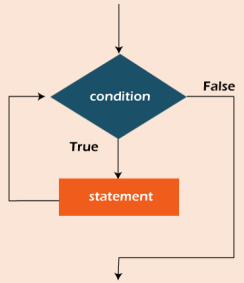
Map x values into smaller ranges





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])
```