

AI VIETNAM
All-in-One Course

Python

A Deeper Step

Quang-Vinh Dinh
Ph.D. in Computer Science

Outline

- **For Loop**
- **While Loop**
- **String**
- **File**
- **Case Studies**

Motivation

❖ Observation

Feature		Label	
	area	price	
	6.7	9.1	
	4.6	5.9	
	3.5	4.6	
	5.5	6.7	

House price data

Features			Label
TV	↕ Radio	↕ Newspaper	↕ Sales
230.1	37.8	69.2	22.1
44.5	39.3	45.1	10.4
17.2	45.9	69.3	12
151.5	41.3	58.5	16.5
180.8	10.8	58.4	17.9

Advertising data

For Loop

keyword

colon

```
# code trước for
for element in iterable:
    # code trong for
# code sau for
```

indentation

Iterables

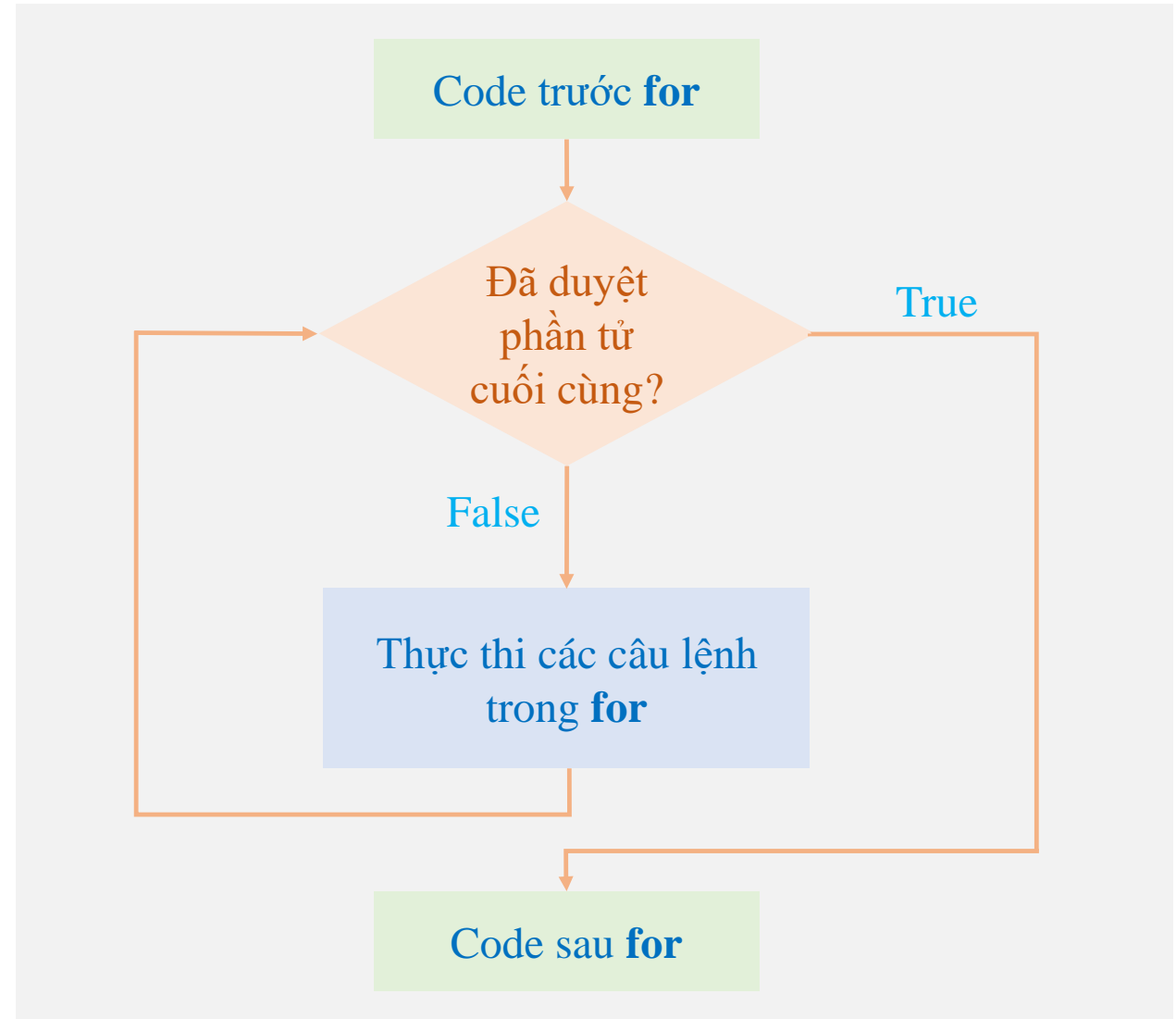
String

Tuple

List

Dictionary

range()



For Loop

Code trước for

Đã duyệt
phần tử
cuối cùng?

True

False

Thực thi các câu lệnh
trong for

Code sau for

```
1 # iterate a list
2
3 fruits = ['apple', 'banana', 'melon', 'peach']
4
5 for fruit in fruits:
6     print(fruit)
```

apple
banana
melon
peach

```
1 # iterate a dictionary
2
3 parameters = {'learning_rate': 0.1,
4               'optimizer': 'Adam',
5               'metric': 'Accuracy'}
6
7 for key in parameters:
8     print(key, parameters.get(key))
```

learning_rate 0.1
optimizer Adam
metric Accuracy

```
1 # iterate a tuple
2
3 fruits = ('apple', 'banana', 'melon')
4
5 for fruit in fruits:
6     print(fruit)
```

apple
banana
melon

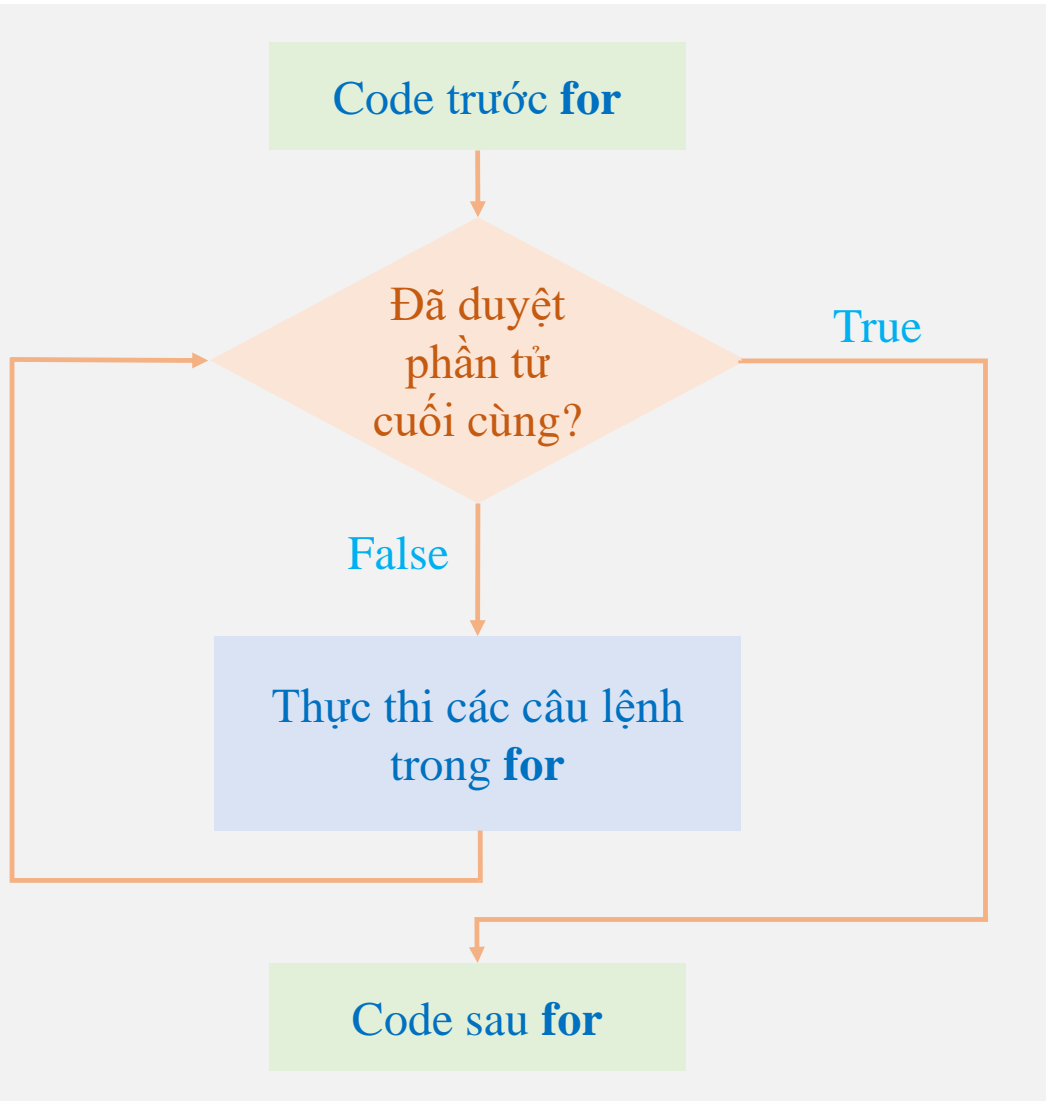
```
1 # iterate a string
2
3 greeting = 'Hello'
4
5 for char in greeting:
6     print(char)
```

H
e
l
l
o

```
1 # use range()
2
3 for i in range(5):
4     print(i)
```

0
1
2
3
4

For Loop



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

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



`0, 1, 2, 3, 4`

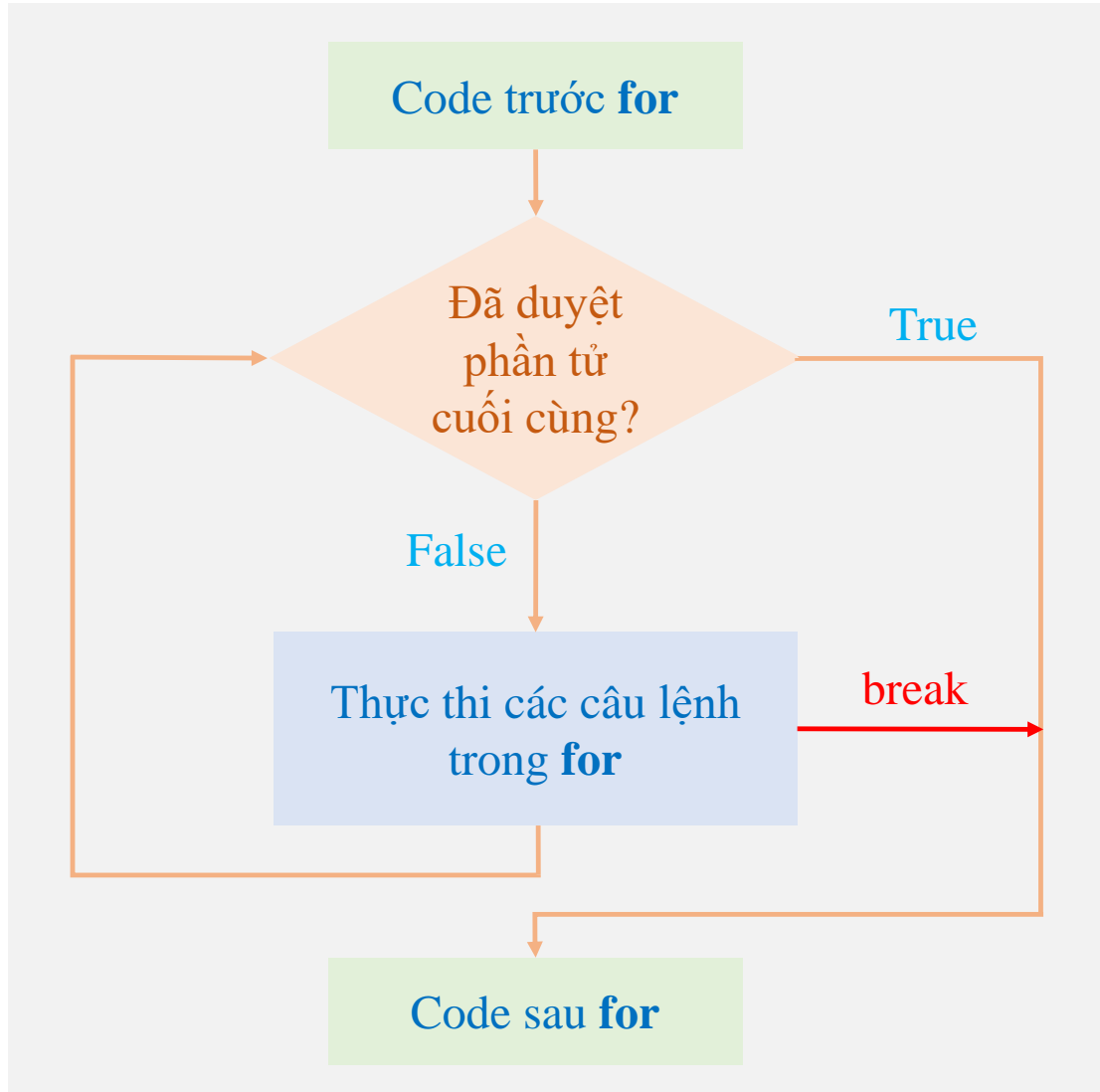
`range(5)`



`0, 1, 2, 3, 4`

Demo

For Loop

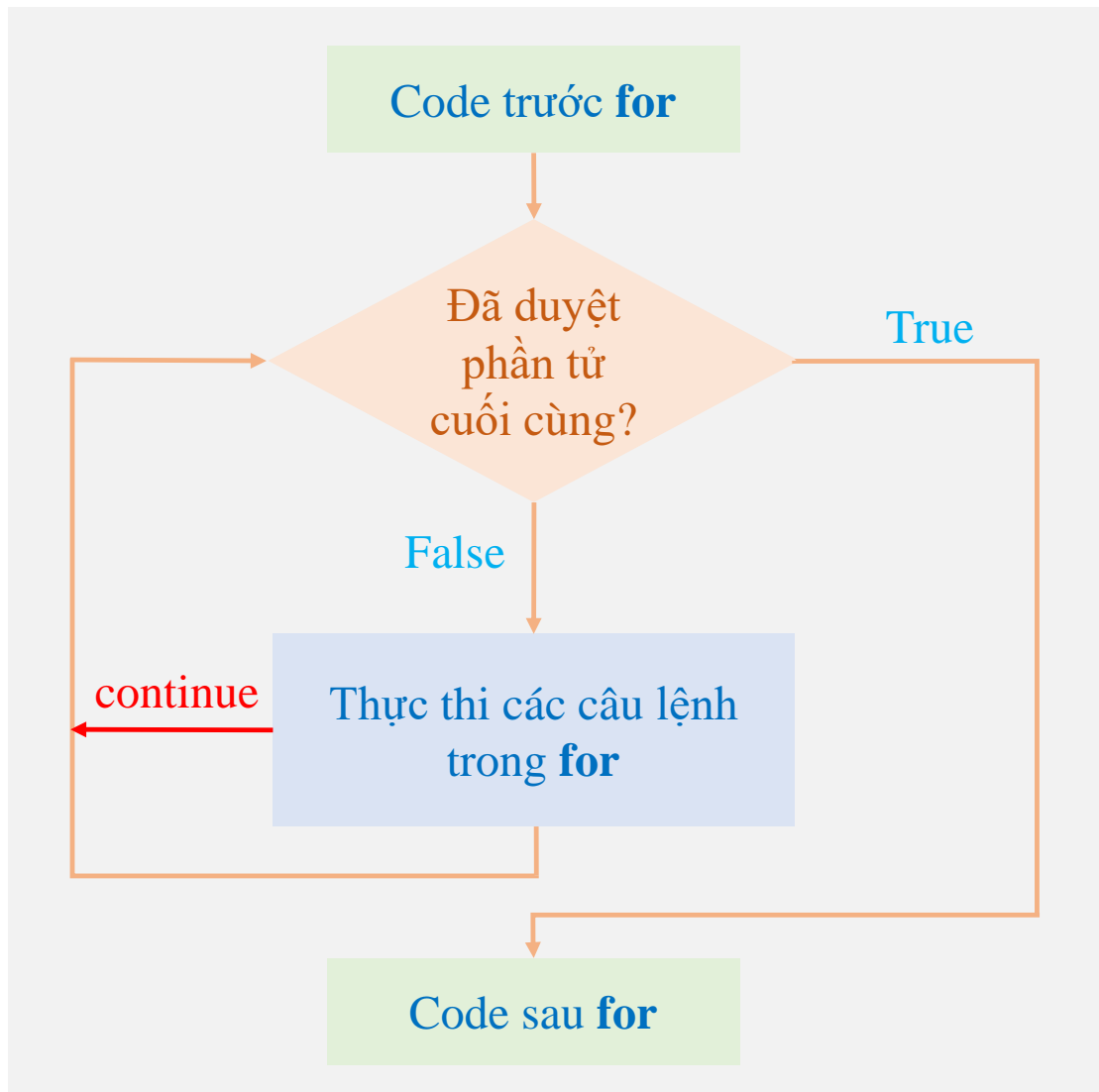


break keyword

```
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 thoát vòng lặp for này
6         break
7
8     # làm gì đó với i
9     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



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
```

Demo

Example

❖ PI estimation

Gregory-Leibniz Series

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

```
1 # Gregory-Leibniz Series
2
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

Nilakantha Series

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

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

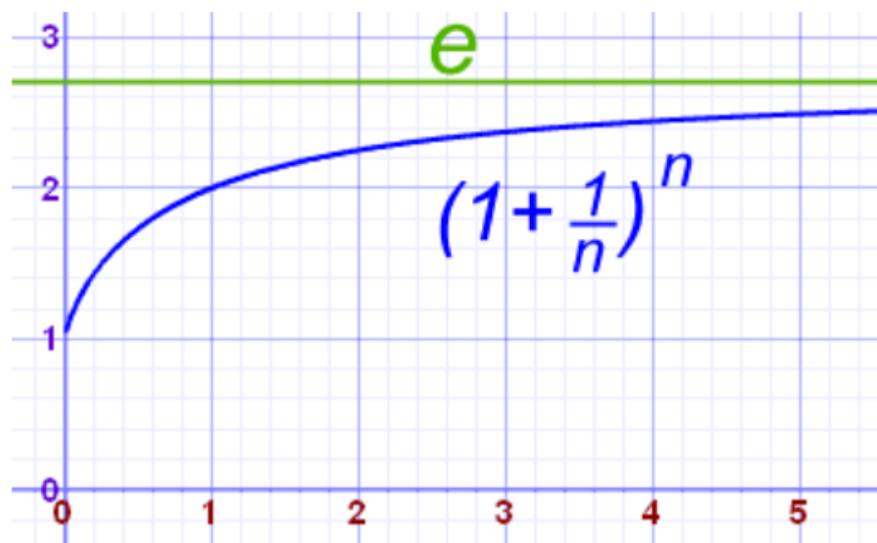
Estimated PI is 3.1415926533405423

Example

❖ Euler's number

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

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



n	$(1 + 1/n)^n$
1	2.00000
2	2.25000
5	2.48832
10	2.59374
100	2.70481
1,000	2.71692
10,000	2.71815
100,000	2.71827

Example

❖ Euler's number

$$e = 2.71828$$

Formula

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

1) Compute factorial

2) Compute sum

Example

❖ Euler's number

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

```
1. # aivietnam.ai
2.
3. # hàm tính giai thừa
4. def factorial(n):
5.     result = 1
6.
7.     for i in range(2, n+1):
8.         result = result*i
9.
10.    return result
11.
12. # hàm ước lượng số e
13. def estimate_e(n):
14.     result = 1
15.
16.     for i in range(1, n+1):
17.         result = result + 1/factorial(i)
18.
19.    return result
20.
21. # ước lượng số e với n = 10
22. print(estimate_e(10))
```

Example: Quadratic Root

❖ Compute quadratic root for the number N

Newton Method

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

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

$n = n + 1$

Compute $\sqrt{9}$

$$N = 9$$

$$\text{set } x_0 = \frac{9}{2} = 4.5$$

$$n = 0$$

$$n = 0$$

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

$$n = 1$$

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

$$n = 2$$

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

Example: Quadratic Root

❖ Compute quadratic root for the number N

Newton Method

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

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

$n = n + 1$

```
1 def compute_square_root(N, num_loops):
2     '''
3     This function aims to compute square root for the number N
4
5     N -- the number needs to take the square root
6     num_loops -- number of loops used for this optimization
7     '''
8
9     x_n = N/2.0
10
11     for i in range(num_loops):
12         x_np1 = (x_n + N/x_n) / 2.0
13         x_n = x_np1
14
15     return x_np1
16
17 print(compute_square_root(N=9, num_loops=10))
18 print(compute_square_root(N=2, num_loops=10))
```

```
3.0
1.414213562373095
```

Questions



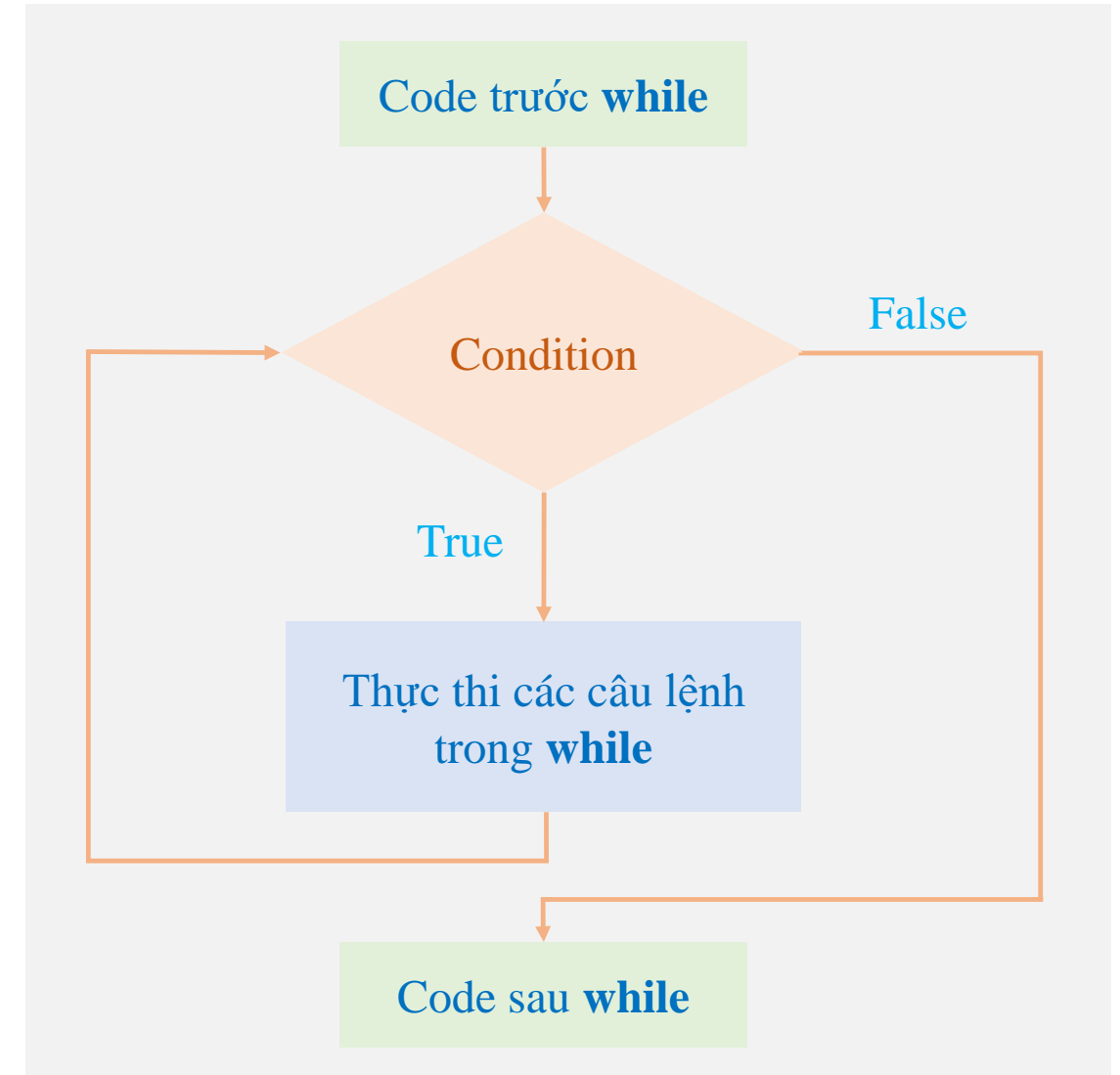
While Loop

keyword

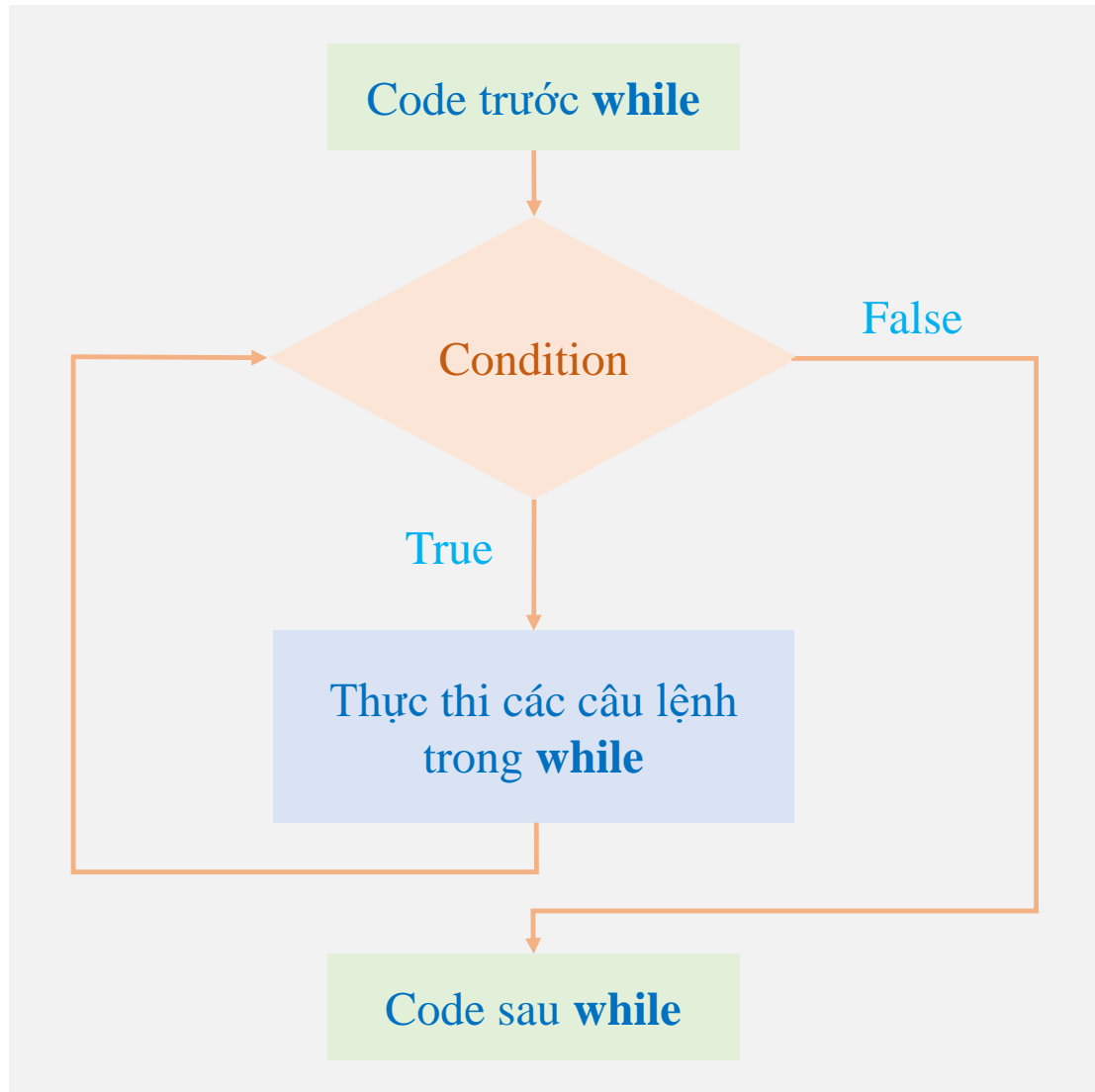
colon

```
# code trước while  
while condition:  
    # khối code trong while  
# code sau while
```

indentation



While Loop

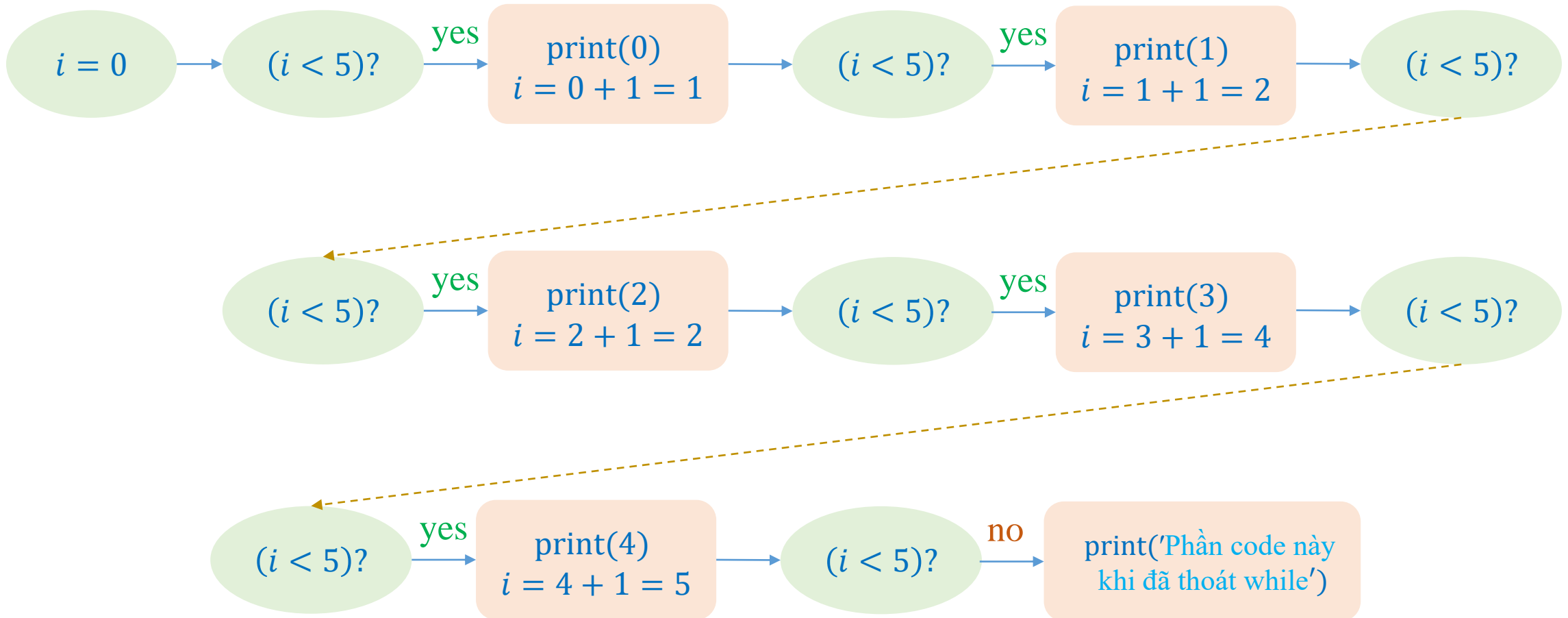


```
1  # tạo biến i
2  i = 0
3
4  # bắt đầu vòng lặp while
5  while i<5:
6      # code inside while
7      print(i)
8      i = i + 1
9
10 print('Phần code này khi đã thoát while')
```

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

While Loop

```
1 # tạo biến i
2 i = 0
3
4 # bắt đầu vòng lặp while
5 while i < 5:
6     # code inside while
7     print(i)
8     i = i + 1
9
10 print('Phần code này khi đã thoát while')
```



While Loop

while-True-break

```
1. import random
2.
3. # cho vòng lặp chạy vô tận
4. while True:
5.     # sinh số ngẫu nhiên
6.     num = random.randint(0,10)
7.     print('Số sinh ra có giá trị là', num)
8.
9.     # kiểm tra num có bằng 5 hay không?
10.    if num == 5:
11.        # nếu có thì thoát khỏi while
12.        break;
13.    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

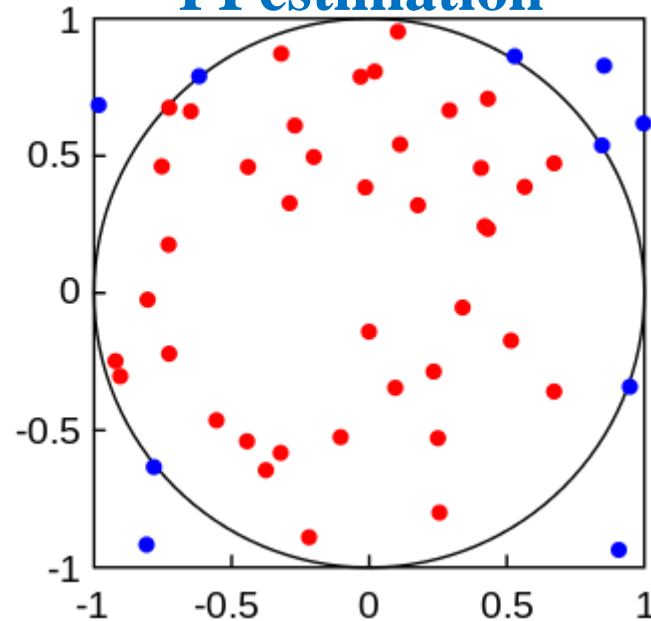
E estimation

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

Simulation of coin tossing



PI estimation



Compute quadratic root for the number N

Newton Method

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

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

$n = n + 1$

Outline

- **For Loop**
- **While Loop**
- **String**
- **File**
- **Case Studies**

String

❖ String creation

```
1 text1 = 'Tôi yêu AI VIET NAM'
2 print(text1)
```

Tôi yêu AI VIET NAM

```
1 text2 = "Tôi yêu AI VIET NAM"
2 print(text2)
```

Tôi yêu AI VIET NAM

```
1 text3 = '''Tôi yêu AI VIET NAM'''
2 print(text3)
```

Tôi yêu AI VIET NAM

```
1 text4 = """Tôi yêu AI VIET NAM"""
2 print(text4)
```

Tôi yêu AI VIET NAM

```
1 text1 = "Tôi yêu 'AI VIET NAM'"
2 print(text1)
```

Tôi yêu 'AI VIET NAM'

```
1 text2 = 'Tôi yêu "AI VIET NAM"'
2 print(text2)
```

Tôi yêu "AI VIET NAM"

```
1 text3 = """Chuỗi có kí tự đặc biệt " và ' bên trong"""
2 print(text3)
```

Chuỗi có kí tự đặc biệt " và ' bên trong

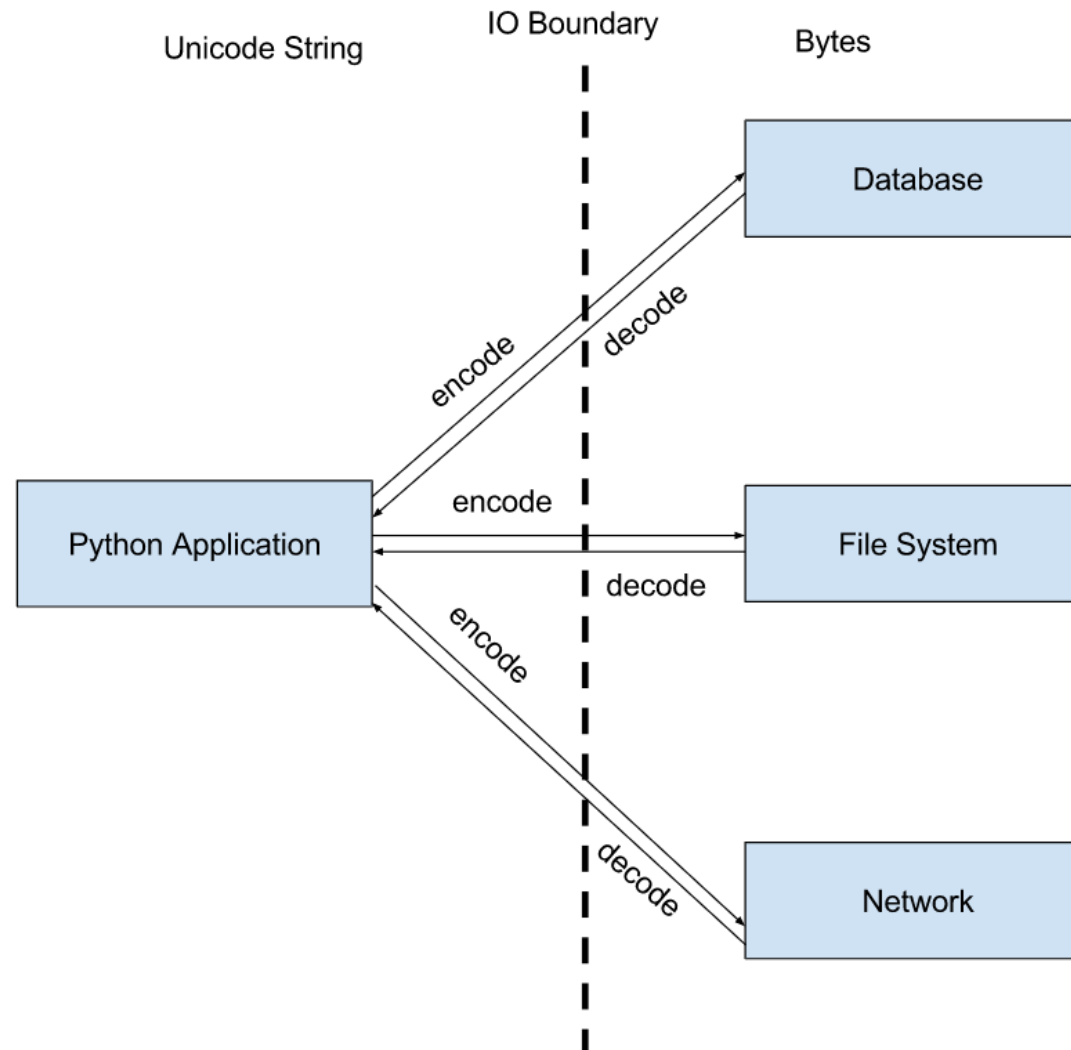
```
1 text4 = "Chuỗi có kí tự đặc biệt \" và \' bên trong"
2 print(text4)
```

Chuỗi có kí tự đặc biệt " và ' bên trong

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	\$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(72	48	110	H	104	68	150	h
9	9	11		41	29	51)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	
29	1D	35		61	3D	75	=	93	5D	135]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	

Variables

❖ Character type



ASCII Table

❖ American Standard code for information interchange

ASCII represents 128 characters

ASCII is stored as 8- bit byte

ASCII is not standardized

<https://upload.wikimedia.org/wikipedia/commons/d/dd/ASCII-Table.svg>

Decimal	Hexadecimal	Binary	Octal	Char
0	0	0	0	[NULL]
1	1	1	1	[START OF HEADING]
2	2	10	2	[START OF TEXT]
3	3	11	3	[END OF TEXT]
4	4	100	4	[END OF TRANSMISSION]
5	5	101	5	[ENQUIRY]
6	6	110	6	[ACKNOWLEDGE]
7	7	111	7	[BELL]
8	8	1000	10	[BACKSPACE]
9	9	1001	11	[HORIZONTAL TAB]
10	A	1010	12	[LINE FEED]
11	B	1011	13	[VERTICAL TAB]
12	C	1100	14	[FORM FEED]
13	D	1101	15	[CARRIAGE RETURN]
14	E	1110	16	[SHIFT OUT]
15	F	1111	17	[SHIFT IN]
16	10	10000	20	[DATA LINK ESCAPE]
17	11	10001	21	[DEVICE CONTROL 1]
18	12	10010	22	[DEVICE CONTROL 2]
19	13	10011	23	[DEVICE CONTROL 3]
20	14	10100	24	[DEVICE CONTROL 4]
21	15	10101	25	[NEGATIVE ACKNOWLEDGE]
22	16	10110	26	[SYNCHRONOUS IDLE]
23	17	10111	27	[ENG OF TRANS. BLOCK]
24	18	11000	30	[CANCEL]
25	19	11001	31	[END OF MEDIUM]
26	1A	11010	32	[SUBSTITUTE]
27	1B	11011	33	[ESCAPE]
28	1C	11100	34	[FILE SEPARATOR]
29	1D	11101	35	[GROUP SEPARATOR]
30	1E	11110	36	[RECORD SEPARATOR]
31	1F	11111	37	[UNIT SEPARATOR]
32	20	100000	40	[SPACE]

ASCII Table

Decimal	Hexadecimal	Binary	Octal	Char
33	21	100001	41	!
34	22	100010	42	"
35	23	100011	43	#
36	24	100100	44	\$
37	25	100101	45	%
38	26	100110	46	&
39	27	100111	47	'
40	28	101000	50	(
41	29	101001	51)
42	2A	101010	52	*
43	2B	101011	53	+
44	2C	101100	54	,
45	2D	101101	55	-
46	2E	101110	56	.
47	2F	101111	57	/

Decimal	Hexadecimal	Binary	Octal	Char
48	30	110000	60	0
49	31	110001	61	1
50	32	110010	62	2
51	33	110011	63	3
52	34	110100	64	4
53	35	110101	65	5
54	36	110110	66	6
55	37	110111	67	7
56	38	111000	70	8
57	39	111001	71	9
58	3A	111010	72	:
59	3B	111011	73	;
60	3C	111100	74	<
61	3D	111101	75	=
62	3E	111110	76	>
63	3F	111111	77	?
64	40	1000000	100	@

ASCII Table

Decimal	Hexadecimal	Binary	Octal	Char
65	41	1000001	101	A
66	42	1000010	102	B
67	43	1000011	103	C
68	44	1000100	104	D
69	45	1000101	105	E
70	46	1000110	106	F
71	47	1000111	107	G
72	48	1001000	110	H
73	49	1001001	111	I
74	4A	1001010	112	J
75	4B	1001011	113	K
76	4C	1001100	114	L
77	4D	1001101	115	M
78	4E	1001110	116	N
79	4F	1001111	117	O
80	50	1010000	120	P
81	51	1010001	121	Q
82	52	1010010	122	R
83	53	1010011	123	S
84	54	1010100	124	T
85	55	1010101	125	U
86	56	1010110	126	V
87	57	1010111	127	W
88	58	1011000	130	X
89	59	1011001	131	Y
90	5A	1011010	132	Z
91	5B	1011011	133	[
92	5C	1011100	134	\
93	5D	1011101	135]
94	5E	1011110	136	^
95	5F	1011111	137	_

Decimal	Hexadecimal	Binary	Octal	Char
96	60	1100000	140	`
97	61	1100001	141	a
98	62	1100010	142	b
99	63	1100011	143	c
100	64	1100100	144	d
101	65	1100101	145	e
102	66	1100110	146	f
103	67	1100111	147	g
104	68	1101000	150	h
105	69	1101001	151	i
106	6A	1101010	152	j
107	6B	1101011	153	k
108	6C	1101100	154	l
109	6D	1101101	155	m
110	6E	1101110	156	n
111	6F	1101111	157	o
112	70	1110000	160	p
113	71	1110001	161	q
114	72	1110010	162	r
115	73	1110011	163	s
116	74	1110100	164	t
117	75	1110101	165	u
118	76	1110110	166	v
119	77	1110111	167	w
120	78	1111000	170	x
121	79	1111001	171	y
122	7A	1111010	172	z
123	7B	1111011	173	{
124	7C	1111100	174	
125	7D	1111101	175	}
126	7E	1111110	176	~
127	7F	1111111	177	[DEL]

String

❖ Insert into a string

```
1 value1 = "AI VIETNAM"
2 print("Hello %s. Have a nice day!" % value1)
3 print("Hello %s. Have a nice day!" % (value1))
4
5 value2 = "Hi"
6 print("***%s %s***" % (value2, value1))
```

```
Hello AI VIETNAM. Have a nice day!
Hello AI VIETNAM. Have a nice day!
***Hi AI VIETNAM***
```

```
1 iteration = 5
2 loss = 48.2
3
4 print("Loss at the %dth iteration is %f" % (iteration, loss))
```

```
Loss at the 5th iteration is 48.200000
```

```
1 value1 = "Hello"
2 value2 = "AIVIETNAM"
3
4 s1 = "{n1}, {n2}!".format(n1=value1, n2=value2)
5 s2 = "{n1}, {n2}!".format(n2=value2, n1=value1)
6
7 print(s1)
8 print(s2)
```

```
Hello, AIVIETNAM!
Hello, AIVIETNAM!
```

```
1 name = "John"
2 age = 21
3 print(f'Hello {name}. Are you {age} years old?')
```

```
Hello John. Are you 21 years old?
```

String

❖ + and * operators

```
1 s1 = 'Tôi thích '  
2 s2 = 'AI!'  
3  
4 s3 = s1 + s2  
5 s4 = s3*2  
6  
7 print(s3)  
8 print(s4)
```

```
Tôi thích AI!  
Tôi thích AI!Tôi thích AI!
```

❖ Logic operators

```
1 s1 = 'a'  
2 s2 = 'b'  
3  
4 print(s1 == s1)  
5 print(s1 == s2)  
6  
7 print(s1 != s2)  
8 print(s1 < s2)  
9 print(s1 > s2)
```

```
True  
False  
True  
True  
False
```

String

isdigit(): Kiểm tra xem string gồm các kí tự số

```
1 # Kiểm tra xem string gồm các chữ số
2 print("10".isdigit())
3 print("abc".isdigit())
```

```
True
False
```

isalpha(): Kiểm tra xem string chỉ được tạo từ các kí tự chữ cái

```
1 # Kiểm tra xem string chỉ được tạo
2 # từ các kí tự chữ cái
3 print("10".isalpha())
4 print("abc".isalpha())
```

```
False
True
```

islower(): Kiểm tra xem string với tất cả các kí tự ở dạng chữ thường

```
1 # Kiểm tra xem string với tất cả
2 # các kí tự ở dạng chữ thường
3 print("ab".islower())
4 print("Ab".islower())
```

```
True
False
```

isupper(): Kiểm tra xem string với tất cả các kí tự ở dạng chữ hoa

```
1 # Kiểm tra xem string với tất cả
2 # các kí tự ở dạng chữ hoa
3 print("Ab".isupper())
4 print("AB".isupper())
```

```
False
True
```

String

isspace(): Kiểm tra xem string chỉ khoảng trắng

```
1 # Kiểm tra xem string chỉ là khoảng trắng
2 print("").isspace()
3 print(" ".isspace())
4 print("   ".isspace())
5 print("   abc   ".isspace())
```

```
False
True
True
False
```

count(): để đếm số kí tự xuất hiện trong một string

len(): để tính chiều dài của một string

```
1 s = "AI VIET NAM"
2
3 print(s.count('A'))
4 print(s.count('a'))
5 print(len(mystr))
```

```
2
0
11
```

istitle(): Kiểm tra xem string có bắt đầu bằng chữ in hoa

```
1 # Kiểm tra xem string có bắt đầu
2 # bằng chữ in hoa
3 print("Hello".istitle())
4 print("HELLO".istitle())
5 print("hello".istitle())
```

```
True
False
False
```

title()

```
1 # Chuyển đổi kí tự đầu từng từ trong một chuỗi
2 # thành kí tự hoa
3 mystr = "Đây là bài học của AI VIET NAM"
4 mystr.title()
```

```
'Đây Là Bài Học Của Ai Viet Nam'
```

swapcase()

```
1 # Chuyển đổi các kí tự từ chữ thường sang chữ
2 # hoa và ngược lại
3 mystr = "Đây là bài học của AI VIET NAM"
4 mystr.swapcase()
```

```
'đÂY LÀ BÀI HỌC CỦA ai viet nam'
```

capitalize()

```
1 # Chuyển đổi chữ cái đầu tiên của một chuỗi thành
2 # chữ hoa và các kí tự sau thành chữ thường
3 mystr = "Đây là bài học của AI VIET NAM"
4 mystr.capitalize()
```

```
'Đây là bài học của ai viet nam'
```

strip()

```
1 # Loại bỏ khoảng trắng ở cả hai đầu của chuỗi
2 mystr = "  Đây là bài học của AI VIET NAM  "
3 mystr.strip()
```

```
'Đây là bài học của AI VIET NAM'
```

replace()

```
1 # Thay thế chuỗi
2 mystr = "Đây là bài học của AI VIET NAM"
3 mystr.replace('AI VIET NAM', 'AIVIETNAM')
```

```
'Đây là bài học của AIVIETNAM'
```

partition()

```
1 # Tách chuỗi
2 mystr = "Đây là bài học của AI VIET NAM"
3 mystr.partition('của')
```

```
('Đây là bài học ', 'của', ' AI VIET NAM')
```


upper()

```
1 # Chuyển đổi tất cả các kí tự trong
2 # chuỗi thành chữ hoa
3 mystr = "Đây là bài học của AI VIET NAM"
4 mystr.upper()
```

```
'ĐÂY LÀ BÀI HỌC CỦA AI VIET NAM'
```

lower()

```
1 # Chuyển đổi các kí tự sang chữ thường
2 mystr = "Đây là bài học của AI VIET NAM"
3 mystr.lower()
```

```
'đây là bài học của ai viet nam'
```

center()

```
1 # Chỉnh chuỗi ở trung tâm
2 # và chiều dài của chuỗi là 40
3 mystr = "Đây là bài học của AI VIET NAM"
4 mystr.center(40, '-')
```

```
'-----Đây là bài học của AI VIET NAM-----'
```

endswith()

```
1 # Kiểm tra phần kết thúc của một string
2 mystr = "Đây là bài học của AI VIET NAM"
3 print(mystr.endswith('VIET NAM'))
4 print(mystr.endswith('bài học'))
```

```
True
False
```

startswith()

```
1 # Kiểm tra phần đầu của một string
2 mystr = "Đây là bài học của AI VIET NAM"
3 print(mystr.startswith('Đây'))
4 print(mystr.startswith('AI'))
```

```
True
False
```

find()

```
1 # Tìm vị trí của một string trong một string khác
2 # trả về -1 nếu không tìm thấy
3 mystr = "Đây là bài học của AI VIET NAM"
4 print(mystr.find('học'))
5 print(mystr.find('hello'))
```

```
11
-1
```

String

❖ Common Error

`index()`: Tìm vị trí xuất hiện của string s1 trong string s2.

Lỗi `ValueError` được trả về trong trường hợp s1 không được tìm thấy trong s2.

```
1 # Tìm vị trí của một string trong một string khác
2 # trả về ValueError nếu không tìm thấy
3 mystr = "Đây là bài học của AI VIET NAM"
4 print(mystr.index('học'))
5 print(mystr.index('hello'))
```

11

```
-----
ValueError                                Traceback (most recent call last)
<ipython-input-89-5077ce643cbd> in <module>
      3 mystr = "Đây là bài học của AI VIET NAM"
      4 print(mystr.index('học'))
----> 5 print(mystr.index('hello'))

ValueError: substring not found
```

String

❖ Common Error

```
1. # aivietnam.ai
2. # Lỗi tìm vị trí của một string không có trong một string khác
3.
4. my_string = "Đây là bài học của AI VIETNAM"
5. my_string.index("hello")
```

```
-----
ValueError                                Traceback (most recent call last)
<ipython-input-1-9d5c7da27233> in <module>
      3
      4 my_string = "Đây là bài học của AI VIETNAM"
----> 5 my_string.index("hello")

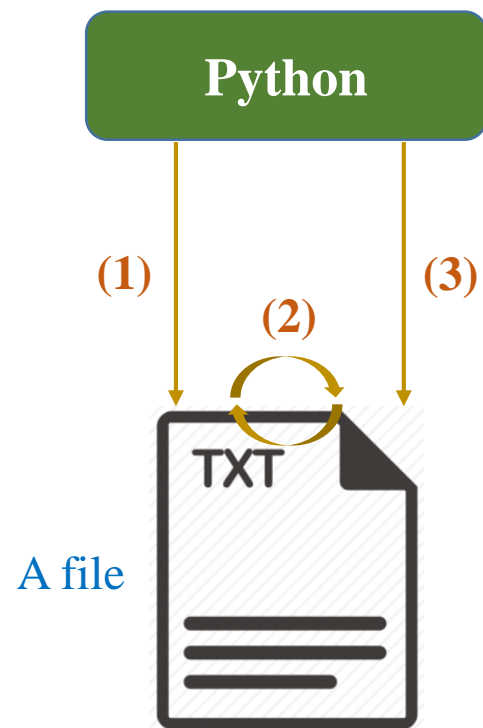
ValueError: substring not found
```

Outline

- **For Loop**
- **While Loop**
- **String**
- **File**
- **Case Studies**

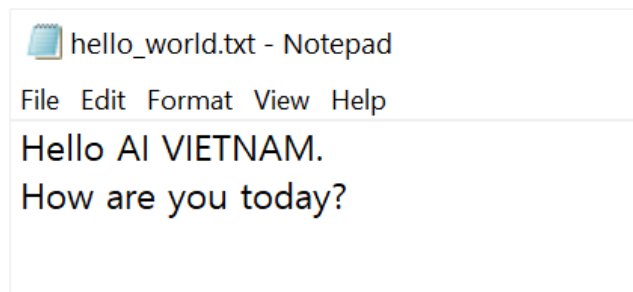
File

❖ Typical procedure



Read from a file (already exist)

- | | |
|-----|-----------------------------------|
| (1) | <code>open(file_path, 'r')</code> |
| (2) | <code>read()</code> |
| (3) | <code>close()</code> |



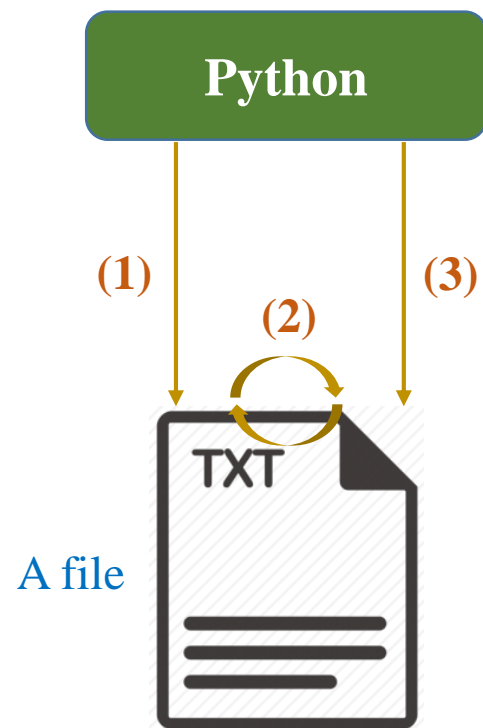
```
1 # kết nối với file
2 a_file = open('hello_world.txt', 'r')
3
4 # read content as string
5 data = a_file.read()
6
7 print(type(data))
8 print(data)
9
10 # Đóng kết nối với file
11 a_file.close()
```

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

- | | |
|-----|-------------------------|
| (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 content from a file as lines

hello_world.txt - Notepad

File Edit Format View Help

Hello AI VIETNAM.
How are you today?

- | | |
|-----|-----------------------------------|
| (1) | <code>open(file_path, 'r')</code> |
| (2) | <code>readlines()</code> |
| (3) | <code>close()</code> |

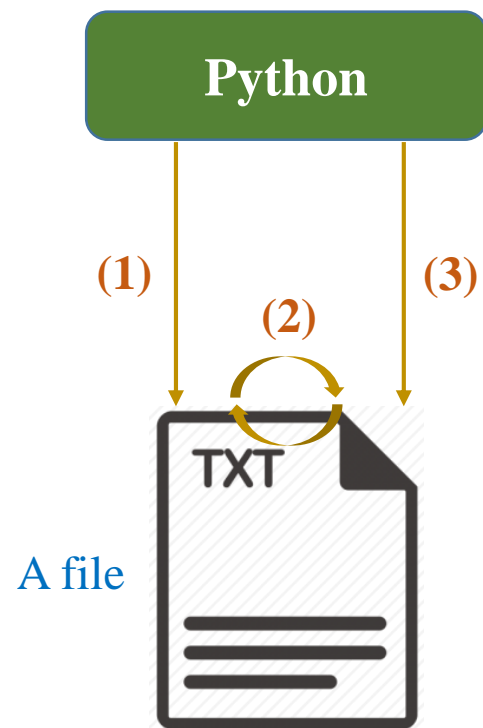
```
1 # kết nối với file
2 a_file = open('hello_world.txt', 'r')
3
4 # read content as string
5 lines = a_file.readlines()
6 for line in lines:
7     print(line)
8
9 # Đóng kết nối với file
10 a_file.close()
```

Hello AI VIETNAM.

How are you today?

File

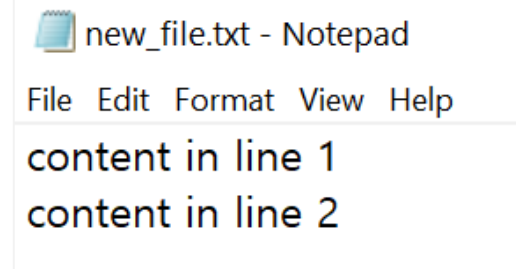
❖ Typical procedure



Write to a file (not exist)

- | | |
|-----|-----------------------------------|
| (1) | <code>open(file_path, 'w')</code> |
| (2) | <code>write()</code> |
| (3) | <code>close()</code> |

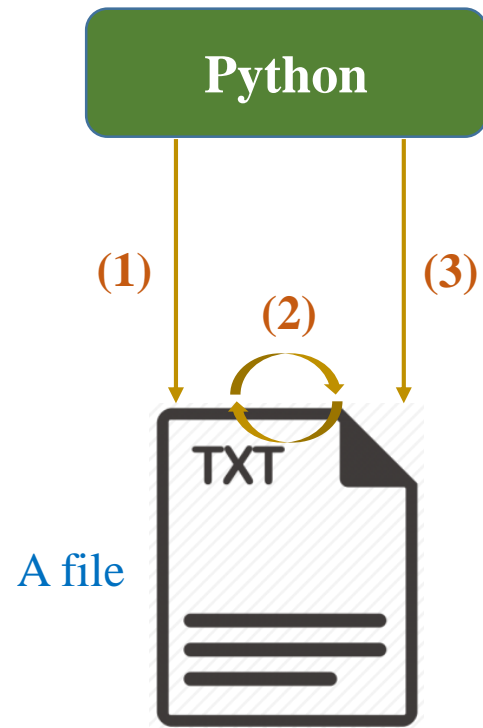
```
1 # kết nối với file
2 a_file = open('new_file.txt', 'w')
3
4 text1 = 'content in line 1 \n'
5 a_file.write(text1)
6
7 text2 = 'content in line 2 \n'
8 a_file.write(text2)
9
10 # Đóng kết nối với file
11 a_file.close()
```



- | | |
|-----|-------------------------|
| (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 |

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

- | | |
|-----|-----------------------------------|
| (1) | <code>open(file_path, 'a')</code> |
| (2) | <code>write()</code> |
| (3) | <code>close()</code> |

new_file.txt - Notepad

File Edit Format View Help

content in line 1
content in line 2

```
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()
```

new_file.txt - Notepad

File Edit Format View Help

content in line 1
content in line 2
content in line 3

```
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()
```

non_existing_file.txt - Notepad

File Edit Format View Help

content in line 3

File

❖ Useful functions

Check if a file exists

```
1 import os
2
3 file_path1 = 'my_file.txt'
4 check1 = os.path.exists(file_path1)
5 print('my_file.txt có tồn tại không?', check1)
6
7 file_path2 = 'non_existence_file.txt'
8 check2 = os.path.exists(file_path2)
9 print('non_existence_file.txt có tồn tại không?', check2)
```

```
my_file.txt có tồn tại không? True
non_existence_file.txt có tồn tại không? False
```

String splitting

```
1 string1 = '001,john,12-06-1999'
2 tokens = string1.split(',')
3
4 for token in tokens:
5     print(token)
```

```
001
john
12-06-1999
```

String joining

```
1 student_info = ['001', 'john', '12-06-1999']
2 string_joined = ','.join(student_info)
3
4 print(string_joined)
```

```
001,john,12-06-1999
```

File

❖ 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



```
1 # open a file
2 a_file = open('hello_world.txt', 'w')
3
4 # write data to file
5 text3 = 'writing line \n'
6 a_file.write(text3)
```



```
1 # open a file
2 a_file = open('hello_world.txt', 'w')
3
4 # write data to file
5 text3 = 'writing line \n'
6 a_file.write(text3)
7
8 # close the file
9 a_file.close()
```

File

❖ with keyword

```
1 # open a file
2 a_file = open('hello_world.txt', 'w')
3
4 # write data to file
5 text3 = 'writing line \n'
6 a_file.write(text3)
7
8 # close the file
9 a_file.close()
```

```
1 # using with
2
3 with open('hello_world.txt', 'w') as file:
4     file.write('writing line \n')
```

File

❖ Example

```
1 path = 'E:\\Data\\AICourse-2021\\1.BasicPython\\file\\hello_world.txt'
2 print(path)
3
4 with open(path, 'r') as file:
5     lines = file.readlines()
6     print(lines)
```

```
E:\Data\AICourse-2021\1.BasicPython\file\hello_world.txt
['hello world!']
```

```
1 path = r'E:\Data\AICourse-2021\1.BasicPython\file\hello_world.txt'
2 print(path)
3
4 with open(path, 'r') as file:
5     lines = file.readlines()
6     print(lines)
```

```
E:\Data\AICourse-2021\1.BasicPython\file\hello_world.txt
['hello world!']
```

File

❖ Example

	area	price	
	6.7	9.1	
	4.6	5.9	
	3.5	4.6	
	5.5	6.7	

data.csv - Notepad			
File Edit Format View Help			
area,price			
6.7,9.1			
4.6,5.9			
3.5,4.6			
5.5,6.7			

```
1 # kết nối với file
2 file = open('data.csv','r')
3
4 # read lines
5 lines = file.readlines()
6
7 # in các dòng
8 for line in lines:
9     print(line)
10
11 # Đóng kết nối với file
12 file.close()
```

area,price

6.7,9.1

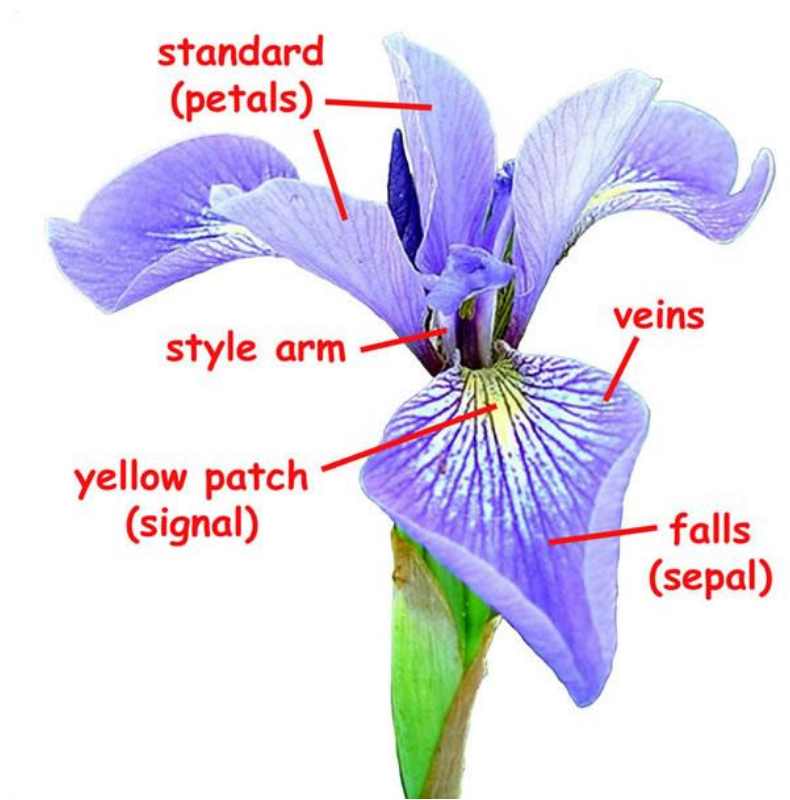
4.6,5.9

3.5,4.6

5.5,6.7

❖ Text data
❖ IRIS data

sepal_length	↕ sepal_width	↕ petal_length	↕ petal_width	↕ species	↕
5.1	3.5	1.4	0.2	Iris-setosa	
4.9	3	1.4	0.2	Iris-setosa	
4.7	3.2	1.3	0.2	Iris-setosa	
4.6	3.1	1.5	0.2	Iris-setosa	
5	3.6	1.4	0.2	Iris-setosa	
5.4	3.9	1.7	0.4	Iris-setosa	
4.6	3.4	1.4	0.3	Iris-setosa	
5	3.4	1.5	0.2	Iris-setosa	
4.4	2.9	1.4	0.2	Iris-setosa	
4.9	3.1	1.5	0.1	Iris-setosa	



File

❖ Example

Petal_Length	Petal_Width	Label
1.4	0.2	0
1.5	0.2	0
3	1.1	1
4.1	1.3	1

```
iris_demo.csv - Notepad
File Edit Format View Help
Petal_Length, Petal_Width, Label
1.4,0.2,0.0
1.5,0.2,0.0
3.0,1.1,1.0
4.1,1.3,1.0
```

```
1 # kết nối với file
2 file = open('iris_demo.csv','r')
3
4 # read lines
5 lines = file.readlines()
6
7 # in các dòng
8 for line in lines:
9     print(line)
10
11 # Đóng kết nối với file
12 file.close()
```

```
Petal_Length, Petal_Width, Label

1.4,0.2,0.0

1.5,0.2,0.0

3.0,1.1,1.0

4.1,1.3,1.0
```


File

❖ Example

```
Id,SepalLength,SepalWidth,PetalLength,PetalWidth,Species
1,5.1,3.5,1.4,0.2,Iris-setosa
2,4.9,3.0,1.4,0.2,Iris-setosa
3,4.7,3.2,1.3,0.2,Iris-setosa
4,4.6,3.1,1.5,0.2,Iris-setosa
5,5.0,3.6,1.4,0.2,Iris-setosa
6,5.4,3.9,1.7,0.4,Iris-setosa
7,4.6,3.4,1.4,0.3,Iris-setosa
8,5.0,3.4,1.5,0.2,Iris-setosa
9,4.4,2.9,1.4,0.2,Iris-setosa
10,4.9,3.1,1.5,0.1,Iris-setosa
11,7.0,3.2,4.7,1.4,Iris-versicolor
12,6.4,3.2,4.5,1.5,Iris-versicolor
13,6.9,3.1,4.9,1.5,Iris-versicolor
14,5.5,2.3,4.0,1.3,Iris-versicolor
15,6.5,2.8,4.6,1.5,Iris-versicolor
16,5.7,2.8,4.5,1.3,Iris-versicolor
17,6.3,3.3,4.7,1.6,Iris-versicolor
18,4.9,2.4,3.3,1.0,Iris-versicolor
19,6.6,2.9,4.6,1.3,Iris-versicolor
20,5.2,2.7,3.9,1.4,Iris-versicolor
21,6.3,3.3,6.0,2.5,Iris-virginica
22,5.8,2.7,5.1,1.9,Iris-virginica
```

Id	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
8	5	3.4	1.5	0.2	Iris-setosa
9	4.4	2.9	1.4	0.2	Iris-setosa
10	4.9	3.1	1.5	0.1	Iris-setosa
11	7	3.2	4.7	1.4	Iris-versicolor
12	6.4	3.2	4.5	1.5	Iris-versicolor
13	6.9	3.1	4.9	1.5	Iris-versicolor
14	5.5	2.3	4	1.3	Iris-versicolor
15	6.5	2.8	4.6	1.5	Iris-versicolor
16	5.7	2.8	4.5	1.3	Iris-versicolor
17	6.3	3.3	4.7	1.6	Iris-versicolor
18	4.9	2.4	3.3	1	Iris-versicolor
19	6.6	2.9	4.6	1.3	Iris-versicolor
20	5.2	2.7	3.9	1.4	Iris-versicolor
21	6.3	3.3	6	2.5	Iris-virginica
22	5.8	2.7	5.1	1.9	Iris-virginica
23	7.1	3	5.9	2.1	Iris-virginica
24	6.3	2.9	5.6	1.8	Iris-virginica
25	6.5	3	5.8	2.2	Iris-virginica
26	7.6	3	6.6	2.1	Iris-virginica
27	4.9	2.5	4.5	1.7	Iris-virginica
28	7.3	2.9	6.3	1.8	Iris-virginica
29	6.7	2.5	5.8	1.8	Iris-virginica
30	7.2	3.6	6.1	2.5	Iris-virginica

Outline

- **For Loop**
- **While Loop**
- **String**
- **File**
- **Case Studies**

Recursive

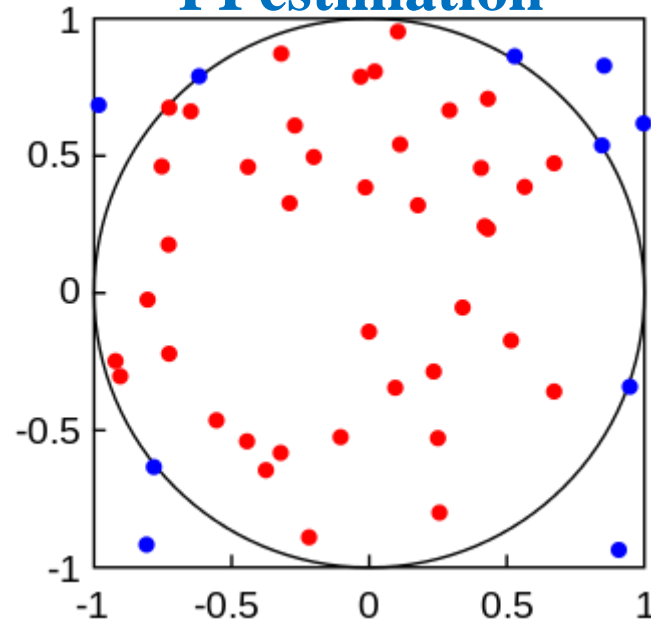
E estimation

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

Simulation of coin tossing



PI estimation



Compute quadratic root for the number N

Newton Method

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

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

$n = n + 1$

Example

❖ Simulation of coin tossing

Event: bất kì sự kiện gì xảy ra

Experiment: sự thử nghiệm cho ra kết quả

Sample space: tập hợp tất cả các kết quả có thể xảy ra của một experiment

Random variable: biến ngẫu nhiên, mô tả outcome của một event



Tiền xu có 2 mặt
(head và tail)

Experiment: ném đồng xu

Sample space $S = \{\text{Head}, \text{Tail}\}$

Event: Gọi A là event mà kết quả của một lần ném là head

$P(A)$ Xác suất để event A xảy ra

Xác suất để kết quả của một lần ném là head

Xác suất

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

$A = \{\text{head}\}$

$$S = \{\text{head}, \text{tail}\} \quad P(A) = \frac{|A|}{|S|} = \frac{1}{2}$$

Example

❖ Simulation of coin tossing



Count #heads

Count #tails

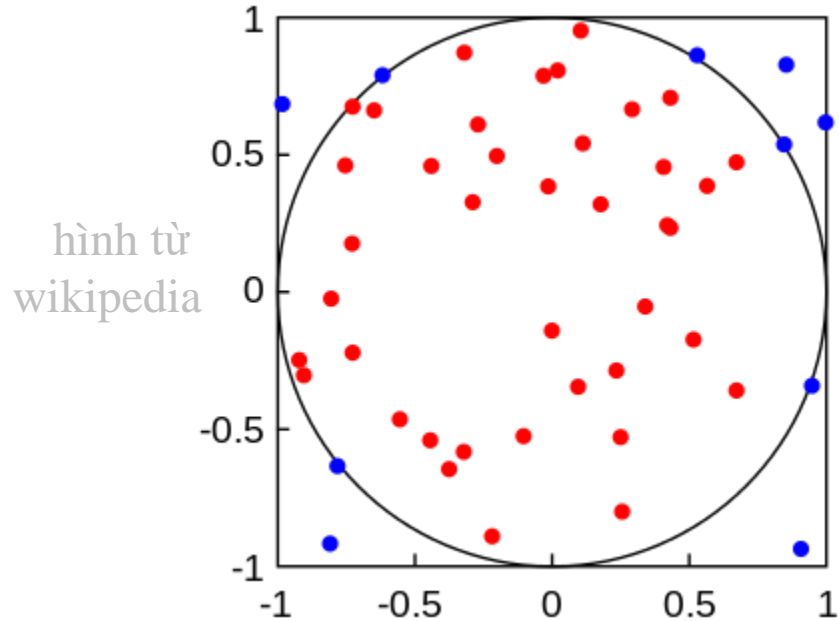
Check if the two
numbers are similar

Demo

```
1. # aivietnam.ai
2. import random
3.
4. # Tổng số lần búng đồng xu
5. total_flips = 0
6.
7. # số lần mặt sau xuất hiện
8. num_tails = 0
9.
10. # số lần mặt trước xuất hiện
11. num_heads = 0
12.
13. for _ in range(1000):
14.     # sinh số ngẫu nhiên nằm trong khoảng [0,1)
15.     n = random.random()
16.     if n < 0.5:
17.         num_tails = num_tails + 1
18.     else:
19.         num_heads = num_heads + 1
20.
21.     # code ở vị trí này không thuộc khối else
22.     total_flips = total_flips + 1
```

Example: PI Estimation

❖ PI estimation



N_s is #random samples within the square
generated according to uniform distribution

N_c is #random samples within the circle
generated according to uniform distribution

circle radius $r = 1$
circle_area $A_c = \pi r^2$

square side $s = 2$
square_area $A_s = s^2$

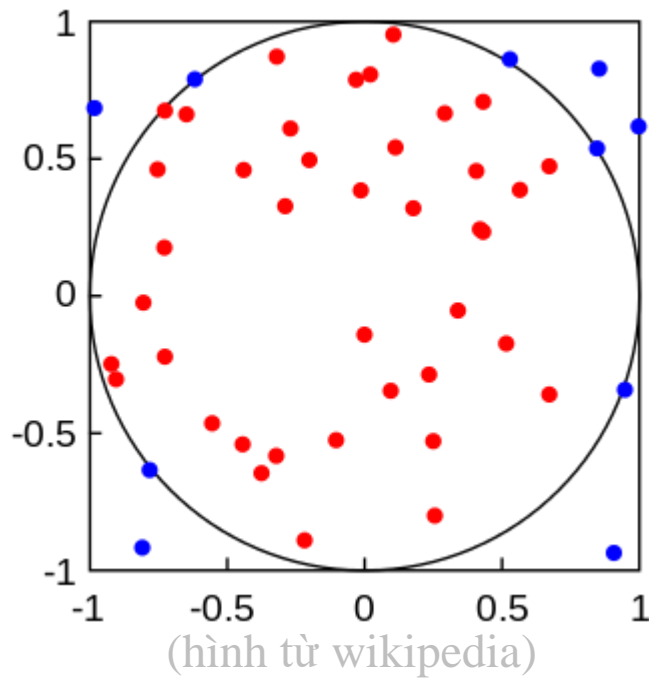
$$\frac{A_s}{A_c} \approx \frac{N_s}{N_c}$$

$$\frac{s^2}{\pi r^2} \approx \frac{N_s}{N_c}$$

$$\pi \approx \frac{s^2 N_c}{N_s}$$

Example

❖ PI estimation



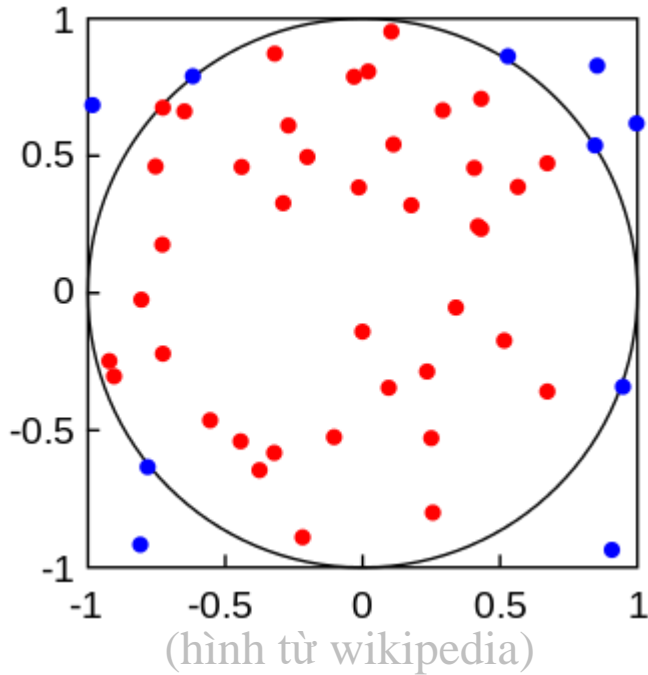
$$\pi \approx \frac{s^2 N_c}{N_s}$$

Random numbers $\in [-1,1]$

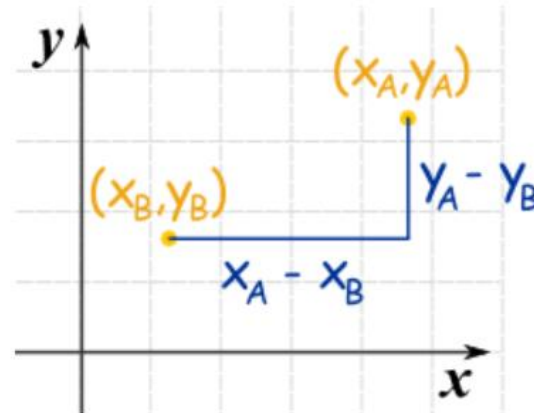
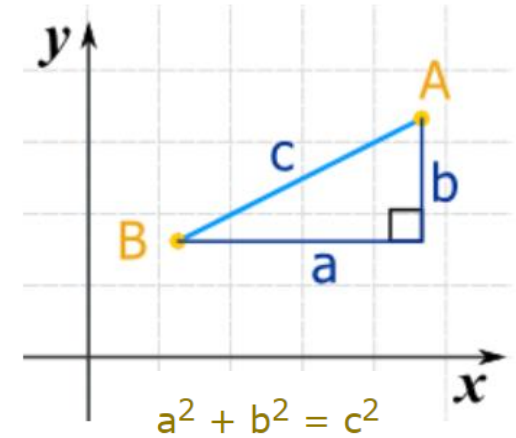
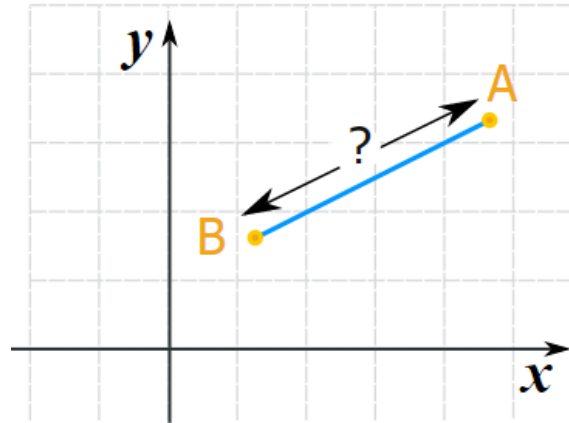
Check if a point is
within the circle

Example

❖ PI estimation



$$\pi \approx \frac{s^2 N_c}{N_s}$$

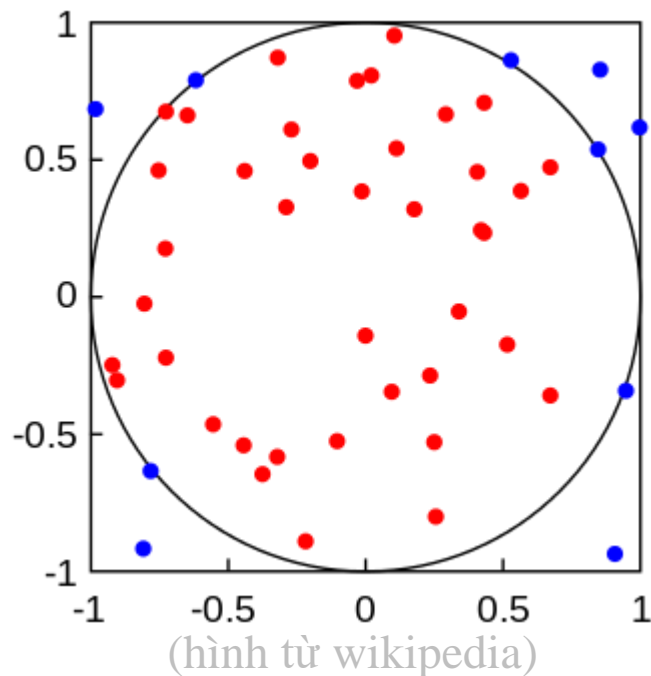


$$c = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$$

<https://www.mathsisfun.com/algebra/distance-2-points.html>

Example

❖ PI estimation

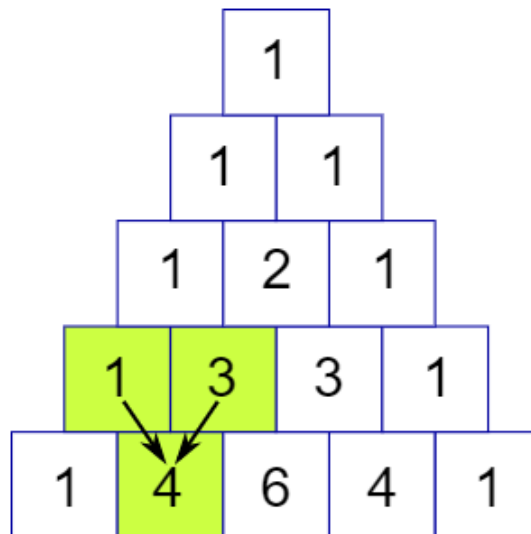


$$\pi \approx \frac{s^2 N_c}{N_s}$$

```
1. # aivietnam.ai
2. import random
3. import math
4.
5.
6. # Tổng số điểm p được sinh ra
7. N = 100000
8.
9. # số điểm thuộc tình tròn
10. N_T = 0
11.
12. # Sinh ra N điểm ngẫu nhiên
13. for i in range(N):
14.     # sinh ra x, y thuộc [-1, 1].
15.     x = random.random()*2 - 1
16.     y = random.random()*2 - 1
17.
18.     x2 = x**2
19.     y2 = y**2
20.
21.     # kiểm tra p có nằm trong đường tròn
22.     if math.sqrt(x2 + y2) <= 1.0:
23.         N_T = N_T + 1
24.
25. # tính PI
26. pi = (N_T / N) * 4
27. print(pi)
```

Exercises

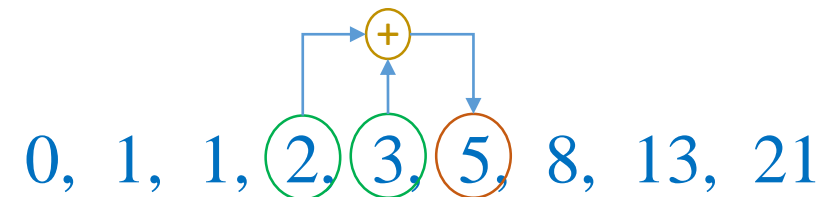
Pascal's Triangle



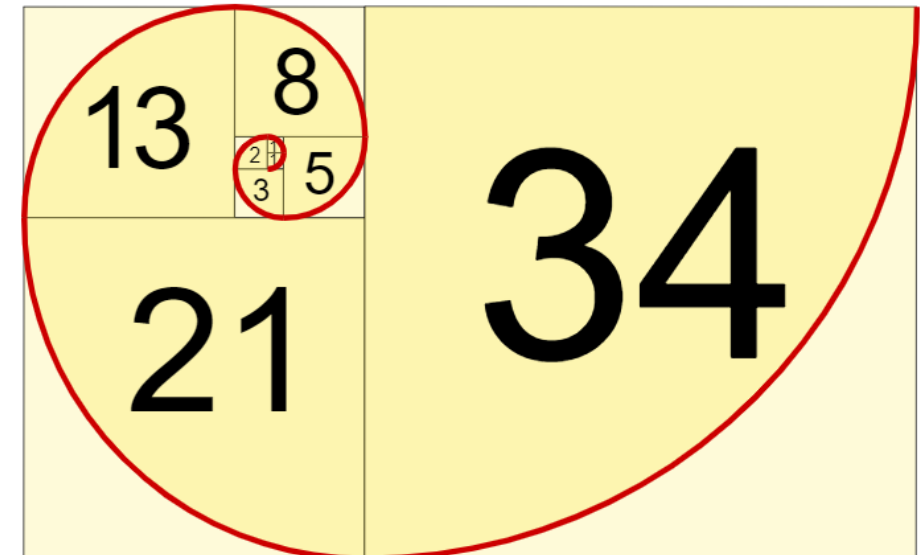
Level = 5

<https://www.mathsisfun.com>

Fibonacci Sequence



Length = 9



Exercises

$$\sin(x) \approx \sum_{n=0}^{\infty} (-1)^n \frac{x^{(2n+1)}}{(2n+1)!} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \dots$$

$$\cos(x) \approx \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n}}{(2n)!} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \frac{x^{10}}{10!} + \dots$$

$$e^x \approx \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + \frac{x^1}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \dots$$

$$\sinh(x) \approx \sum_{n=0}^{\infty} \frac{x^{(2n+1)}}{(2n+1)!} = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} + \frac{x^9}{9!} + \dots$$

$$\cosh(x) \approx \sum_{n=0}^{\infty} \frac{x^{2n}}{(2n)!} = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \frac{x^6}{6!} + \frac{x^8}{8!} + \frac{x^{10}}{10!} + \dots$$

Cheat Sheet 1

Run a Python program

- _ Python files have “.py” at the end of the filename
- _ To run a Python file:
`python file.py`

Virtual Environment

- _ Install Python Virtual Environment (in Linux):
`sudo apt-get install -y python3-env`
- _ Create a virtual environment:
`python3 -m venv a_name`
- _ Activate a virtual environment:
`source a_name/bin/activate`

Install new package

- _ e.g: to install *matplotlib*:
`pip install matplotlib`

Variable

- _ Declare a variable:
`variable_name = variable_value`
- _ Variable types:

Integer	1, 2, 0, -1, -2
Float	1.5, 0.5, -3.21
String	‘AI’, “VIETNAM”
Boolean	True, False

Basic Operators

Operators	Meaning
+	Summation
-	Subtraction
*	Multiplication
/	Division
%	Modulo
//	Floor Division
**	Power

Function

_ Define a function:

```
def function_name(parameters):  
    '''  
    docstring  
    '''  
    # your code goes here  
    return result
```

_ Default values:

```
def function_name(p1=0, p2=0):  
    # your code goes here
```

Condition

- _ Comparison Operators:

==	Equal
!=	Not equal
>	Greater than
<	Less than
>=	Greater or equal than
<=	Less or equal than

- _ Conditional sentence:

```
if condition1:  
    # your code  
elif condition2:  
    # your code  
else:  
    # your code
```

Built-in Functions

<code>print(params)</code>	Print value onto the screen
<code>type(params)</code>	Return class type of variable
<code>input(prompt)</code>	Ask user to input a string
<code>int()</code> , <code>float()</code>	Type conversion

Overflow/Underflow

- _ Underflow:
`result = 1e-100`
`print(result) # 1e-100`
`result = 1e-1000`
`print(result) # 0.0`
- _ Overflow:
`result = 1e100`
`print(result) # 1e+100`
`result = 1e1000`
`print(result) # inf`

For Loop

- _ Create a loop using for:
`for element in iterable:`
 # code inside your for
- _ Some iterables:

String	“aivietnam”
Tuple	(1, 2, 3)
List	[1, 2, 3]
Dictionary	{‘key1’: 1}
range()	range(0, 5, 1)

- _ Special keywords:

<code>break</code>	Exit the loop
<code>continue</code>	Move to next iteration

Cheat Sheet 2

Random & Math module

_ Math module's common methods and constants:

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

_ 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

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

PI estimation

_ Monte Carlo Method:

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

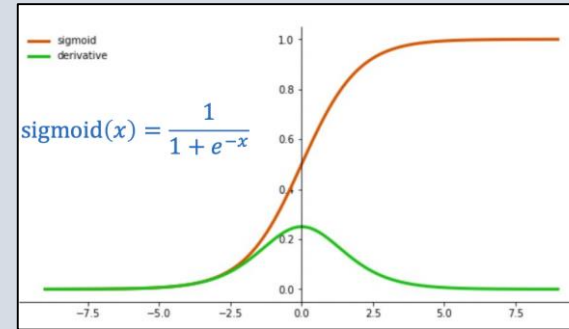
_ Gregory-Leibniz Series:

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

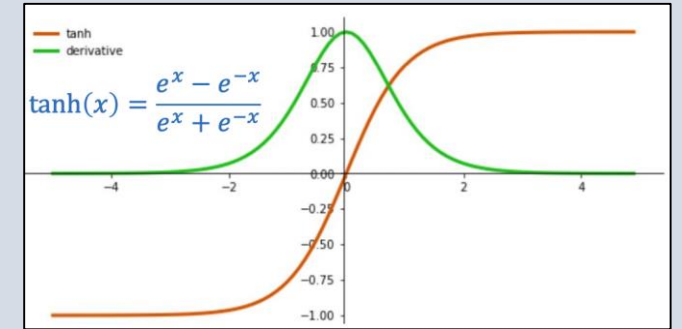
_ Nilakantha Series:

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

Activation Functions



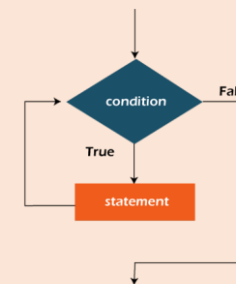
Map x values into smaller ranges



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])
```

Cheat Sheet – For Loop

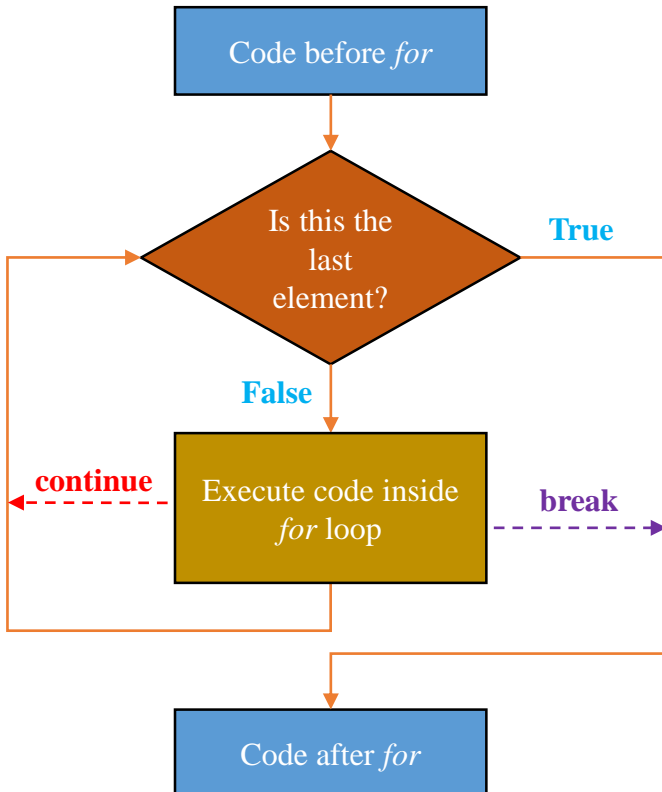
for syntax

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)
```

Tuple:

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

range(start, end, step):

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

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

List:

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

Dictionary:

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

```
# 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

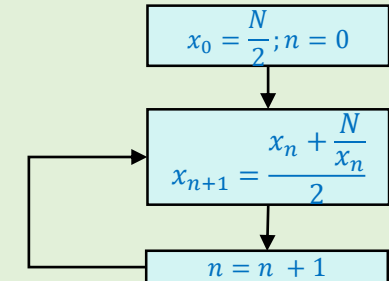
Coin tossing

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

Euler's number

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

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:

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

