สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น)
Technology Promotion Association (Thailand-Japan)

สถาบันเทคโนโลยีไทย-ญี่ปุ่น
Thai-Nichi Institute of Technology
泰日工業大学

พื้นฐานการเขียนโปรแกรม python และการประยุกต์ใช้งานทางด้าน AI (การเรียนรู้ของเครื่องและการทำนายผลลัพธ์)

19-20 สิงหาคม 2566

แนะนำทีมงาน

- รศ. ดร.วรากร ศรีเชวงทรัพย์
- ผศ. ดร.กันติชา กิตติพีรชล
- อ.ชาตรี ทองวรรณ

หัวข้อการอบรม

- แนะนำภาษา Python
- หลักภาษาและไวยกรณ์ทั่วไปของภาษา Python
- การเขียนคำสั่งเกี่ยวกับเงื่อนไขทางเลือก
- การเขียนคำสั่งวนลูป
- การใช้งาน String และ List
- การสร้างและใช้งานฟังก์ชัน (Function)
- การประยุกต์ใช้ในงานทางด้าน AI (การเรียนรู้ของเครื่องและทำนายผลลัพธ์)

สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น)
Technology Promotion Association (Thailand-Japan)

สถาบันเทคโนโลยีไทย-ญี่ปุ่น
Thai-Nichi Institute of Technology
泰日工業大学

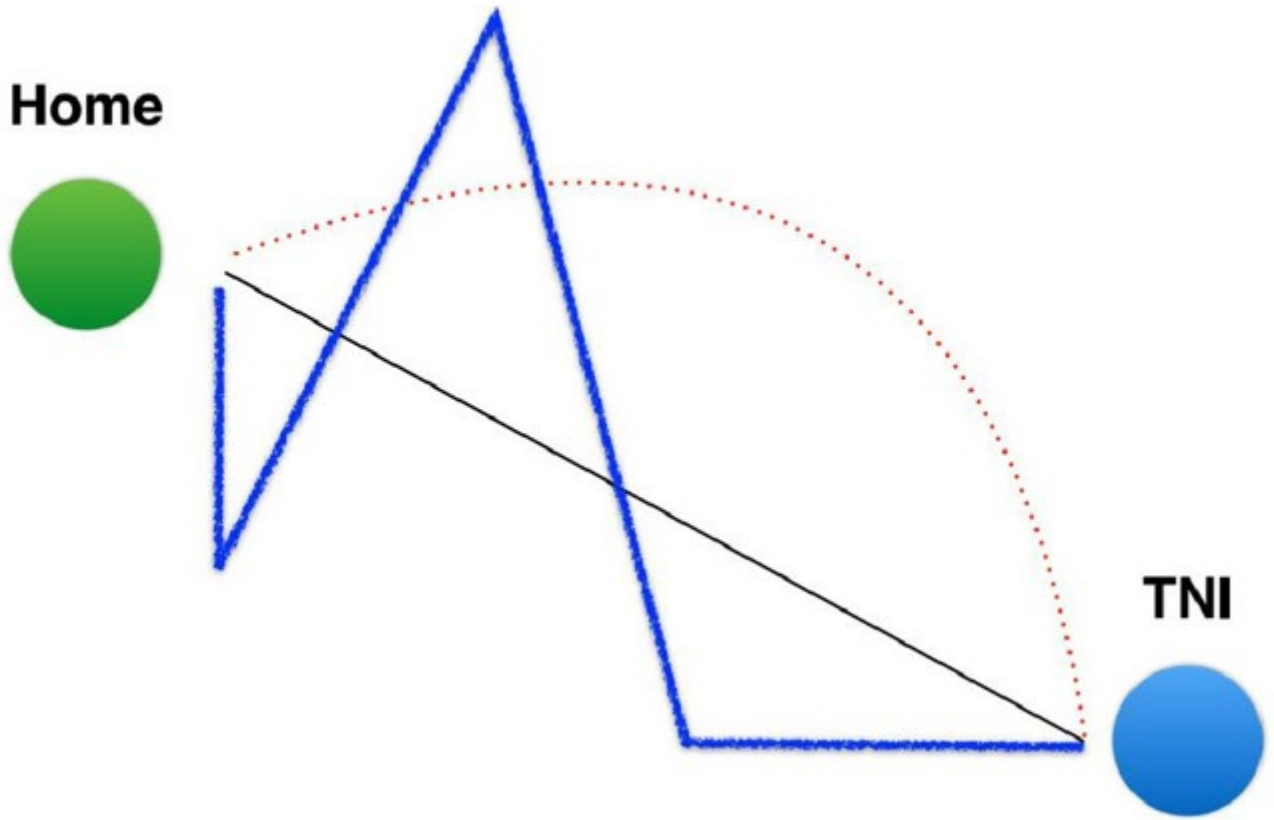
แนะนำภาษา PYTHON

หลักภาษาและไวยกรณ์ทั่วไป

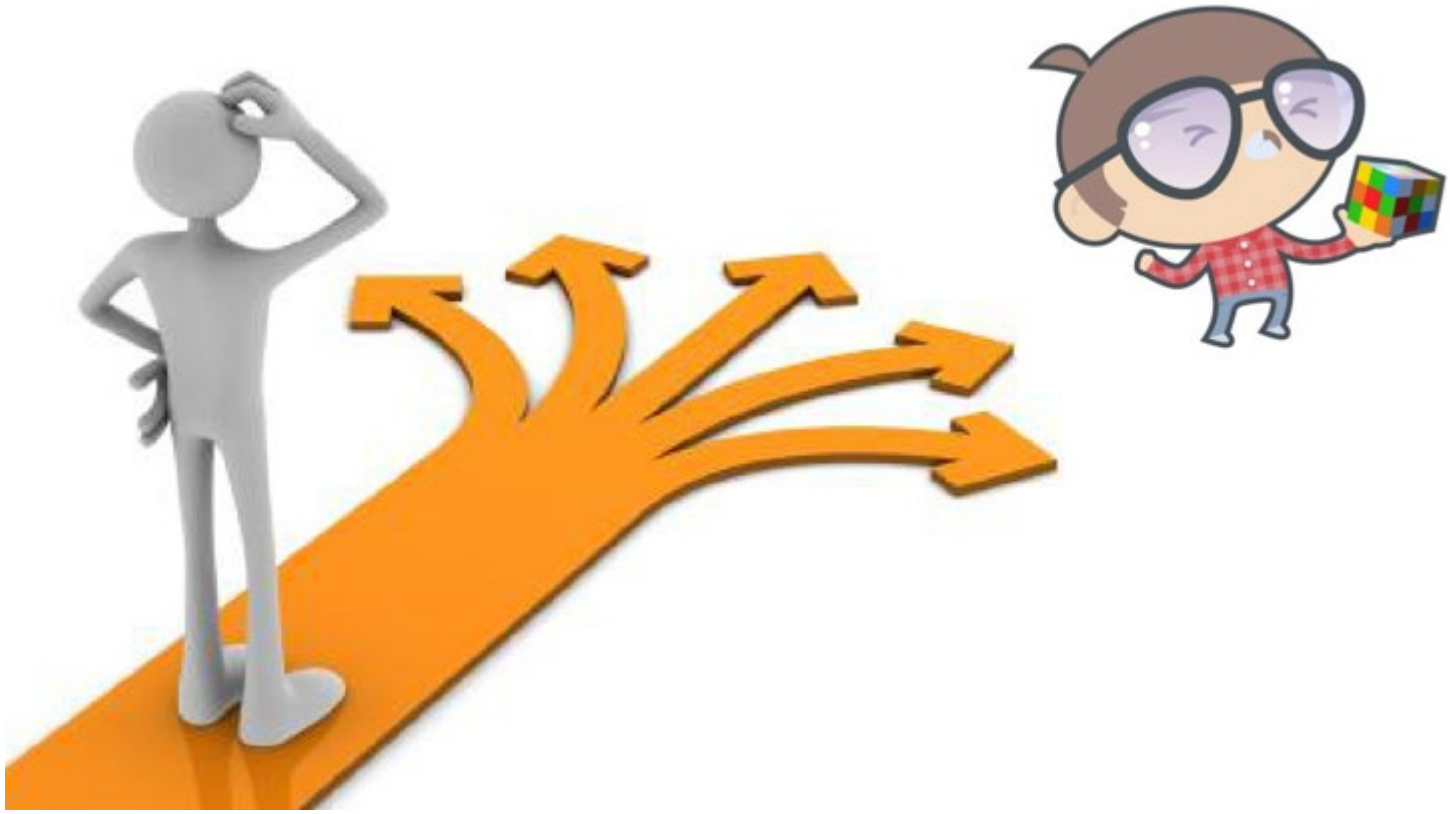
Program

“A computer program is a set of instructions...”

Program



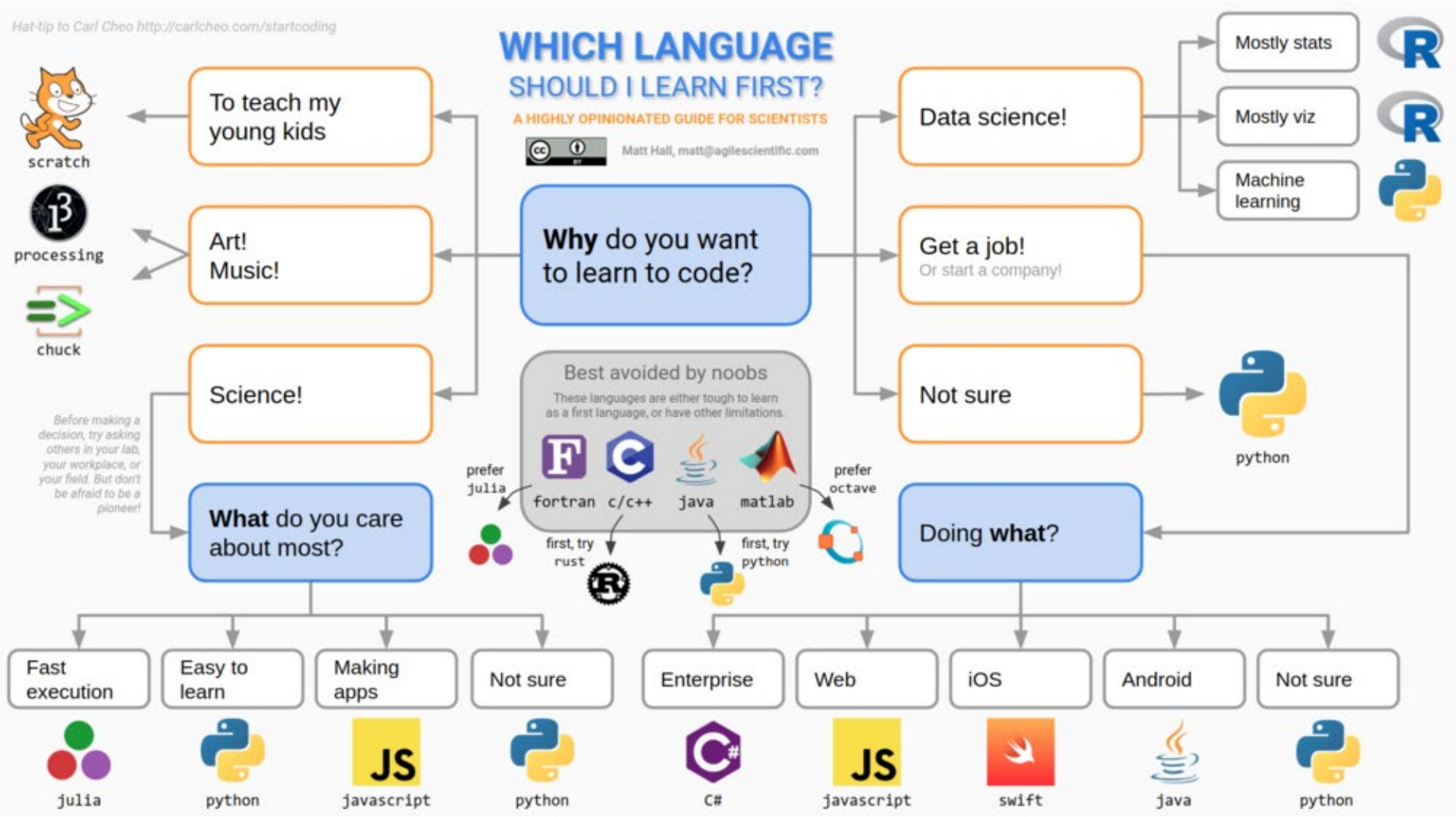
Program



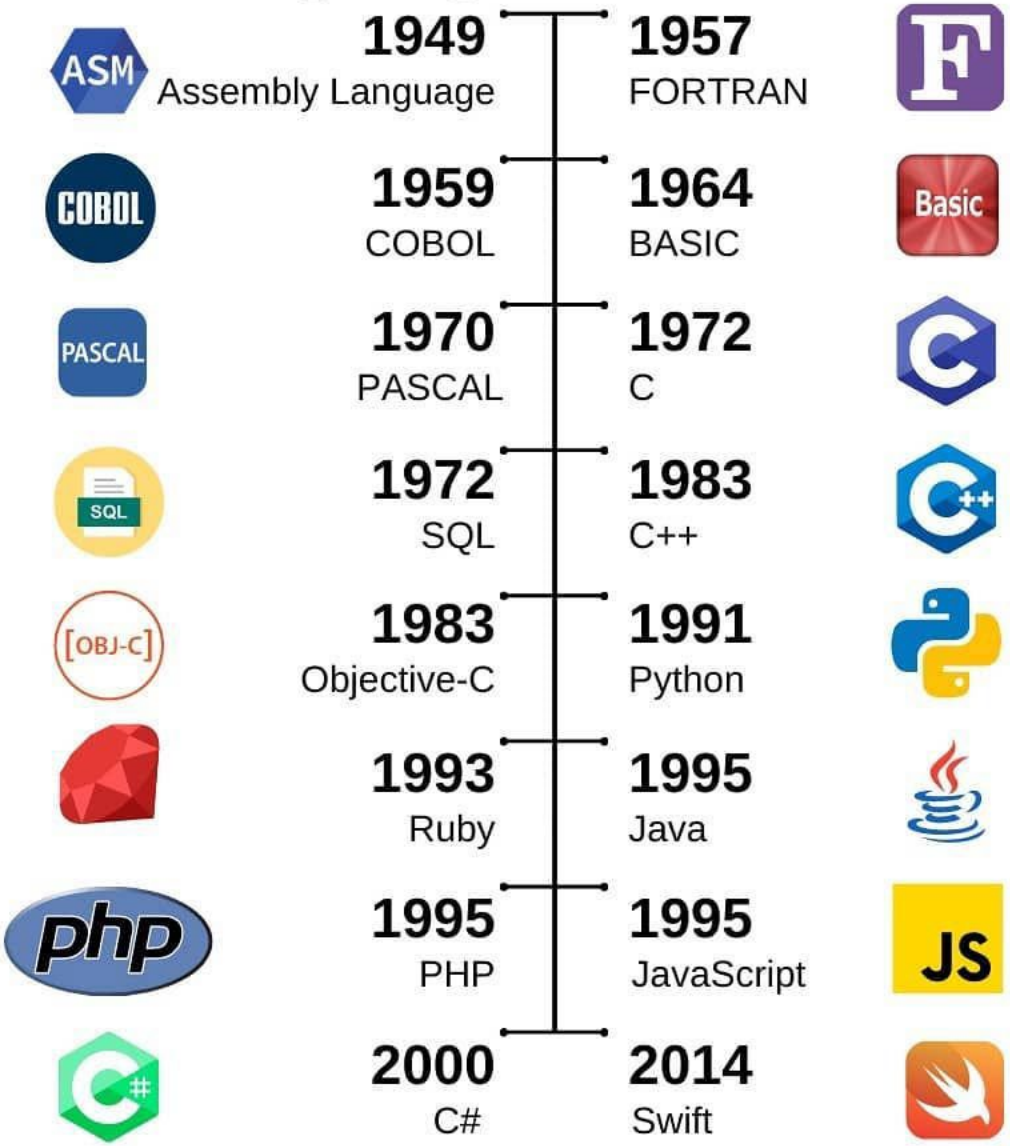
Program


How to Pick
Your First Programming Language
Based on the Life You Want
PHP
C#
R
C
JAVA SCRIPT
C++
JAVA
PYTHON
MATLAB
RUBY

Program

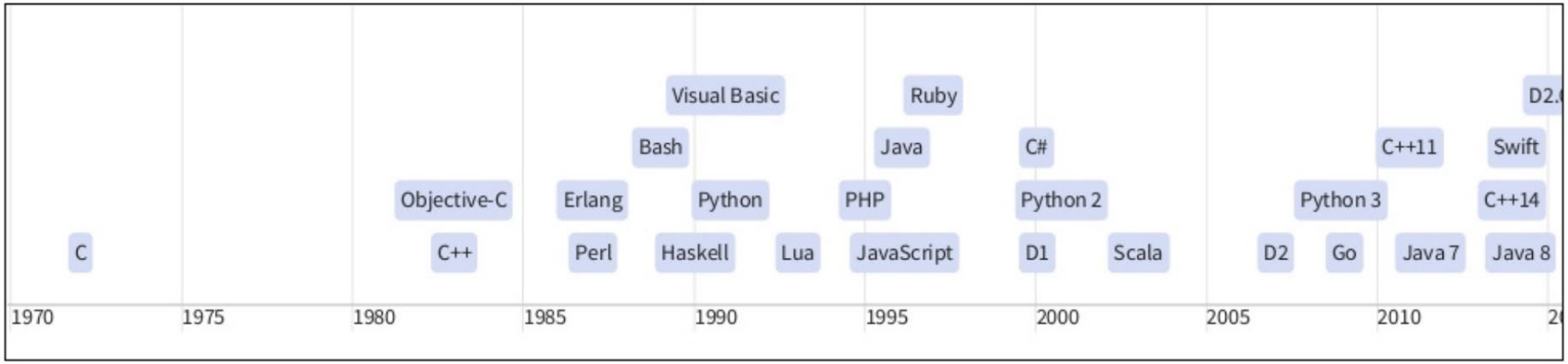

Hat-tip to Carl Cheo http://carlcheo.com/startcoding
WHICH LANGUAGE
SHOULD I LEARN FIRST?
A HIGHLY OPINIONATED GUIDE FOR SCIENTISTS
Matt Hall, matt@agilescientific.com
Why do you want to learn to code?
To teach my young kids
scratch
Art! Music!
processing
chuck
Science!
What do you care about most?
Before making a decision, try asking others in your lab, your workplace, or your field. But don't be afraid to be a pioneer!
Data science!
Mostly stats
Mostly viz
Machine learning
Get a job!
Or start a company!
Not sure
python
Doing what?
Best avoided by noobs
These languages are either tough to learn as a first language, or have other limitations.
fortran
c/c++
java
matlab
prefer julia
prefer octave
first, try rust
first, try python
Fast execution
julia
Easy to learn
python
Making apps
javascript
Not sure
python
Enterprise
C#
Web
javascript
iOS
swift
Android
java
Not sure
python

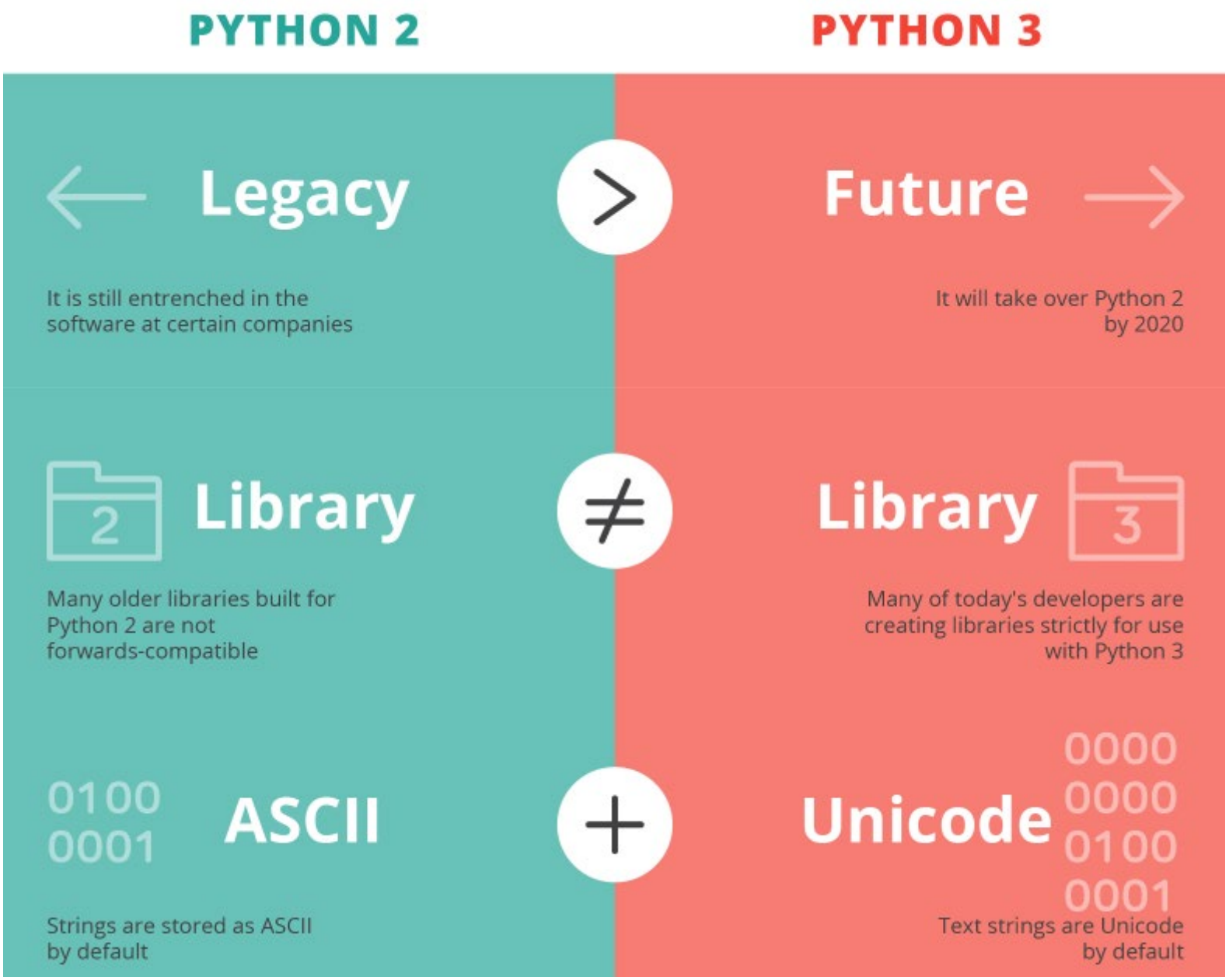
History of Popular Programming Languages: A timeline

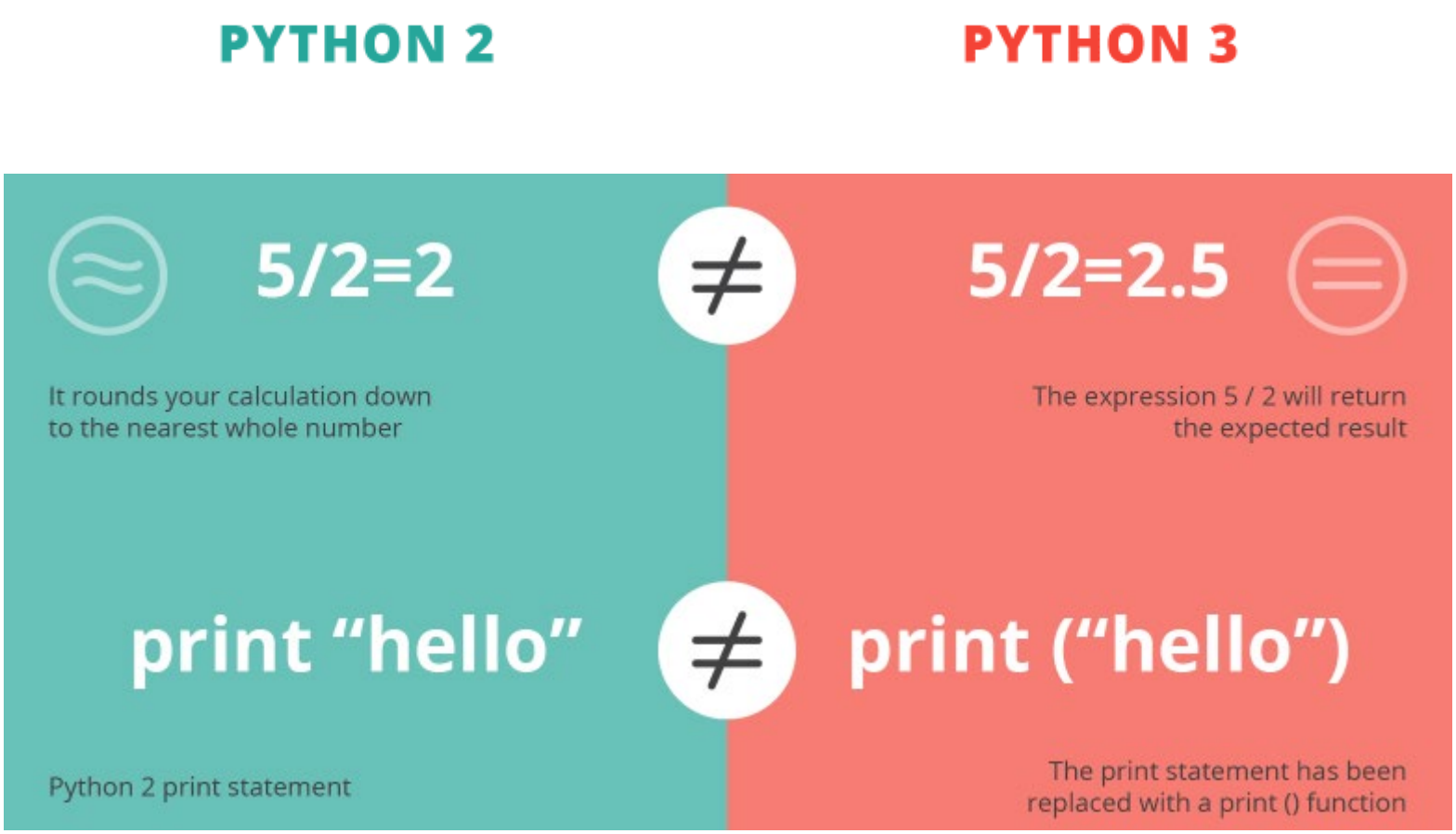


python.learning

Program

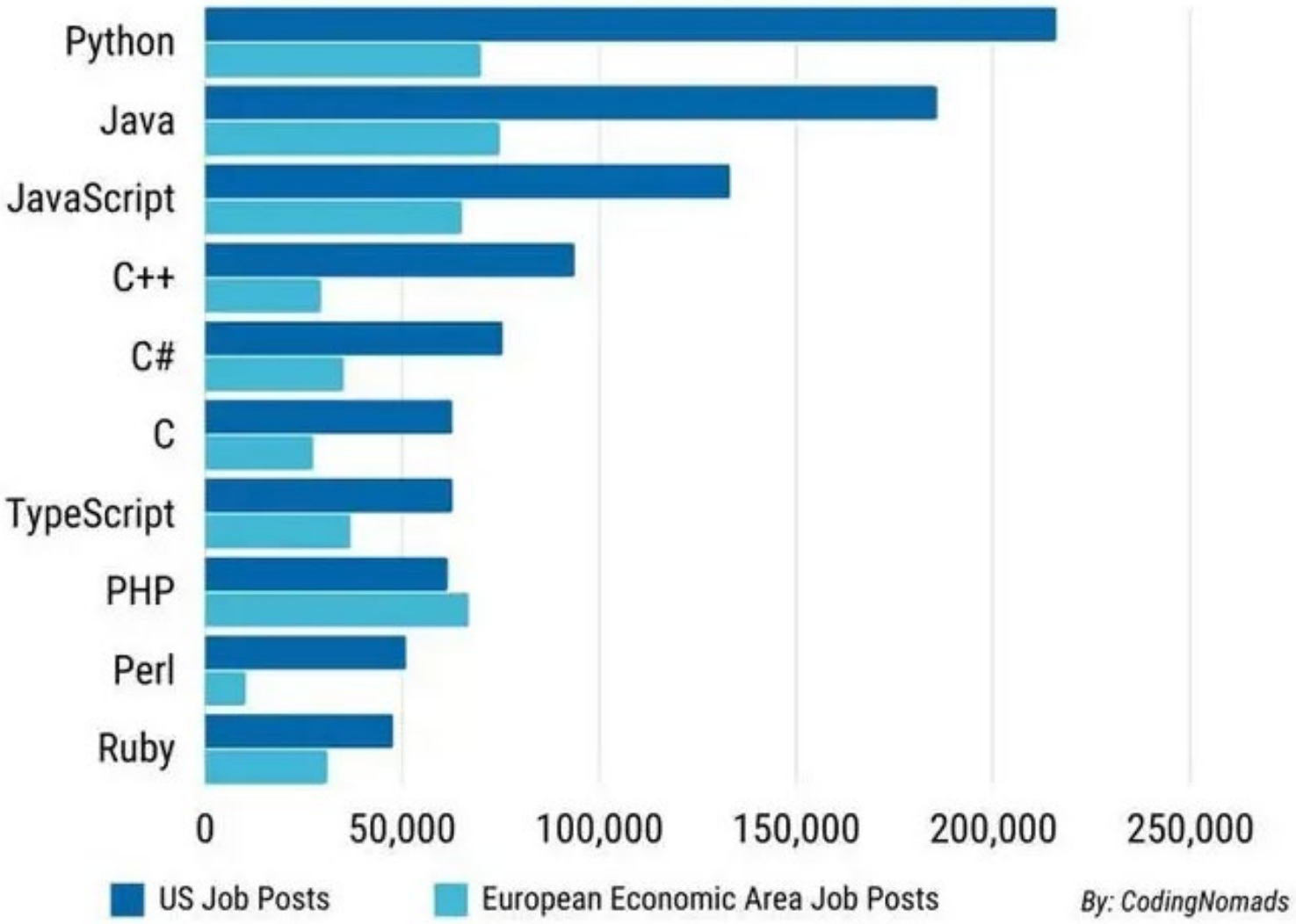

Visual Basic
Ruby
Bash
Java
C#
C++11
Swift
Objective-C
Erlang
Python
PHP
Python 2
Python 3
C++14
C
C++
Perl
Haskell
Lua
JavaScript
D1
Scala
D2
Go
Java 7
Java 8
1970
1975
1980
1985
1990
1995
2000
2005
2010


PYTHON 2
PYTHON 3
Legacy
>
Future
It is still entrenched in the software at certain companies
It will take over Python 2 by 2020
2
Library
≠
Library
3
Many older libraries built for Python 2 are not forwards-compatible
Many of today's developers are creating libraries strictly for use with Python 3
0100 0001
ASCII
+
Unicode
0000 0000 0100 0001
Strings are stored as ASCII by default
Text strings are Unicode by default

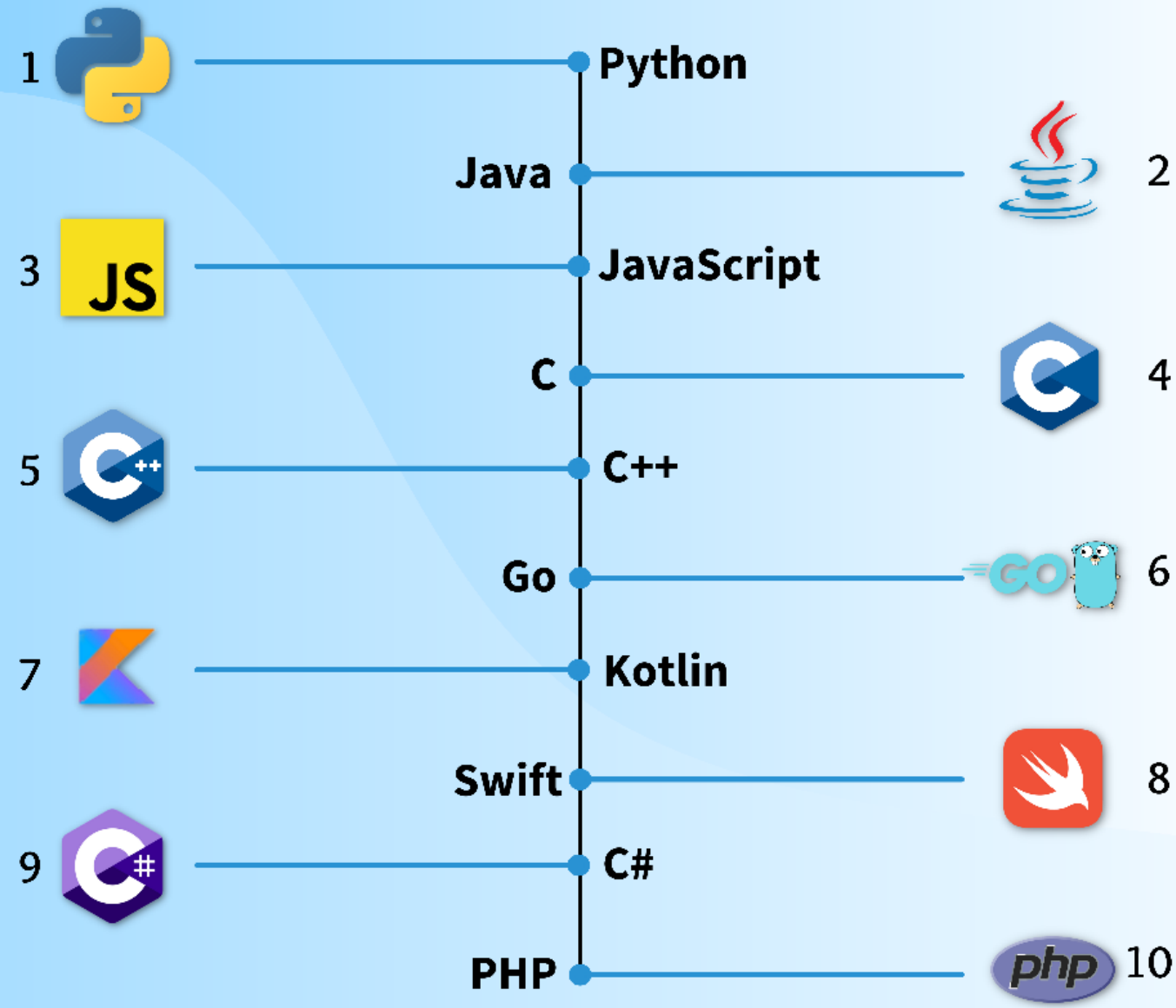

PYTHON 2
PYTHON 3
5/2=2
≠
5/2=2.5
It rounds your calculation down to the nearest whole number
The expression 5 / 2 will return the expected result
print “hello”
≠
print (“hello”)
Python 2 print statement
The print statement has been replaced with a print () function

Most in-demand programming languages of 2022

Based on LinkedIn job postings in the USA & Europe



10 MOST POPULAR CODING LANGUAGES OF 2021



{ik} INTERVIEW KICKSTART

Program

1. Python

> Released in: **1991**
> Career opportunities: **Very high**
> Average base salary: **$108,602** per year (as of June 2021)
> Current number of jobs: **40,026** (as of June 2021)

> Source: *Indeed.com*

As one of the most popular coding languages globally, Python finds invaluable use in solving problems and gaining insights. The language has recorded a 27% year-over-year user growth, mainly fueled by data analysts and data scientists.

Its application in AI analysis, deep learning, and data analysis is indispensable. Python is definitely among the top 3 popular programming languages of 2021.

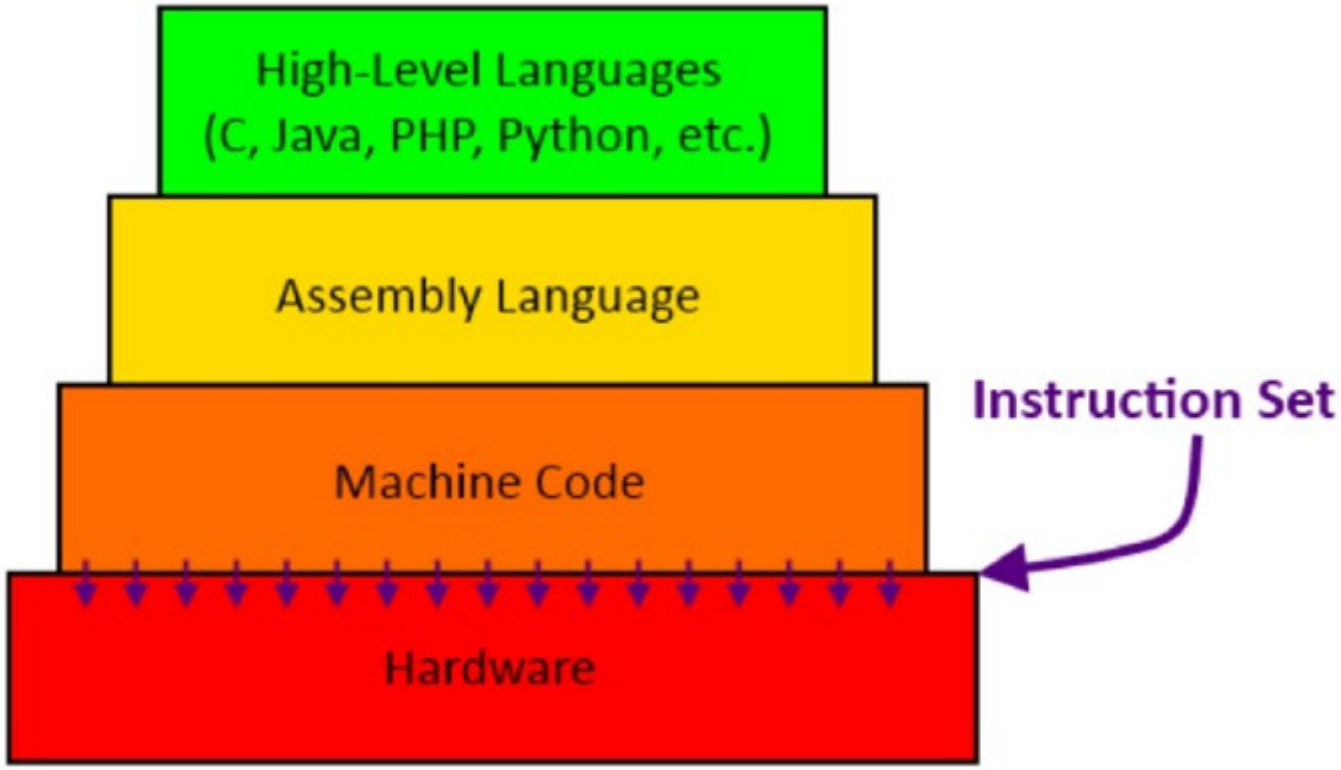
Why Should You Learn Python?

- Python is a general-purpose, high-level, interpreted language known for its ease of use, extensive libraries and tools, and code readability.
- Python can handle complex algorithms and is excellent for data automation, making it a natural fit for machine learning.

Why Should You Learn Python?

- Being open-source, it has support from a large community of Python users committed to making it better and more efficient.
- Python has several mobile and web development frameworks like Django, TurboGears, Bottle, Pyramid, and Flask, as well as scientific application frameworks like TensorFlow, Keras, and SciPy.
- Software engineers use Python to develop GUI-based applications, handle AI computations, and information science.

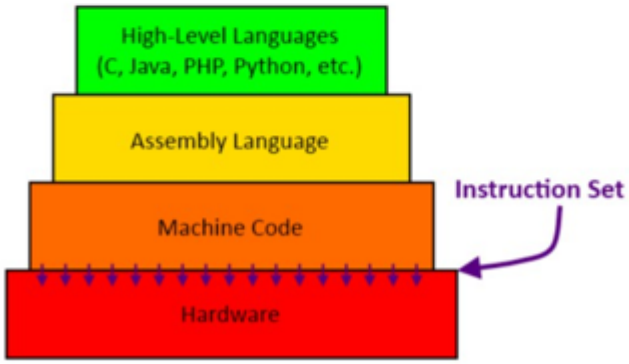
Python


High-Level Languages
(C, Java, PHP, Python, etc.)
Assembly Language
Machine Code
Instruction Set
Hardware

Python

Python

print ("HELLO WORLD")



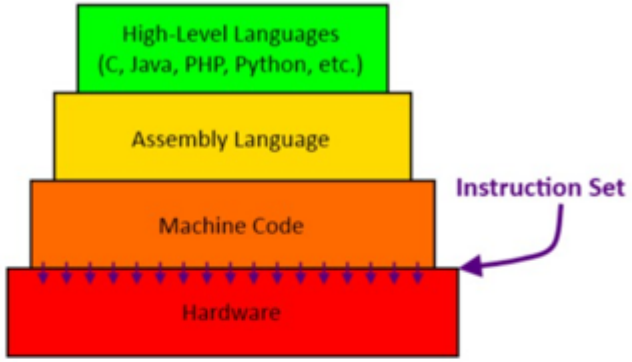
Assembly

```
        .or $300
main    ldy #$00
.1      lda str,y
        beq .2
        jsr $fded
        iny
        bne .1
.2      rts
str     .as "HELLO WORLD"
        .hs 0D00
```

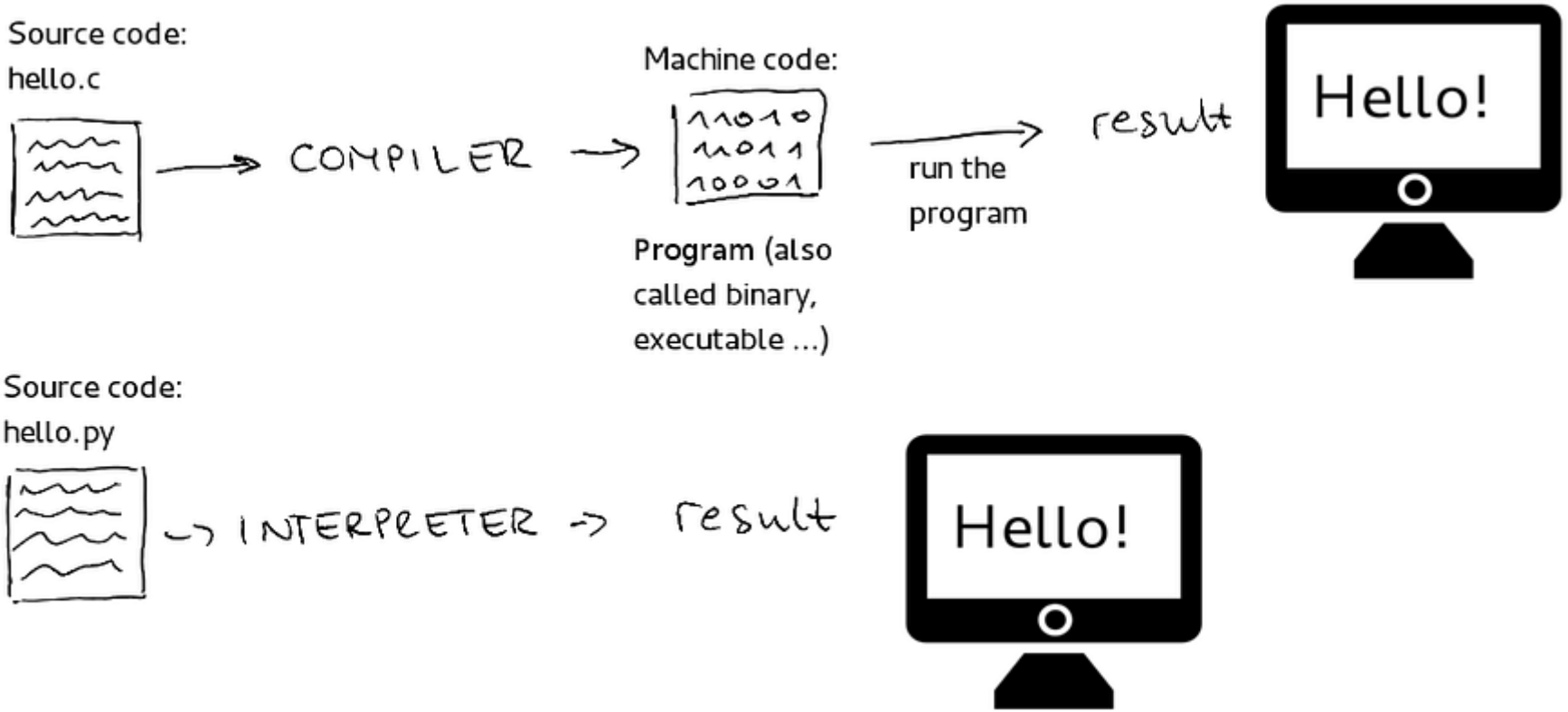
Python

Machine Code

c7 3c 2a 3c 2a 2b 2a 5c 3c 28 5c 2a 2b 2a 5c 3c
28 5c 2a 2b 2a 5c 3c 28 5c 2a 2b 2a 5c 3c 28 5c
2a 2b 2a 5c 3c 28 5c 2a 2b 2a 5c 3c 28 5c 2a 2b
2a 5c 3c 28 5c 2a 2b 2a 5c 3c 28 5c 2a 2b 2a 5c
3c 28 5c 2a 2b 2a 5c 3c 28 5c 2a 2b 2a 5c 3c 28
5c 2a 2b 2a 5c 3c 28 5c 2a 2b 2a 5c 3c 28 5c 2a
2b 2a 00 00 01 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 64 48 65 6c 6c 6f 2c 20 57
6f 72 6c 64 21 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00



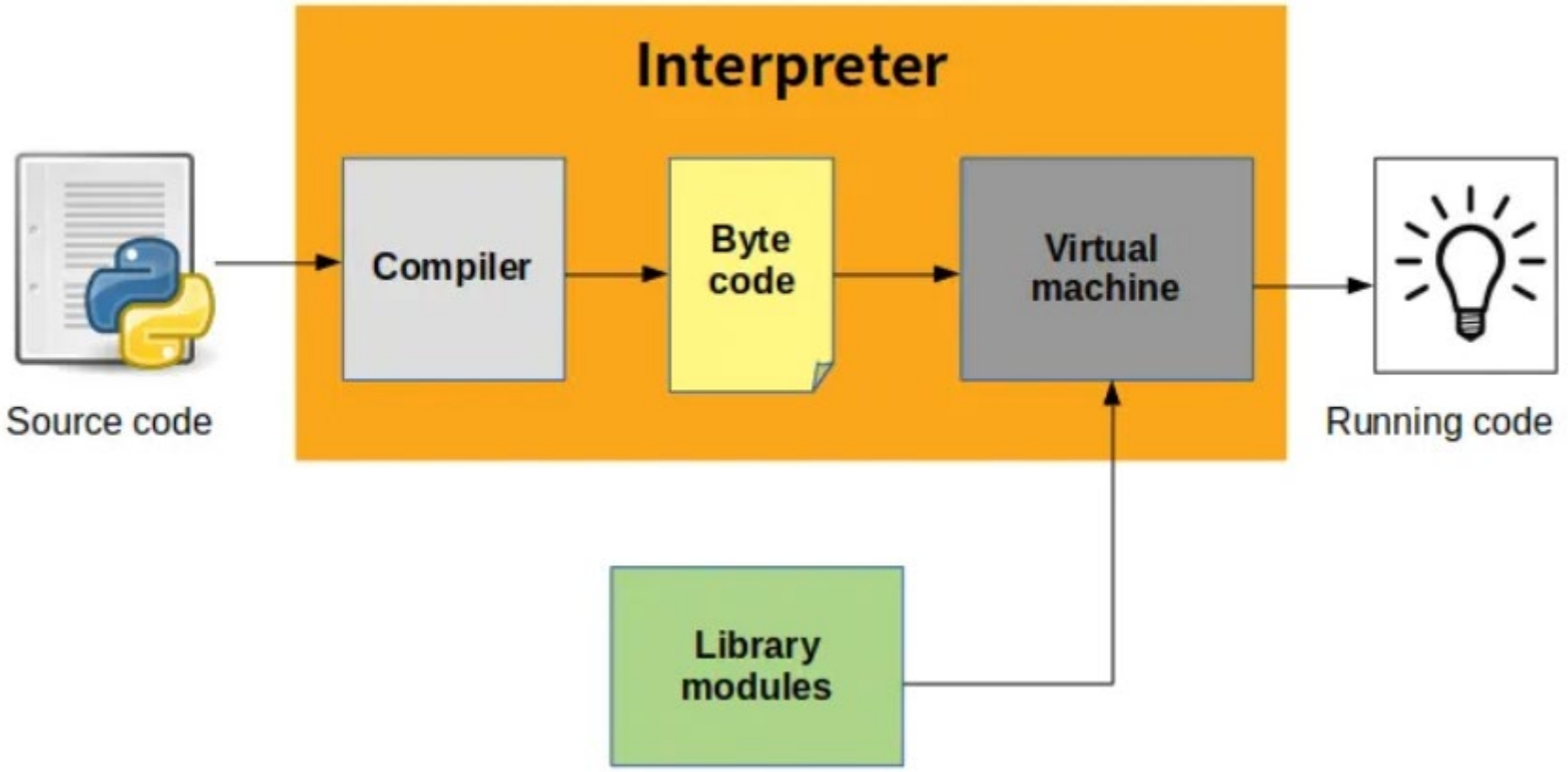
Compiled vs. Interpreted



Compiled vs. Interpreted

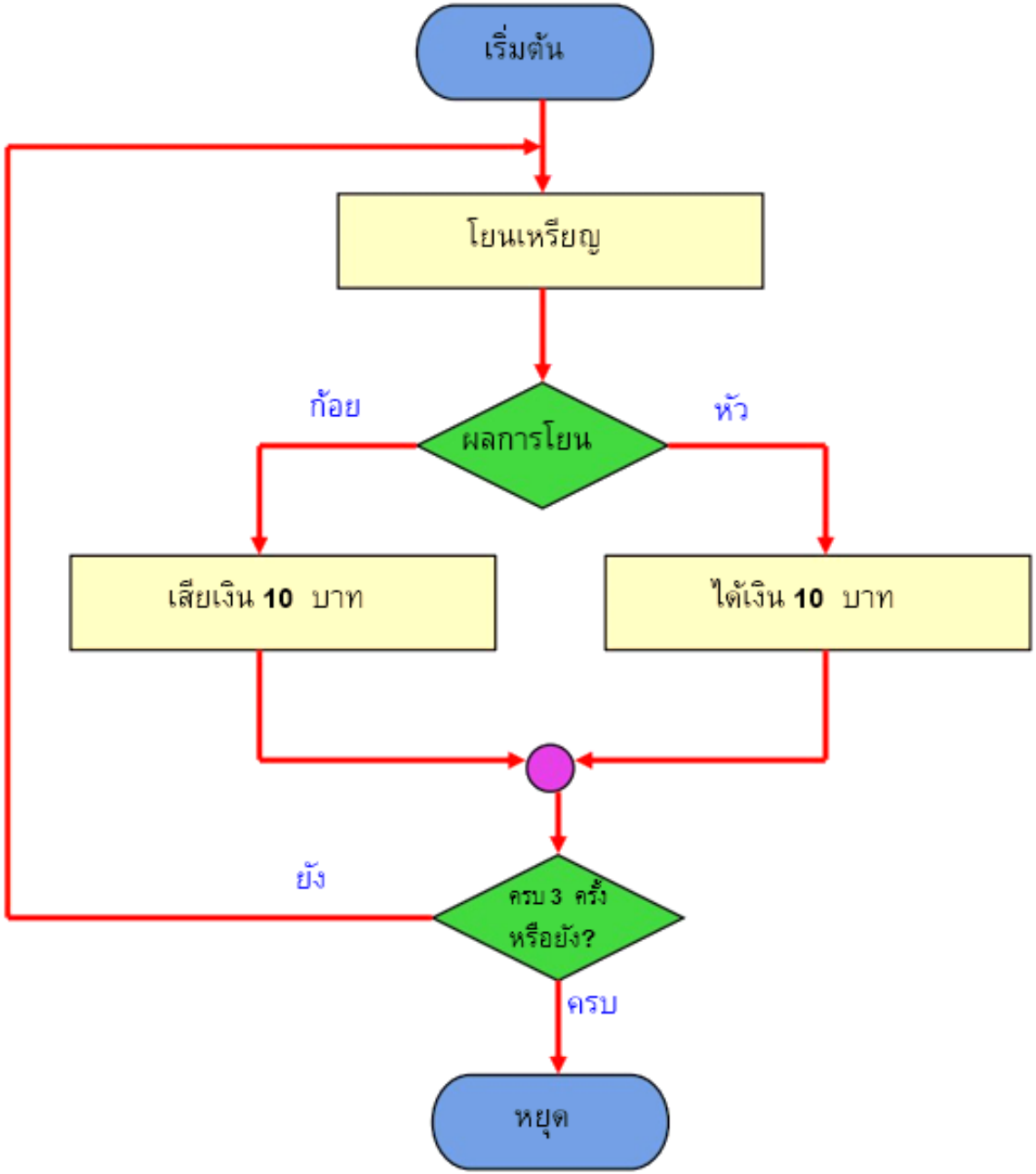
Compiled		Interpreted	
PROS	CONS	PROS	CONS
ready to run	**not** cross platform	cross-platform	interpreter required
often **faster**	inflexible	simpler to test	often **slower**
source code is **private**	extra step	easier to debug	source code is **public**

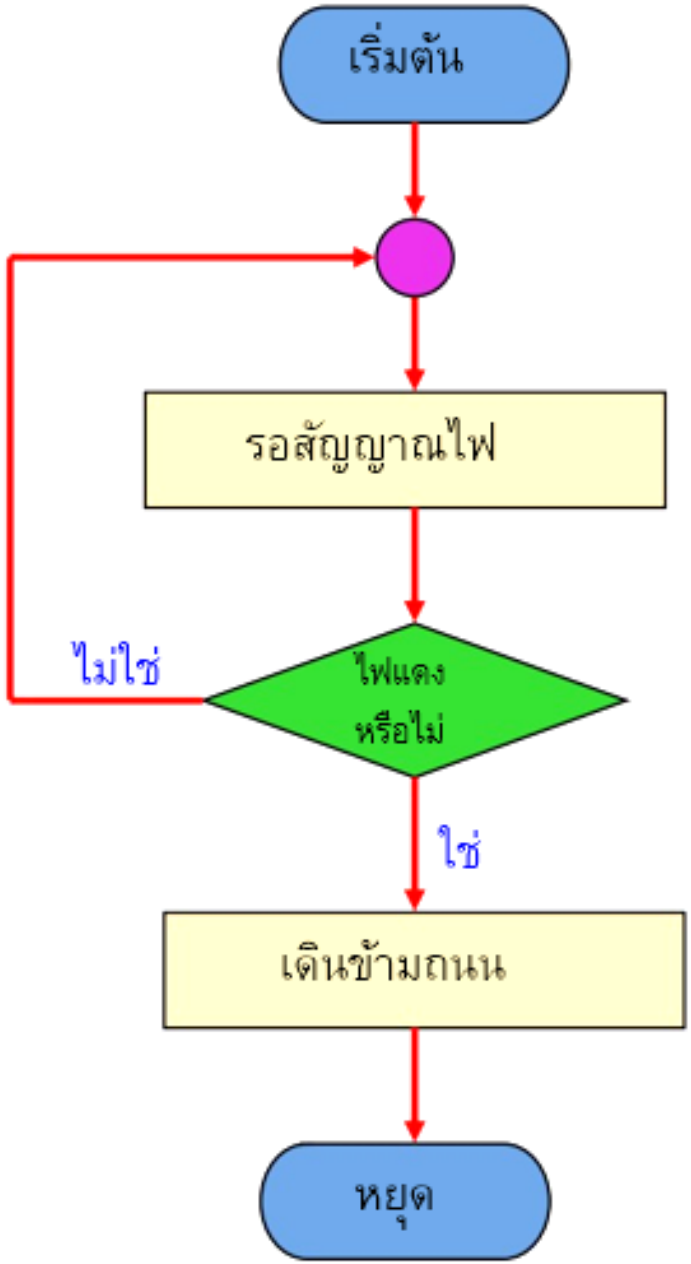
Program


Interpreter
Compiler
Byte code
Virtual machine
Source code
Running code
Library modules

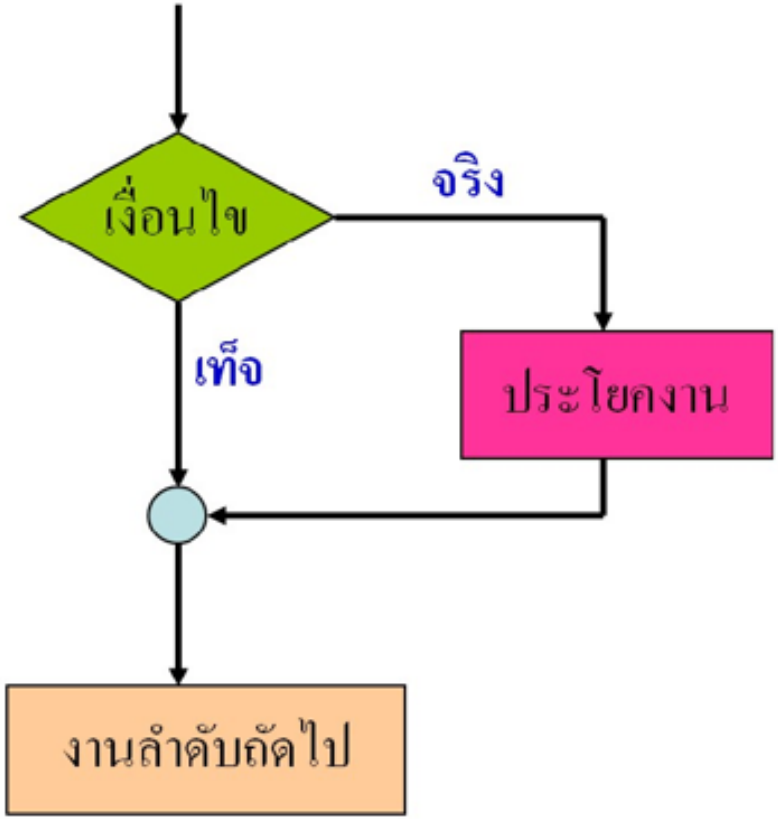
Flowchart

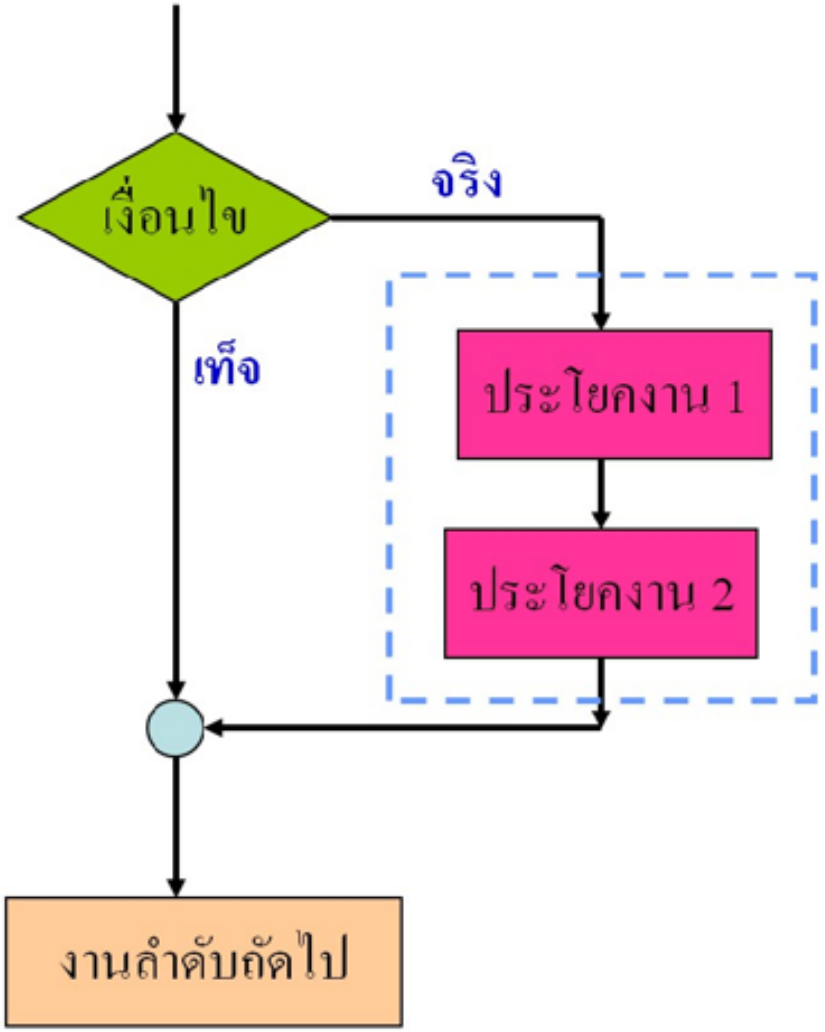
- เครื่องมือที่ช่วยในการเขียนโปรแกรม
- เครื่องหมายแสดงลำดับขั้นตอน
- รูปภาพหรือสัญลักษณ์
- เข้าใจลำดับขั้นตอนการเขียนโปรแกรม
- ง่ายต่อการตรวจสอบความถูกต้อง


เริ่มต้น
โยนเหรียญ
ผลการโยน
ก้อย
หัว
เสียเงิน 10 บาท
ได้เงิน 10 บาท
ครบ 3 ครั้ง
หรือยัง?
ยัง
ครบ
หยุด


เริ่มต้น
รอสัญญาณไฟ
ไฟแดง
หรือไม่
ไม่ใช่
ใช่
เดินข้ามถนน
หยุด

Flowchart

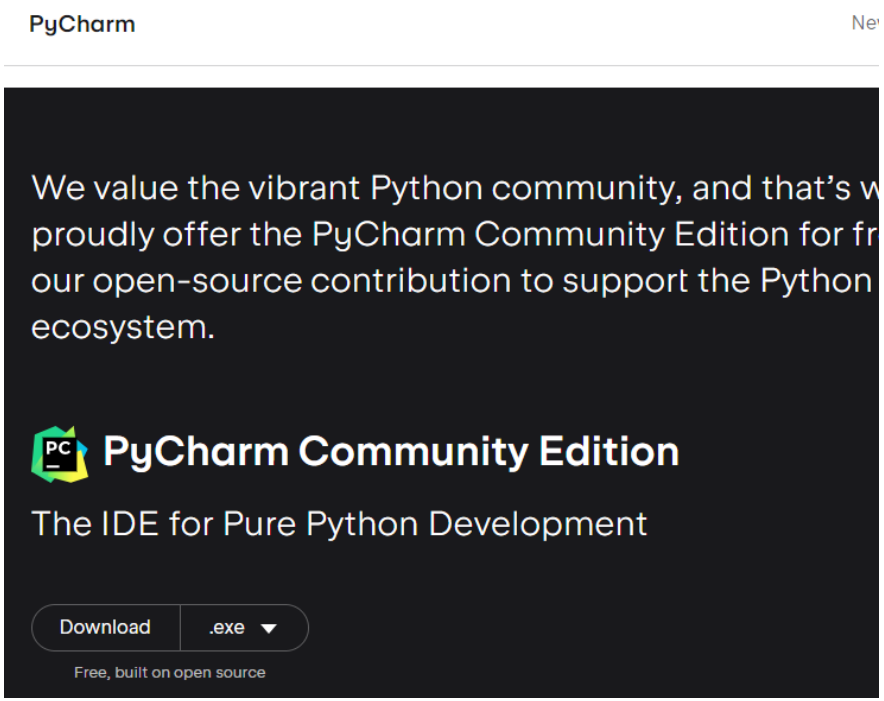

เงื่อนไข
จริง
เท็จ
ประโยคงาน
งานลำดับถัดไป


เงื่อนไข
จริง
เท็จ
ประโยคงาน 1
ประโยคงาน 2
งานลำดับถัดไป

Python IDE

- Reference：http://www.jetbrains.com/pycharm/download/

  https://www.python.org



Variables (Types)

10	Integer
5.5	Floating point
Hello	String
TRUE & FALSE	Boolean

Variables (Types)

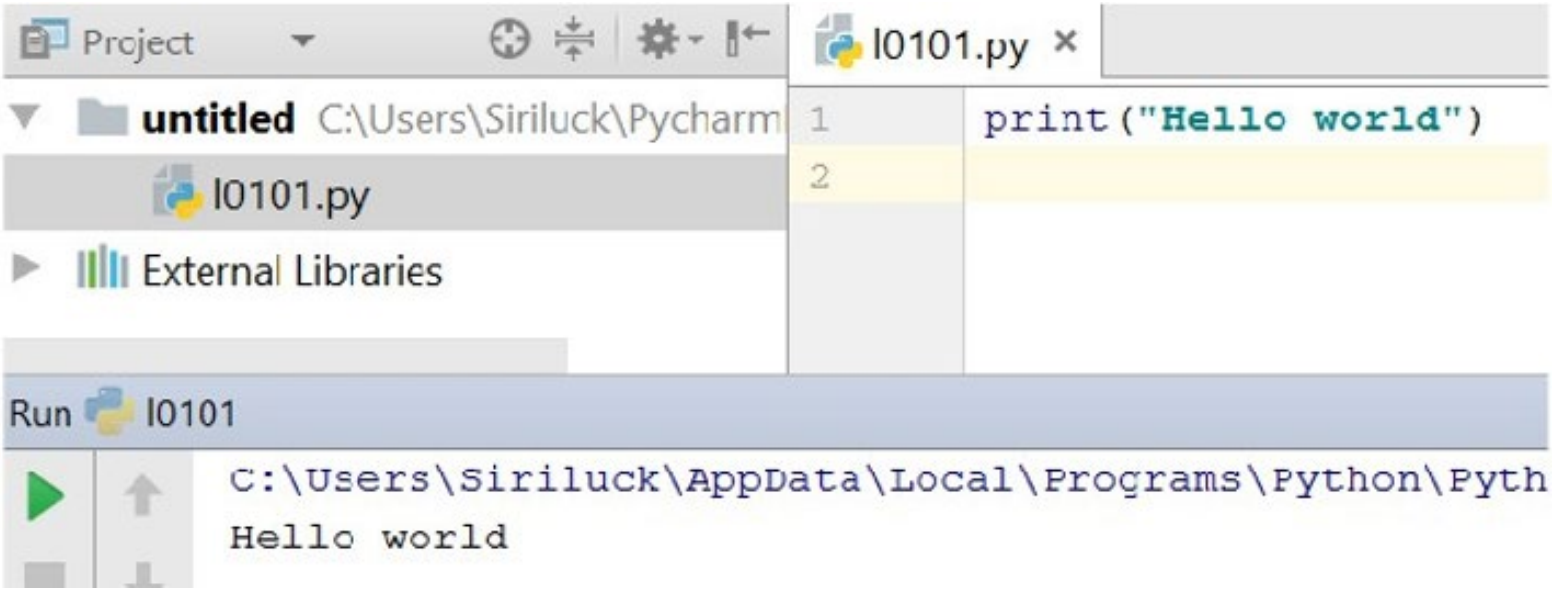
l0101.py

```
1   myInt = 10              #integer
2   myFloat = 5.5           #Floating point
3   myString = "Python"     #String
```

print

print (“Hello word”)

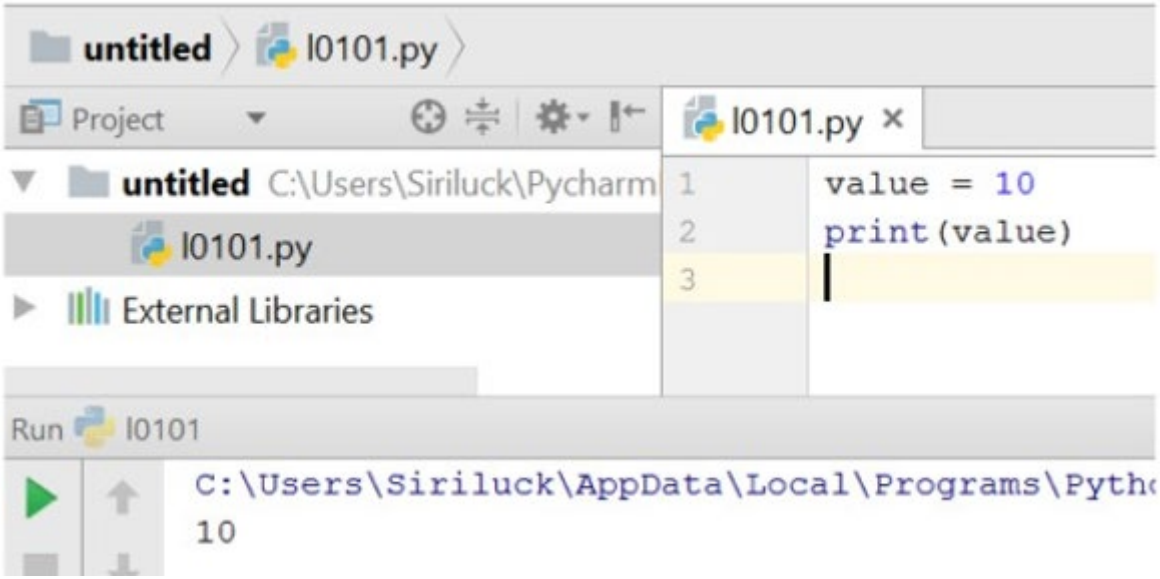
print (name_data)



print

print (“Hello word”)

print (name_data)

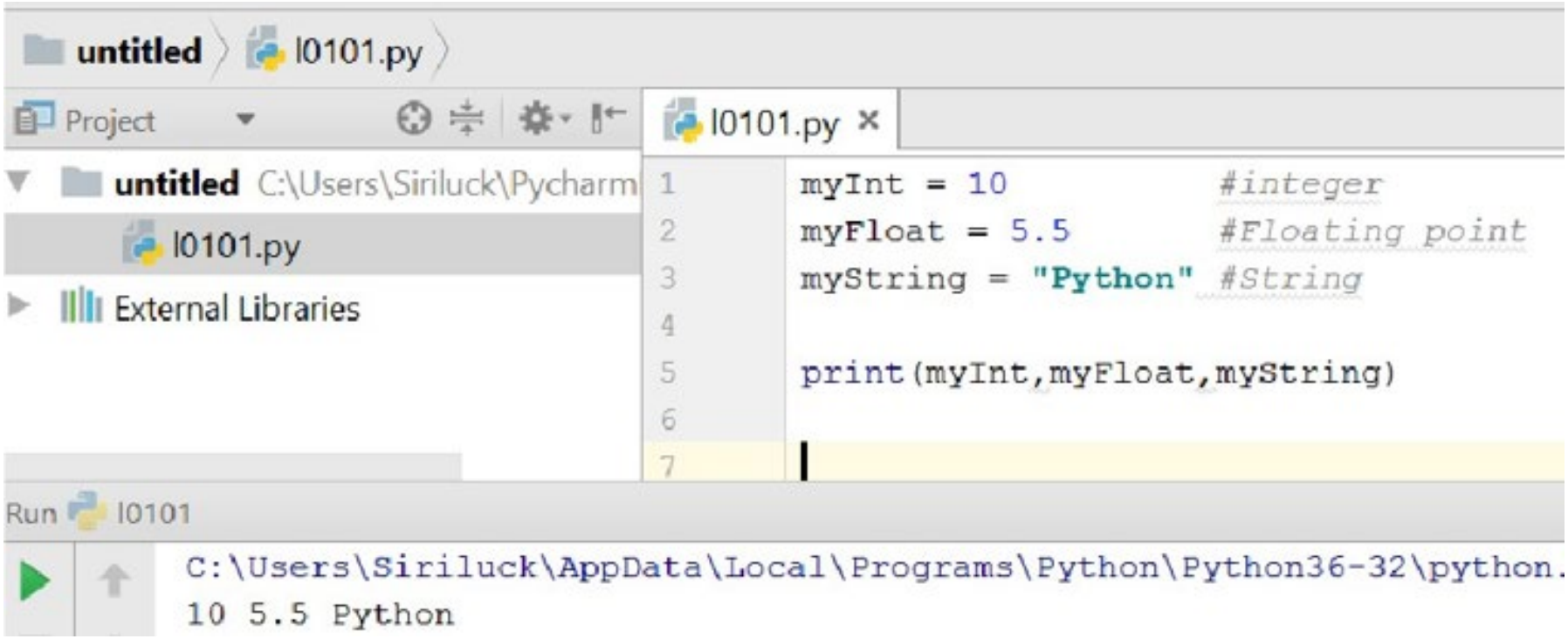


print('{:,.3f}'. format(value))

print('value is %.2f' %(value))

print

print (a, b)



Variables (operators)

+	addition
-	subtraction
*	multiplicationi
/	classic division
%	modulus
//	floor division
**	Exponentiation

Variables (operators)

(50 - 5 * 6) / 4

50 - 5 * 6 / 4

17 / 3

17 // 3

17 % 3

5 * 3 + 2

Operator	Description
**	Exponentiation
* / % //	Multiply, divide, modulo and floor division
+ -	Addition and subtraction

highest

lowest

Variables (Strings)

"John"

"hello"

'123'

"8/5"

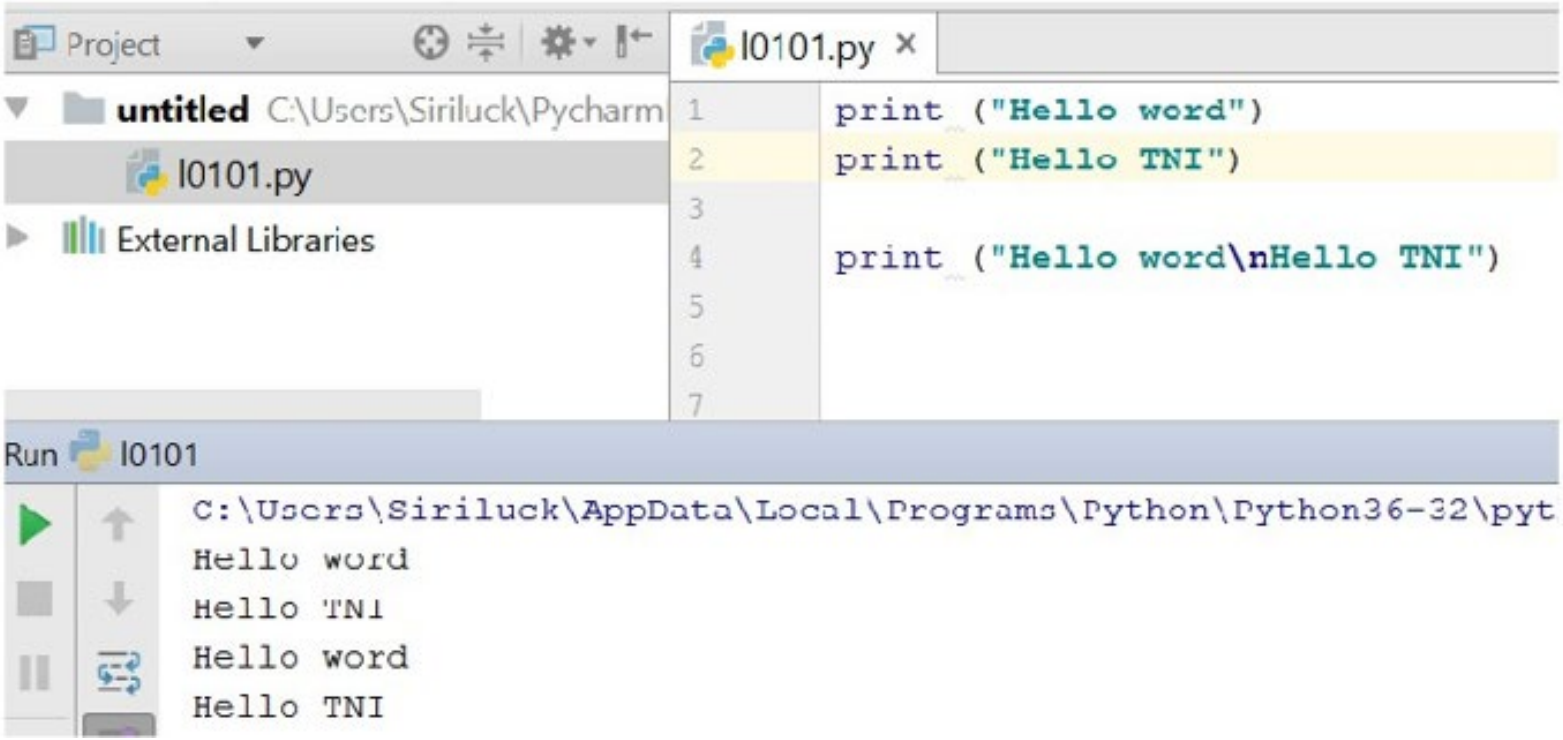
"....."	double quote
'.....'	single quote

print (“Hello word”)

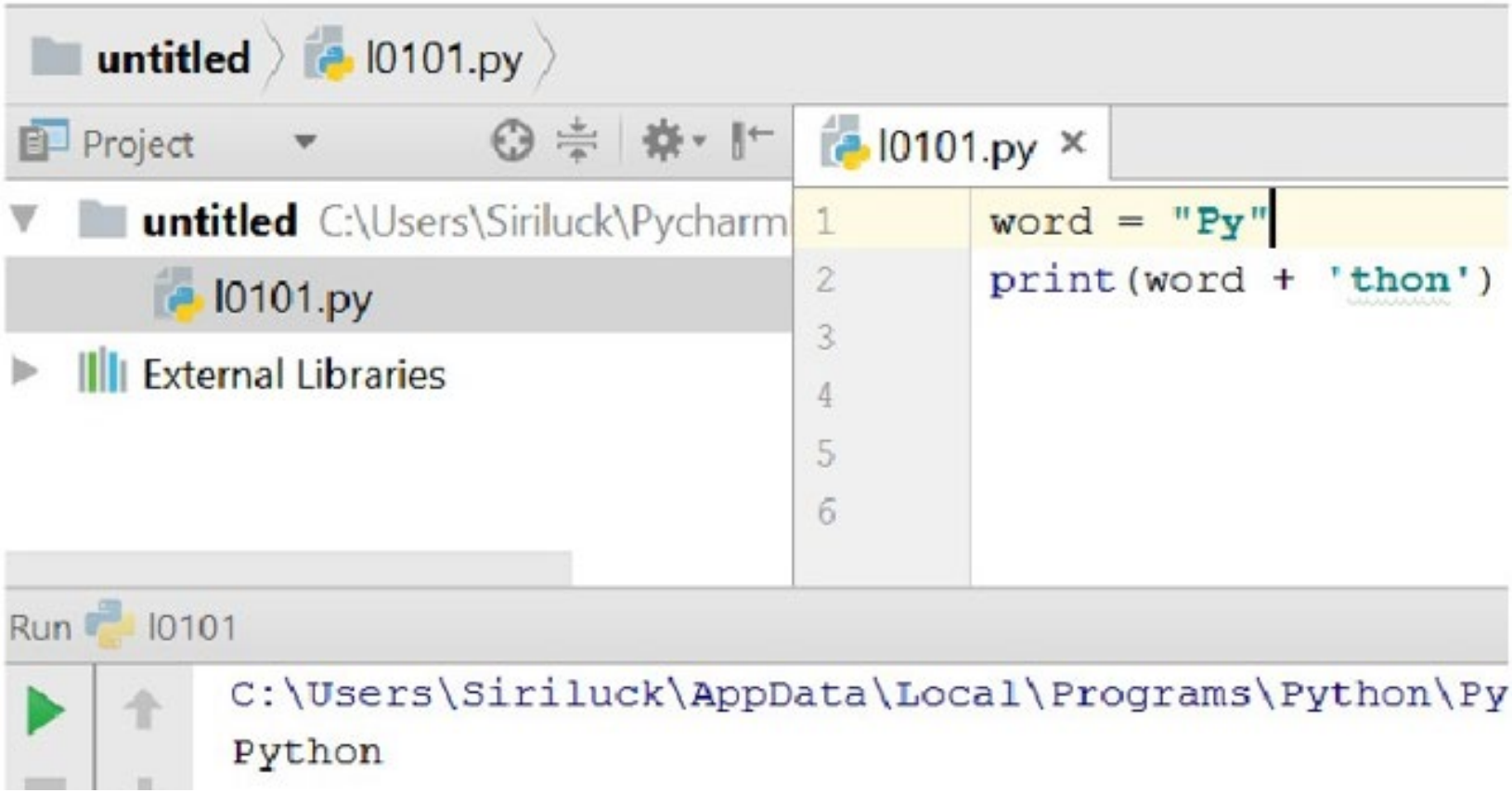
print (“Hello TNI”)

or

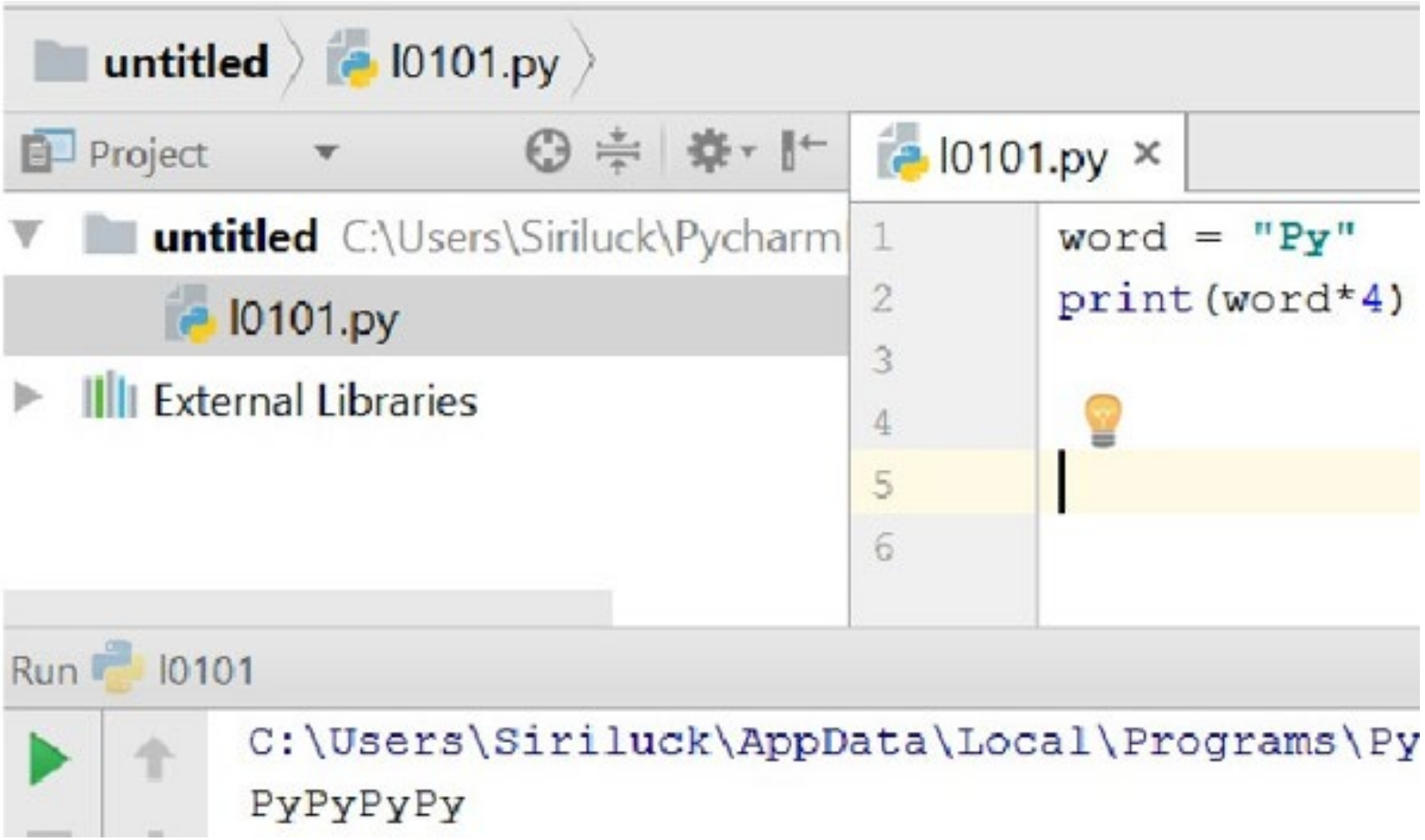
print (“Hello word\nHello TNI”)



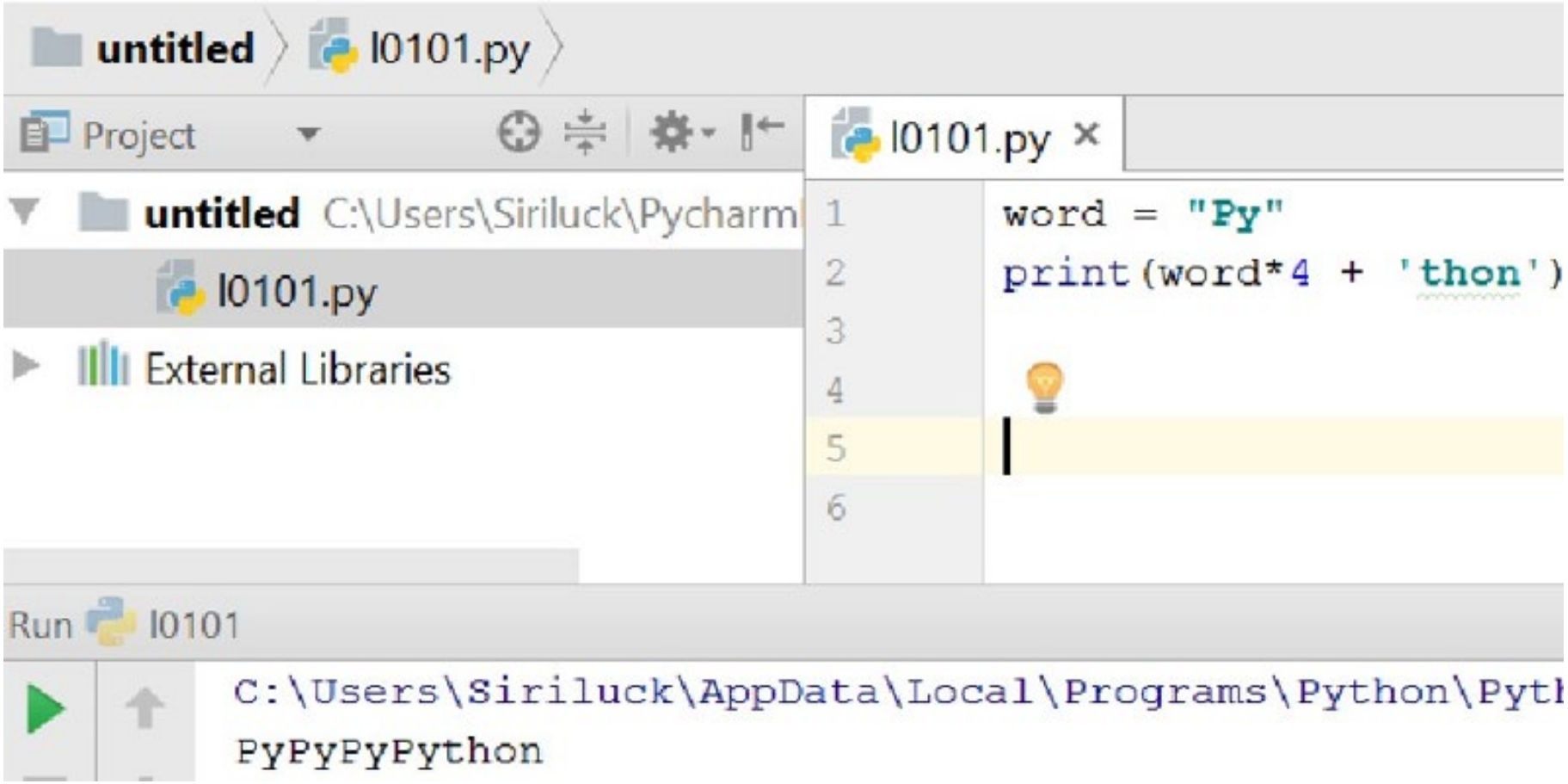
word = “Py”
print(word + 'thon')



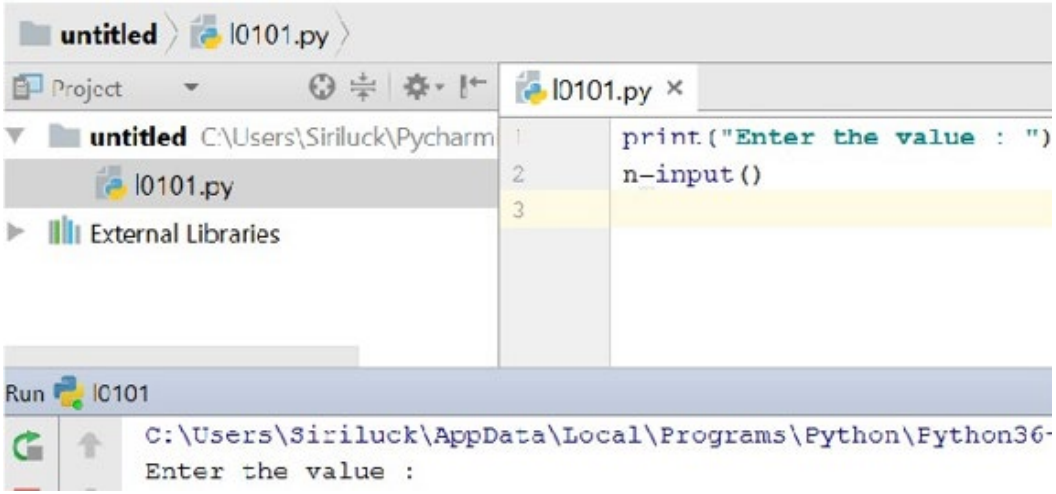
word = “Py”
print(word*4)

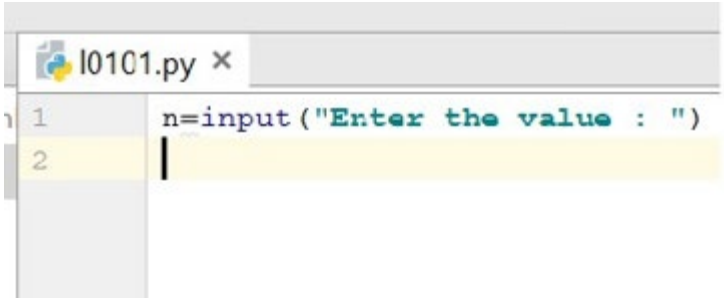


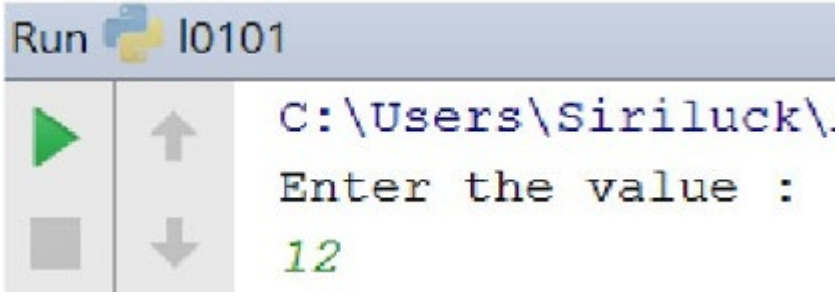
word = “Py”
(word*4 + ‘thon’)



```
print("Enter the value : ")
n=input()
or
n=input("Enter the value : ")
```







```
print("Enter the value : ")
n=input()
or
n=input("Enter the value : ")


if n = 2
A = n * 2          // 22        print (A)
A = int(n) * 2     // 4         print(str(A))
```

Question

- เขียนโปรแกรมคำนวณ พ. ศ. เกิดจากการรับอายุ (ตัวเลข)

```
How old are you : 31
So,you're born in 2535
```

- โปรแกรมคำนวณพื้นที่สี่เหลี่ยม จากการรับค่าความกว้าง

```
Enter the length of side : 12
Area of square : 144
```

conditionals and loops

Comparison operator

operator	Description	Example
=	Checks if the value of two operands is equal or not, if yes then condition becomes true.	(a == b) is not true.
!=	Checks if the value of two operands is equal or not, if values are not equal then condition becomes true.	(a != b) is true.
<>	Checks if the value of two operands is equal or not, if values are not equal then condition becomes true.	(a <> b) is true. This is similar to != operator.
>	Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true.	(a > b) is not true.

Comparison operator

operator	Description	Example
<	Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true.	(a < b) is true.
>=	Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true.	(a >= b) is not true.
<=	Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true.	(a <= b) is true.

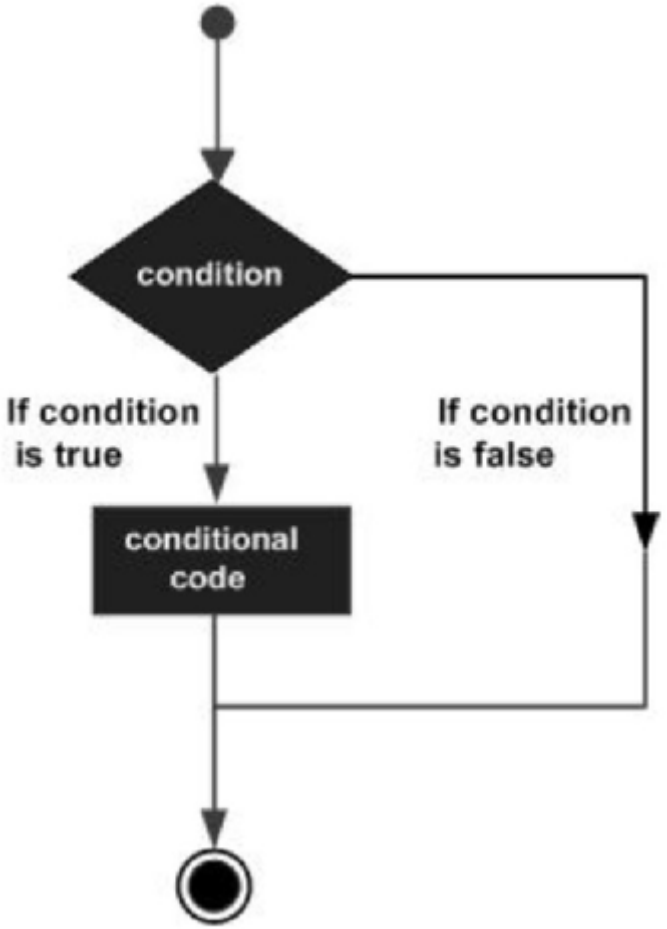
Comparison operator

operator	Description	Example
=	Simple assignment operator, Assigns values from right side operands to left side operand	c = a + b will assigne value of a + b into c
+=	Add AND assignment operator, It adds right operand to the left operand and assign the result to left operand	c += a is equivalent to c = c + a
-=	Subtract AND assignment operator, It subtracts right operand from the left operand and assign the result to left operand	c -= a is equivalent to c = c - a
*=	Multiply AND assignment operator, It multiplies right operand with the left operand and assign the result to left operand	c *= a is equivalent to c = c * a

Comparison operator

operator	Description	Example
/=	Divide AND assignment operator, It divides left operand with the right operand and assign the result to left operand	c /= a is equivalent to c = c / a
%=	Modulus AND assignment operator, It takes modulus using two operands and assign the result to left operand	c %= a is equivalent to c = c % a
**=	Exponent AND assignment operator, Performs exponential (power) calculation on operators and assign value to the left operand	c **= a is equivalent to c = c ** a
//=	Floor Dividion and assigns a value, Performs floor division on operators and assign value to the left operand	c //= a is equivalent to c = c // a

Flowchart : if

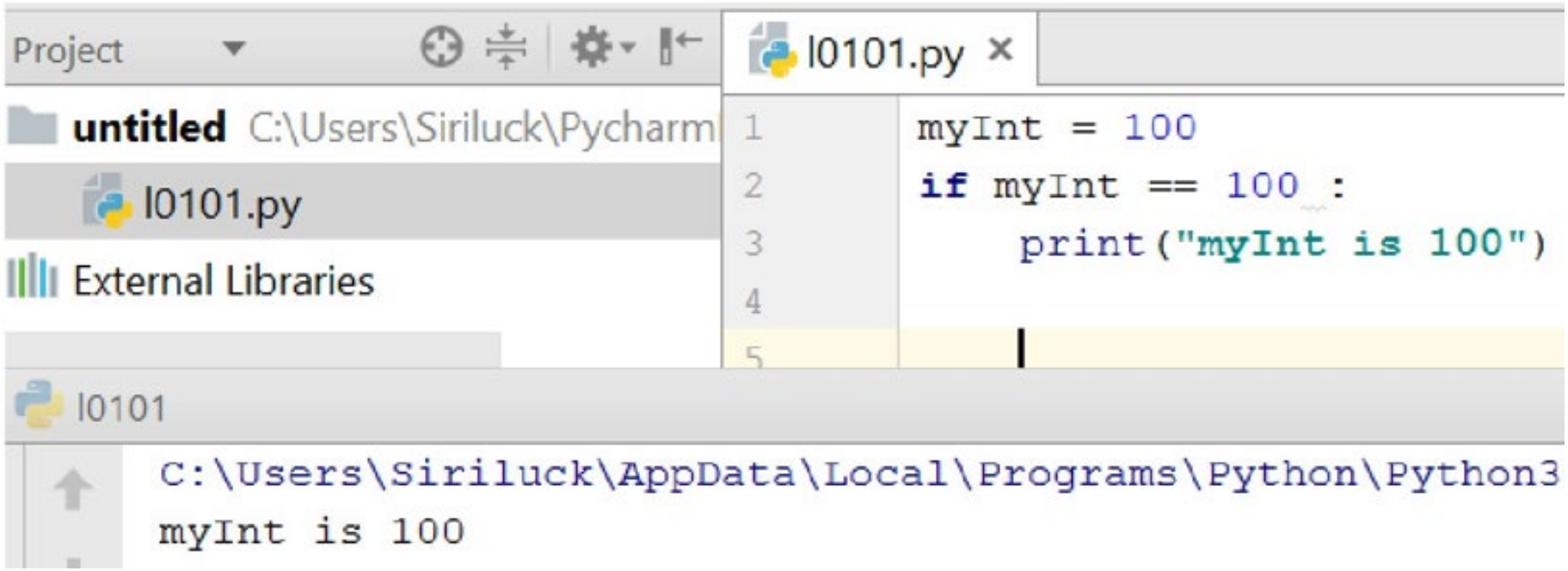


Flowchart : if


```
if expression:
    #statement goes here
    #.............
    #.........
```

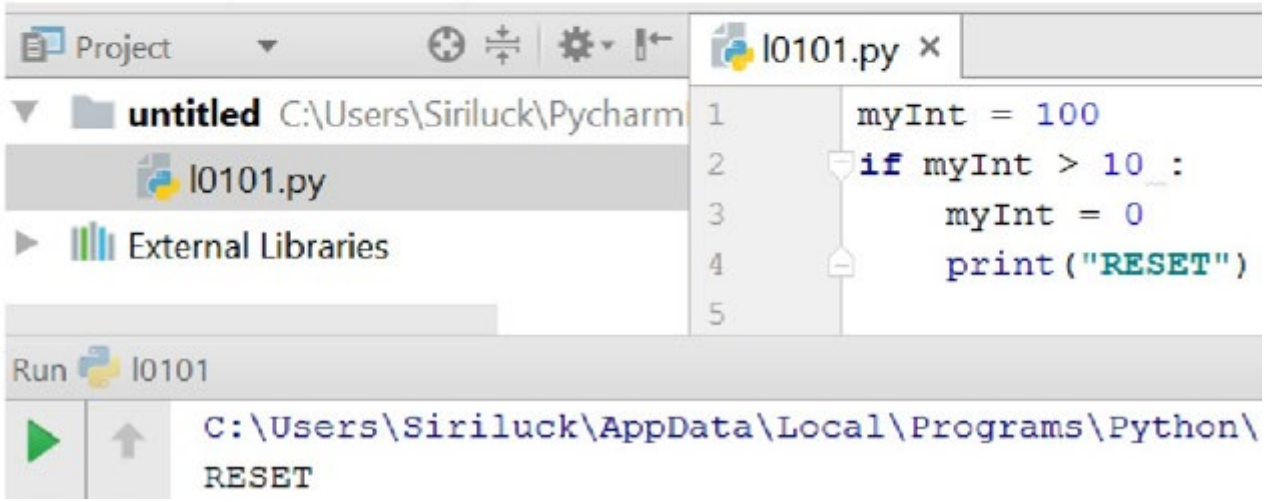
Flowchart : if


```
myInt = 100
if myInt == 100 :
    print("myInt is 100")
```



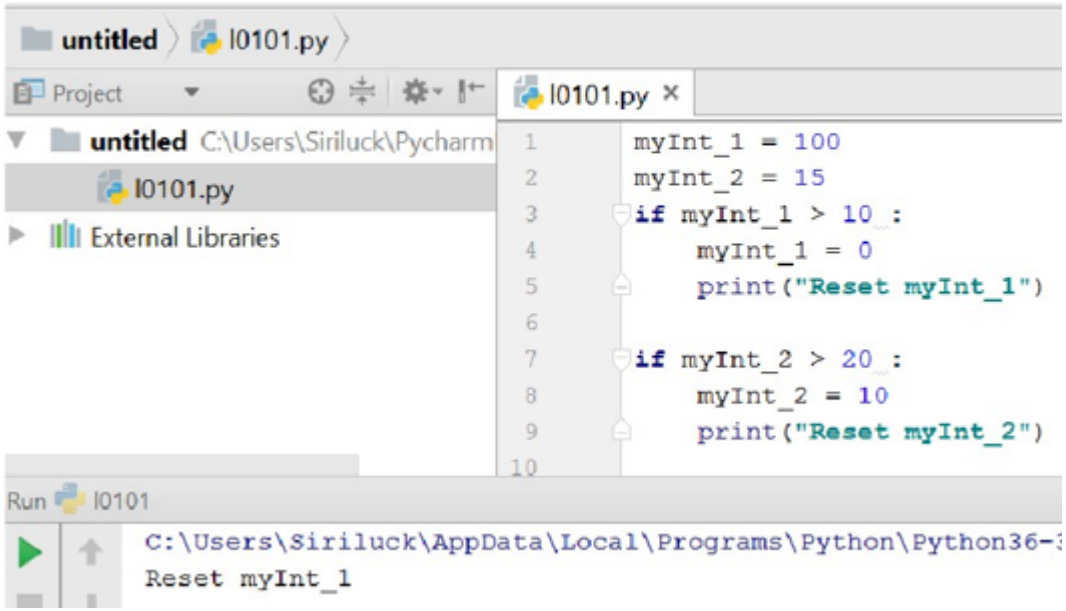
Flowchart : if


```
myInt = 100
if myInt > 10 :
    myInt = 0
    print("RESET")
```

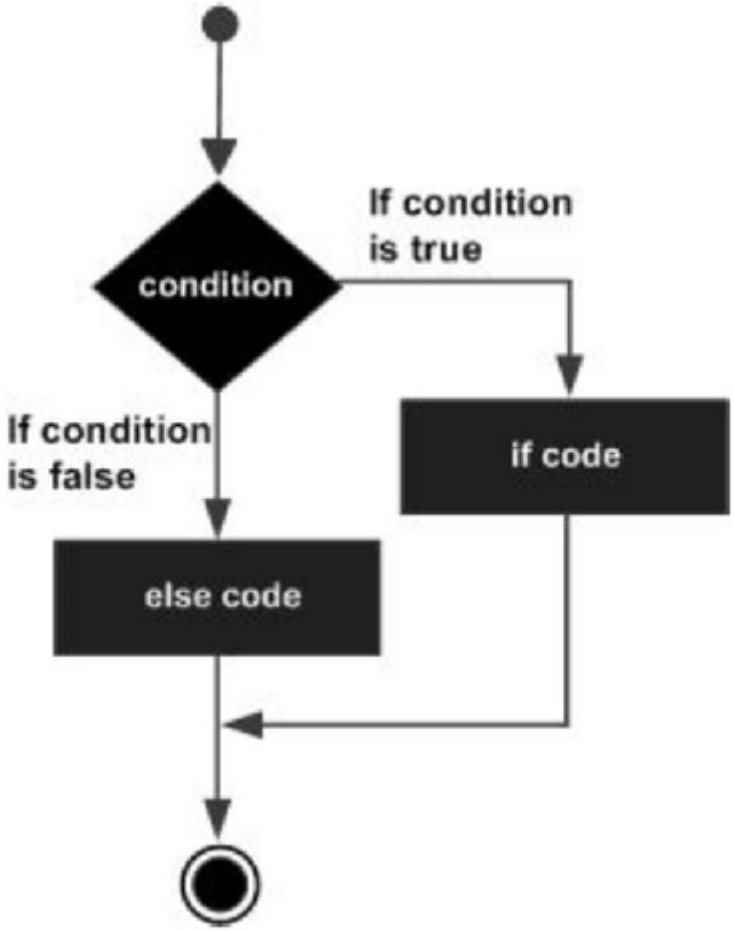


Flowchart : if


```
myInt_1 = 100
myInt_2 = 15
if myInt_1 > 10 :
    myInt_1 = 0
    print("Reset myInt_1")

if myInt_2 > 20 :
    myInt_2 = 10
    print("Reset myInt_2")
```



Flowchart : else if

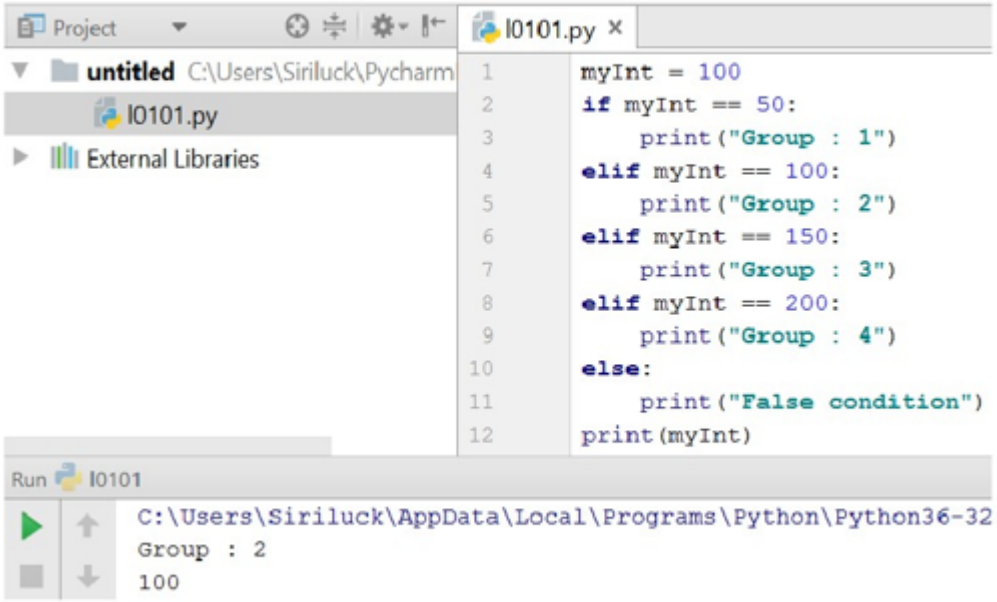


Flowchart : else if

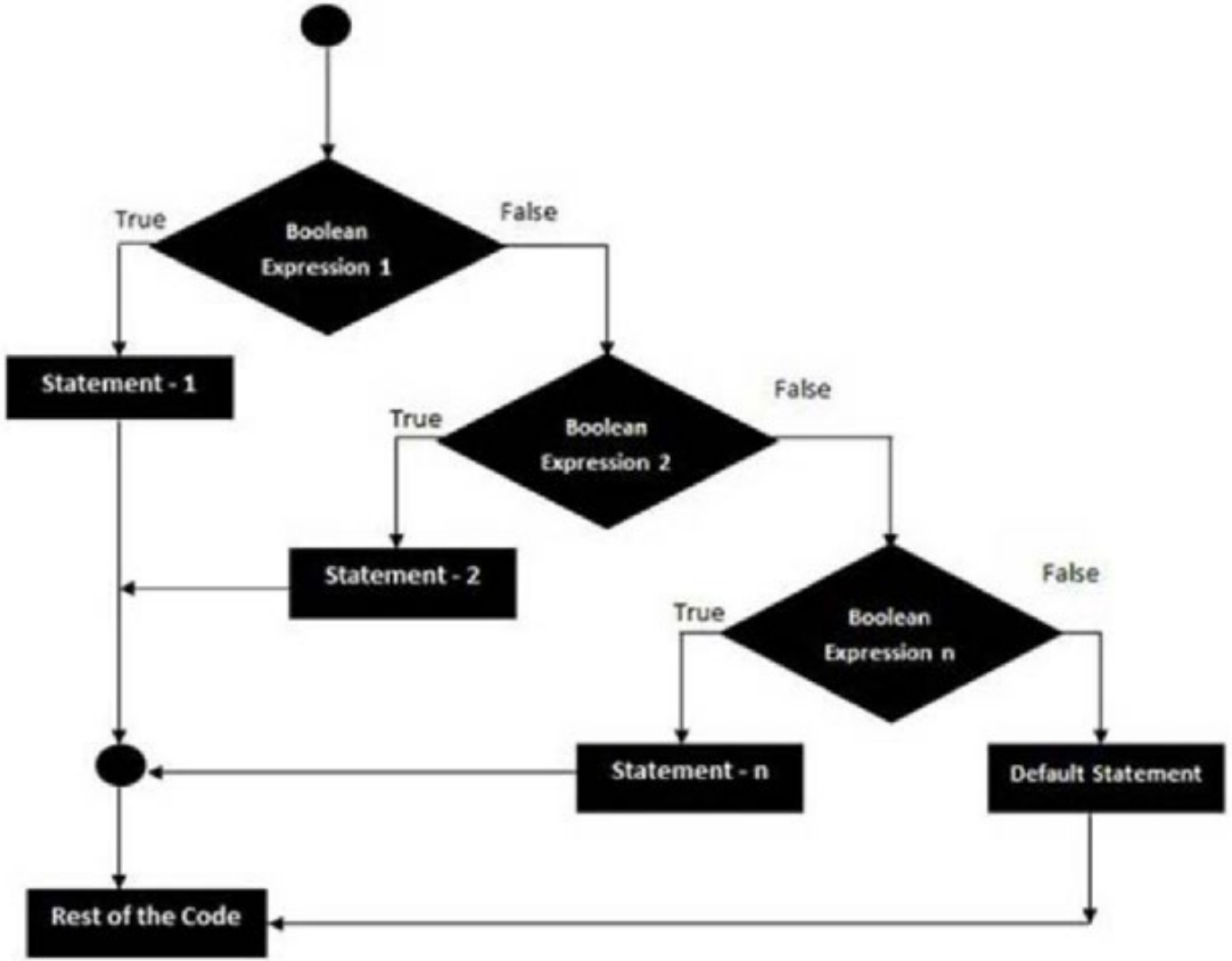

```
if expression1:
    statement(s)
elif expression2:
    statement(s)
elif expression3:
    statement(s)
else:
    statement(s)
```

Flowchart : else if


```
myInt = 100
if myInt == 50:
    print("Group : 1")
elif myInt == 100:
    print("Group : 2")
elif myInt == 150:
    print("Group : 3")
elif myInt == 200:
    print("Group : 4")
else:
    print("False condition")
print(myInt)
```



Flowchart : else if

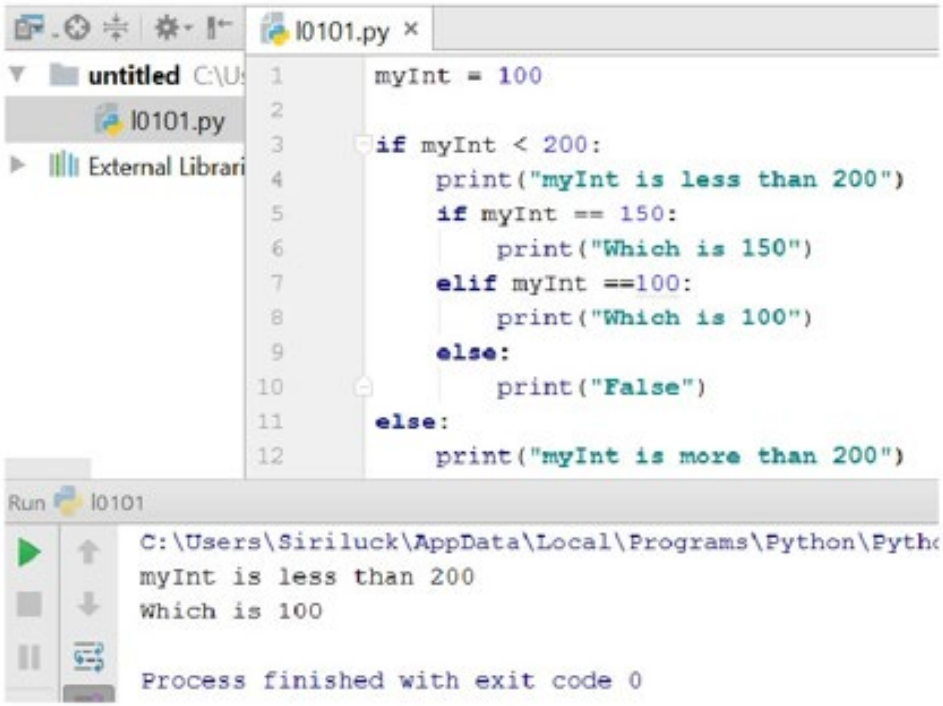


Flowchart : else if

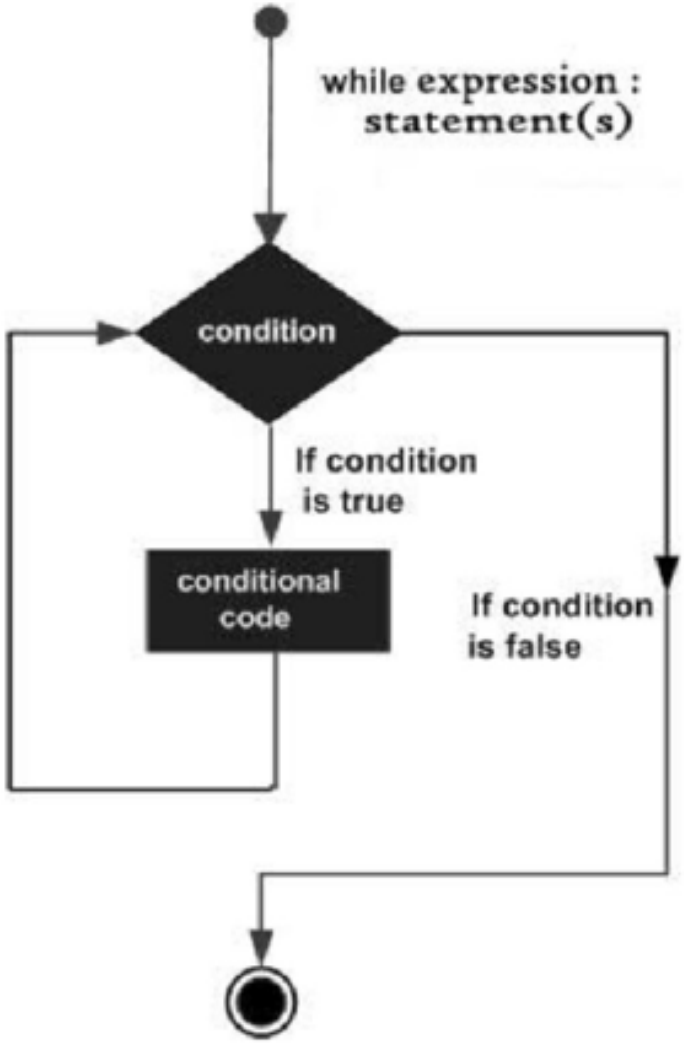

```
if expression1:
    statement(s)
    if expression2:
        statement(s)
    elif expression3:
        statement(s)
    else
        statement(s)
elif expression4:
    statement(s)
else:
    statement(s)
```

Flowchart : else if


```
myInt = 100

if myInt < 200:
    print("myInt is less than 200")
    if myInt == 150:
        print("Which is 150")
    elif myInt ==100:
        print("Which is 100")
    else:
        print("False")
else:
    print("myInt is more than 200")
```



Flowchart : while loop

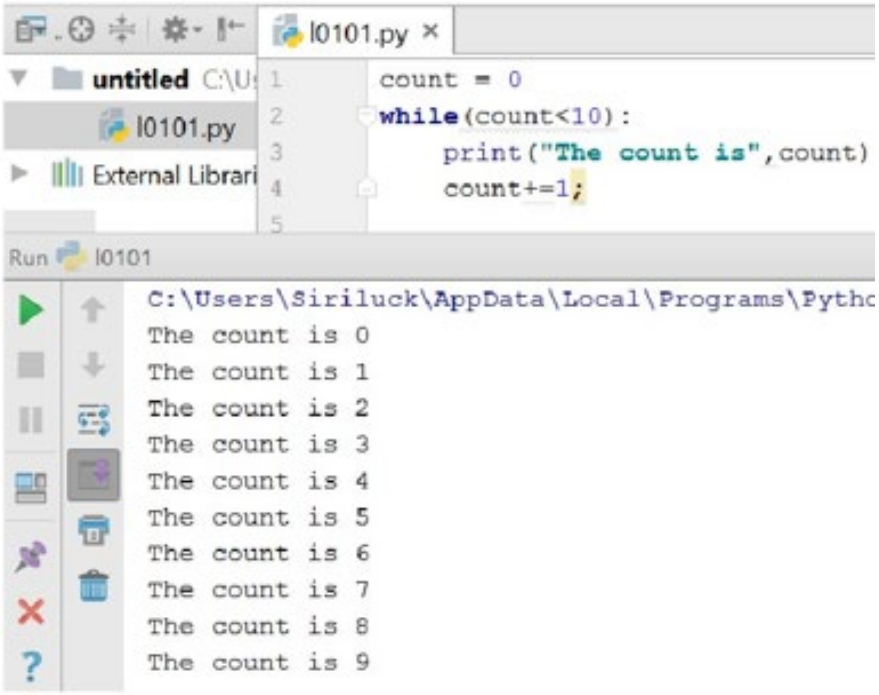


Flowchart : while loop

while *expression*:
statement(s)

Flowchart : while loop


```
count = 0
while(count<10):
    print("The count is",count)
    count+=1;
```



range

range(stop)

range([start], stop[, step])

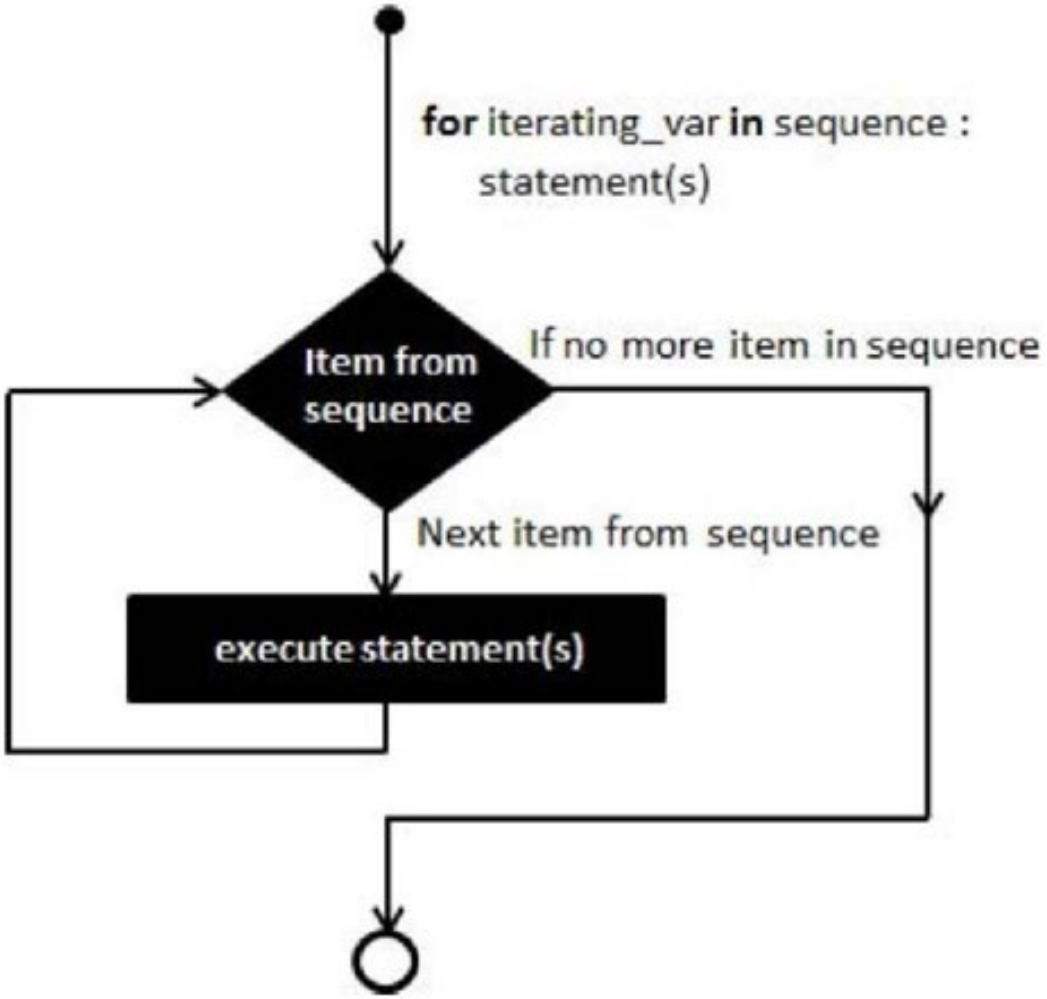
```
>>> # One parameter
>>> for i in range(5):
...     print(i)
...
0
1
2
3
4
```

```
>>> # Two parameters
>>> for i in range(3, 6):
...     print(i)
...
3
4
5
```

```
>>> # Three parameters
>>> for i in range(4, 10, 2):
...     print(i)
...
4
6
8
```

```
>>> # Going backwards
>>> for i in range(0, -10, -2):
...     print(i)
...
0
-2
-4
-6
-8
```

Flowchart : for loop

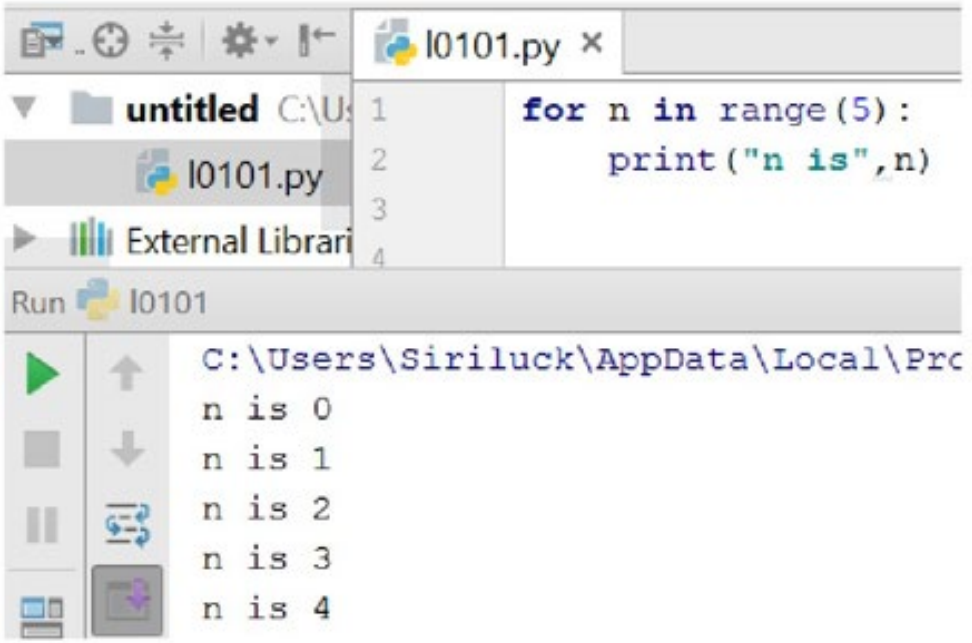


Flowchart : for loop


```
for iterating_var in sequence:
    statements(s)
```

Flowchart : for loop


```
for n in range(5):
    print("n is",n)
```



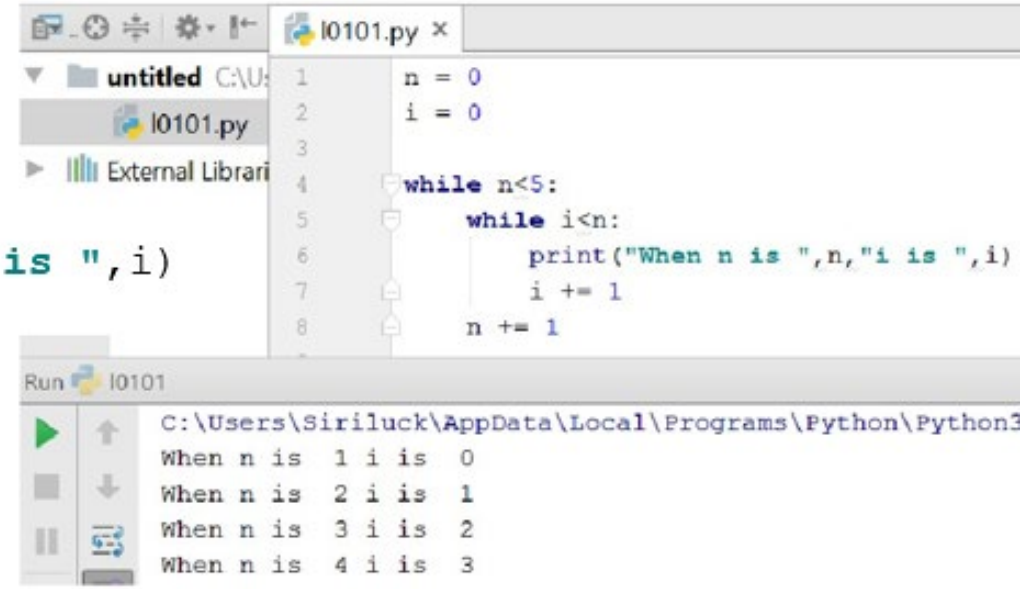
Flowchart : Nested loop


```
for iterating_var in sequence:
    for iterating_var in sequence:
        statements(s)
    statements(s)
```

```
while expression:
    while expression:
        statement(s)
    statement(s)
```

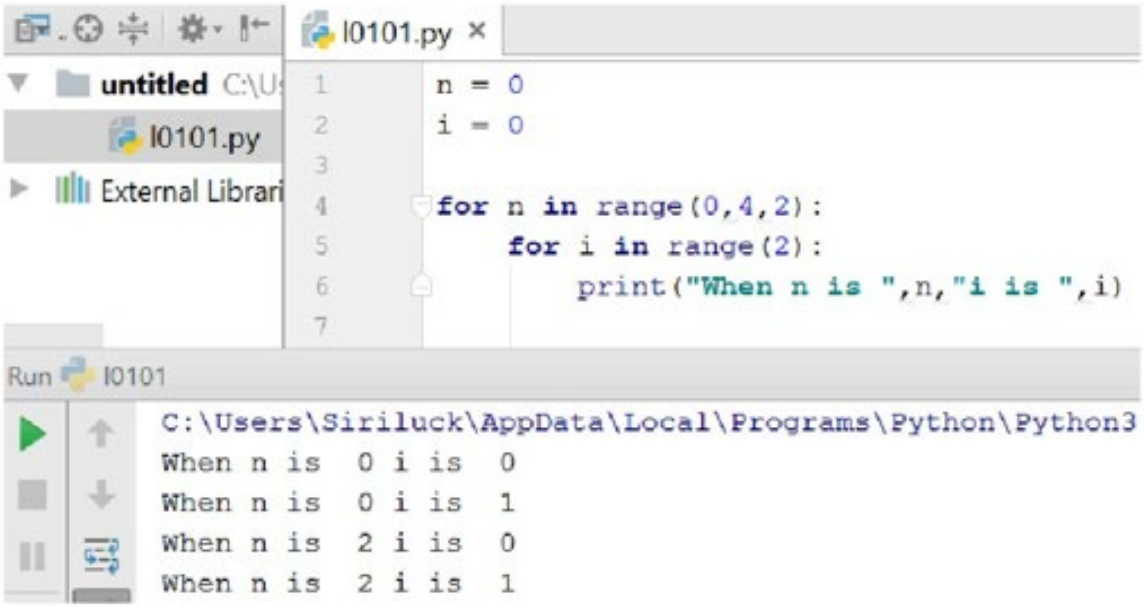
Flowchart : Nested loop


```
n = 0
i = 0

while n<5:
    while i<n:
        print("When n is ",n,"i is ",i)
        i += 1
    n += 1
```



Flowchart : Nested loop


```
n = 0
i = 0

for n in range(0,4,2):
    for i in range(2):
        print("When n is ",n,"i is ",i)
```



Question

- เขียนโปรแกรมสำหรับคำนวณเงินของพนักงานจากระยะเวลาในการทำงาน โดยมีเงื่อนไขดังต่อไปนี้
  - จำนวนชั่วโมงน้อยกว่าหรือเท่ากับ 8 ชั่วโมง ชั่วโมงละ 50 บาท
  - จำนวนชั่วโมงมากกว่า 8 ชั่วโมง จำนวนที่มากกว่า 8 ชั่วโมง คิดชั่วโมงละ 80 บาท
  - หากทำงานมากกว่า 12 ชั่วโมง จะได้รับเงินทั้งหมด 800 บาท

Enter working time : 4	Enter working time : 11	Enter working time : 100
Total wage is 200	Total wage is 640	Total wage is 800

Question

- เขียนโปรแกรมสำหรับรับตัวเลขไปเรื่อย ๆ จนกว่าผู้ใช้จะป้อนเลข 0 เข้ามาในระบบ พร้อมทั้งหาผลรวมของตัวเลขที่ป้อนเข้ามาทั้งหมด : โดยใช้ loop while

```
Enter number : 3.2
Enter number : 53
Enter number : 1
Enter number : 3
Enter number : 0
Sum :  60.2
```

สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น)
Technology Promotion Association (Thailand-Japan)

สถาบันเทคโนโลยีไทย-ญี่ปุ่น
Thai-Nichi Institute of Technology
泰日工業大学

String และ List

Access values

var = "python"

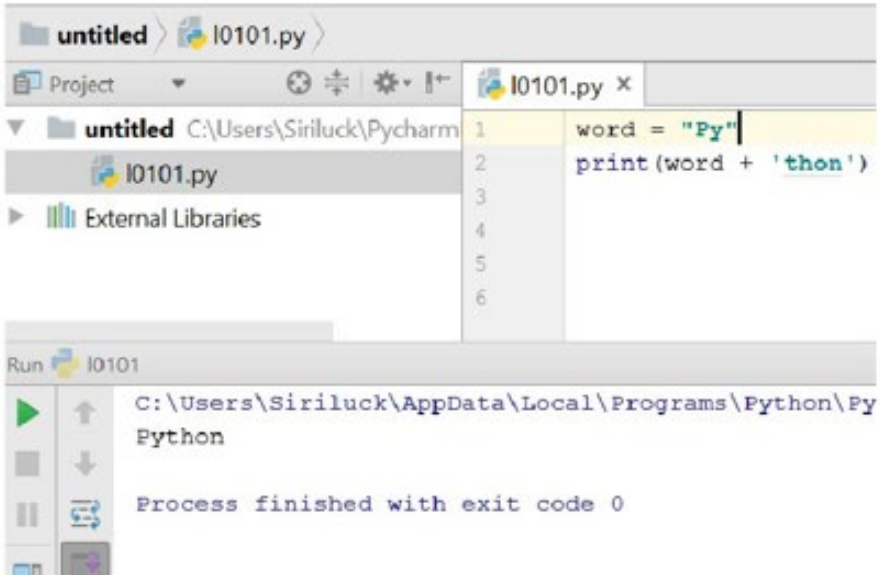
0	1	2	3	4	5
p	y	t	h	o	n

String

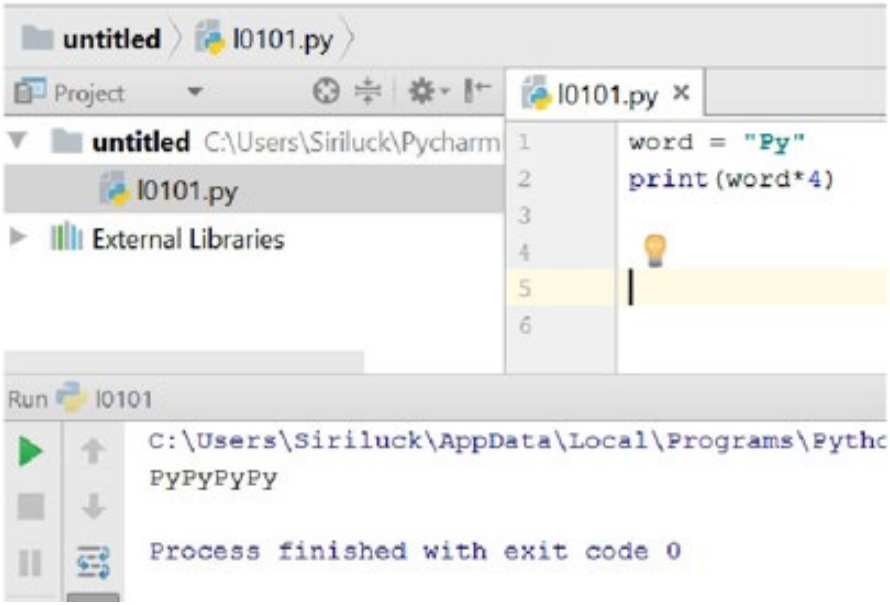
Operator	Description	Example
+	Concatenation - Adds values on either side of the operator	a + b will give HelloPython
	Repetition - Creates new strings, concatenating multiple copies of the same string	a*2 will give -HelloHello
[]	Slice - Gives the character from the given index	a[1] will give e
[:]	Range Slice - Gives the characters from the given range	a[1:4] will give ell
in	Membership - Returns true if a character exists in the given string	H in a will give 1
not in	Membership - Returns true if a character does not exist in the given string	M not in a will give 1
%	Format - Performs String formatting	See at next section

String : + , *


```
word = “Py”
print(word + 'thon')
```




```
word = “Py”
print(word*4)
```



String : [ ] , [:]

0	1	2	3	4	5
p	y	t	h	o	n

word = “Python”

print(word[3])

```
1  word = "Python"
2  print(word[3])
3
4
5
```

Run l0101

```
C:\Users\Siriluck\AppData\Local\Pr
h
```

word = “Python”

print(word[2:4])

```
1  word = "Python"
2  print(word[2:4])
3
4
5
```

Run l0101

```
C:\Users\Siriluck\AppData\Local\Pr
th
```

String : %

Format Symbol	Conversion
%c	character
%s	string conversion via str() prior to formatting
%d , %i	signed decimal integer
%f	floating point real number

```
1  name = input("Enter your name : ")
2  w = int(input("Enter your weight : "))
3
4  print ("My name is %s and weight is %d kg!" % (name, w))
5
```

Run I0101

```
C:\Users\Siriluck\AppData\Local\Programs\Python\Python36
Enter your name : Zara
Enter your weight : 30
My name is Zara and weight is 30 kg!
```

String : find()

Description

It determines if string *str* occurs in string, or in a substring of string if starting index *beg* and ending index *end* are given.

Syntax

```
str.find(str, beg=0, end=len(string))
```

Parameters

- **str** -- This specifies the string to be searched.
- **beg** -- This is the starting index, by default its 0.
- **end** -- This is the ending index, by default its equal to the length of the string.

Return Value

Index if found and -1 otherwise.

String : find()

0	1	2	3	4	5
p	y	t	h	o	n

```
1  word = "python"
2  str = "th"
3
4  print (word.find(str))
5  print (word.find(str,1))
6  print (word.find(str,1,3))
7  print (word.find(str,5))
```

Run l0101

```
C:\Users\Siriluck\AppData\Local\Prc
2
2
-1
-1
```

List


```
list1 = ['physics', 'chemistry', 1997, 2000]
list2 = [1, 2, 3, 4, 5 ]
list3 = ["a", "b", "c", "d"]
```

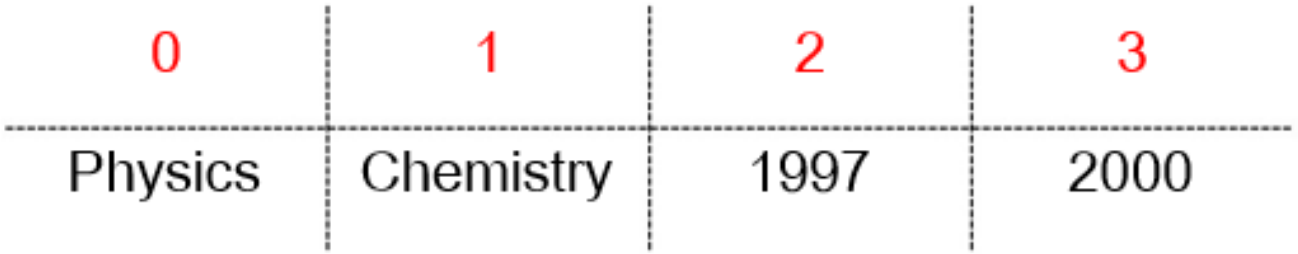
String vs List

word = “python”

0	1	2	3	4	5
p	y	t	h	o	n

```
list1 = ['physics', 'chemistry', 1997, 2000]
list2 = [1, 2, 3, 4, 5 ]
list3 = ["a", "b", "c", "d"]
```

List [ ] , [:]




```
list_name = ['physics','chemistry',1997,2000]

print(list_name[1])
print(list_name[1:4])
```

```
C:\Users\Siriluck\AppData\
chemistry
['chemistry', 1997, 2000]
```

List : update value

0	1	2	3
Physics	Chemistry	1997	2000

```
list_name = ['physics','chemistry',1997,2000]

list_name[2] = "mathematic"
print(list_name)
```

0	1	2	3
Physics	Chemistry	mathematic	2000

```
C:\Users\Siriluck\AppData\Local\Programs\Pyt
['physics', 'chemistry', 'mathematic', 2000]
```

List : Nested list

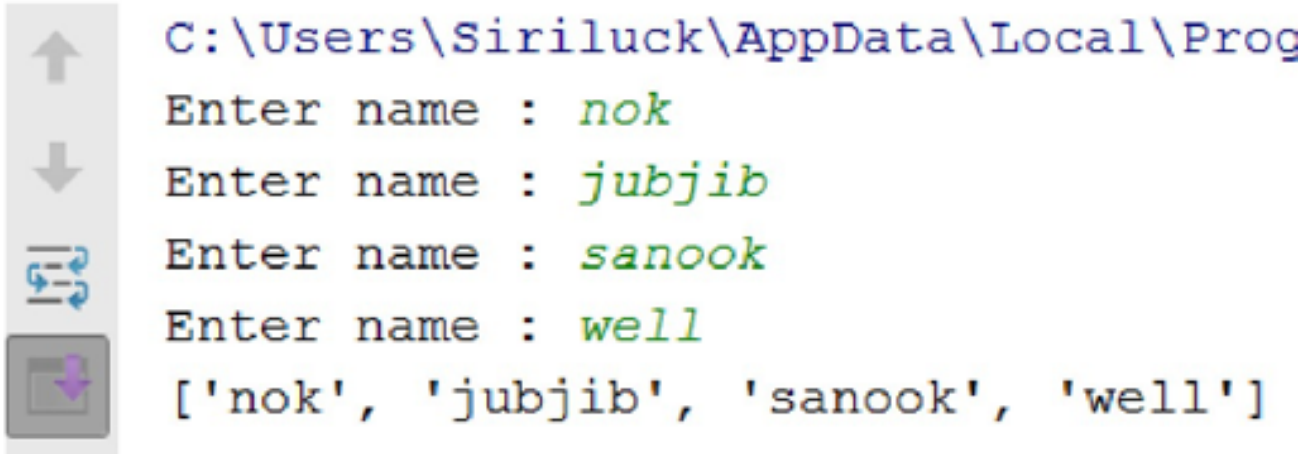
1	2	3	4
5	6	7	8
9	10	11	12

```
matrix = [ [1,2,3,4],[5,6,7,8],[9,10,11,12]]

print("matrix[1][2] : ",matrix[1][2])
```

```
C:\Users\Siriluck\A
matrix[1][2] :  7
```

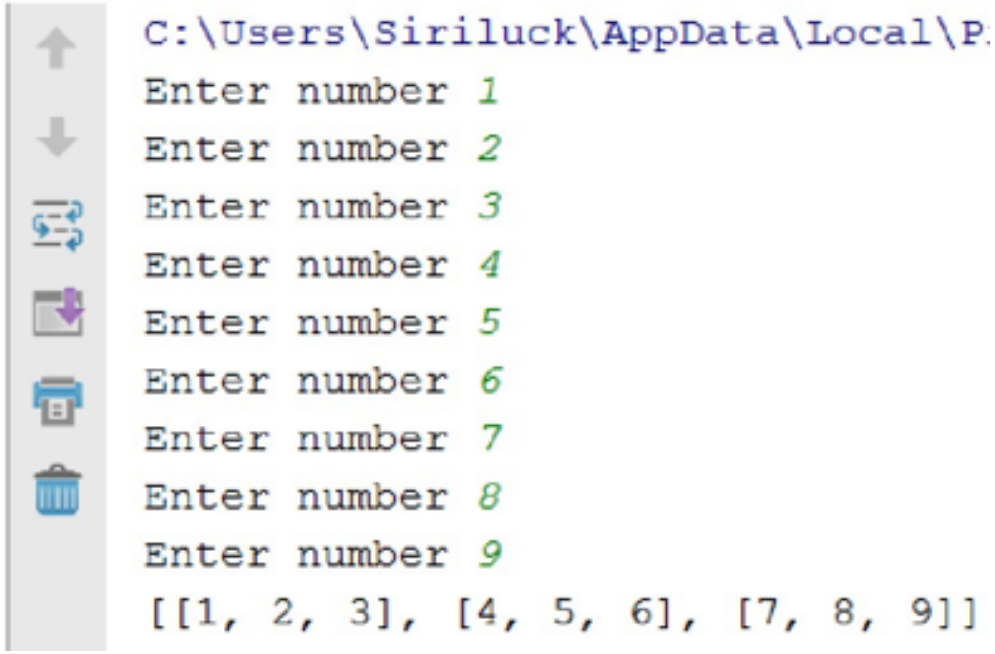
List : input nested list : append


```
matrix = []

for i in range(4):
    n = input("Enter name : ")
    matrix.append(n)

print(matrix)
```



List : input nested list : append


```
matrix = []

for i in range(3):
    matrix.append([])
    for j in range(3):
        n = int(input("Enter number "))
        matrix[i].append(n)
print(matrix)
```



List : basic operations

Python Expression	Results	Description
len([1, 2, 3])	3	Length
[1, 2, 3] + [4, 5, 6]	[1, 2, 3, 4, 5, 6]	Concatenation
['Hi!'] * 4	['Hi!', 'Hi!', 'Hi!', 'Hi!']	Repetition
3 in [1, 2, 3]	True	Membership
max([1, 2, 3])	3	Maximum value
min([1, 2, 3])	1	Minimum value

List : length

Python Expression	Results	Description
max([1, 2, 3])	3	Maximum value

```
list_name = [1,4,6,2,3,8,10]

print("Max : ",max(list_name))
print("Min : ",min(list_name))
```

```
C:\Users\Siriluck\AppData\Local\
Max :  10
Min :  1
```

List : length

Python Expression	Results	Description
len([1, 2, 3])	3	Length

```
list_name = ['physics','chemistry','mathematic',1997,2000]

print(list_name)
print(len(list_name))
```

```
C:\Users\Siriluck\AppData\Local\Programs\Python\Pyt
['physics', 'chemistry', 'mathematic', 1997, 2000]
5
```

List : delete

0	1	2	3	4
Physics	Chemistry	mathematic	1997	2000

```
list_name = ['physics','chemistry','mathematic',1997,2000]

print(list_name)
del(list_name[2])
print(list_name)
```

```
C:\Users\Siriluck\AppData\Local\Programs\Python\Py
['physics', 'chemistry', 'mathematic', 1997, 2000]
['physics', 'chemistry', 1997, 2000]
```

0	1	2	3
Physics	Chemistry	1997	2000

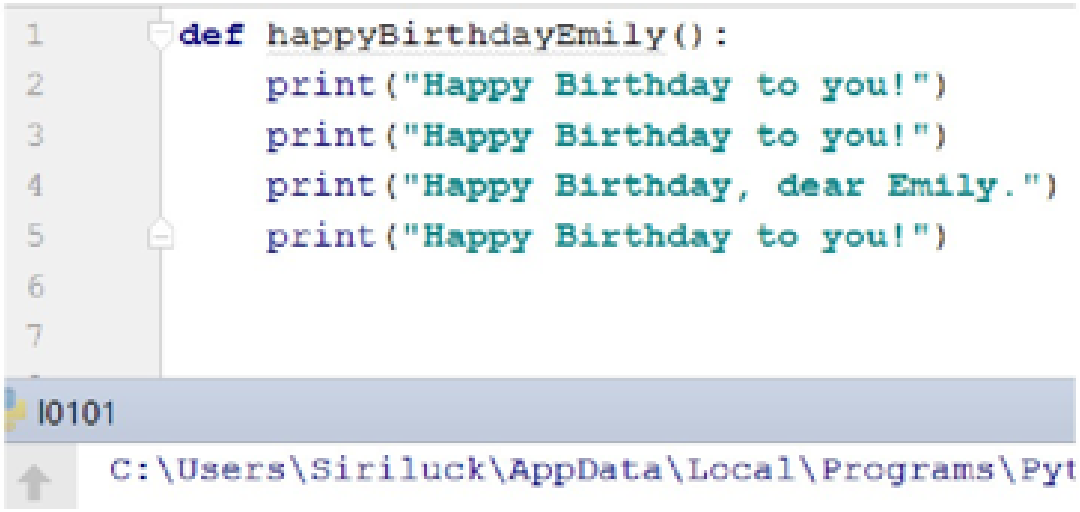
Function

Function : Definition


```
def function_name( parameters ):
    statement
    ....
    statement
    return expression
```

Function


```
def happyBirthdayEmily():
    print("Happy Birthday to you!")
    print("Happy Birthday to you!")
    print("Happy Birthday, dear Emily.")
    print("Happy Birthday to you!")
```



Function : Executable


```
1   def happyBirthdayEmily():
2       print("Happy Birthday to you!")
3       print("Happy Birthday to you!")
4       print("Happy Birthday, dear Emily.")
5       print("Happy Birthday to you!")
6
7   happyBirthdayEmily()
```

```
C:\Users\Siriluck\AppData\Local\Programs\Pyt
Happy Birthday to you!
Happy Birthday to you!
Happy Birthday, dear Emily.
Happy Birthday to you!

Process finished with exit code 0
```

Function


```
1   def happyBirthdayEmily():
2       print("Happy Birthday to you!")
3       print("Happy Birthday to you!")
4       print("Happy Birthday, dear Emily.")
5       print("Happy Birthday to you!")
6
7   def happyBirthdayJack():
8       print("Happy Birthday to you!")
9       print("Happy Birthday to you!")
10      print("Happy Birthday, dear Jack.")
11      print("Happy Birthday to you!")
12
13  happyBirthdayEmily()
14  happyBirthdayJack()
15
```

```
I0101
C:\Users\Siriluck\AppData\Local\Programs\Pyth
Happy Birthday to you!
Happy Birthday to you!
Happy Birthday, dear Emily.
Happy Birthday to you!
Happy Birthday to you!
Happy Birthday to you!
Happy Birthday, dear Jack.
Happy Birthday to you!
```

Function


```
1   def happyBirthday(name):
2       print("Happy Birthday to you!")
3       print("Happy Birthday to you!")
4       print("Happy Birthday, dear",name,".")
5       print("Happy Birthday to you!")
6
7   happyBirthday("Emily")
8   happyBirthday("Jack")
9
```

```
l0101
C:\Users\Siriluck\AppData\Local\Programs\Pytho
Happy Birthday to you!
Happy Birthday to you!
Happy Birthday, dear Emily .
Happy Birthday to you!
Happy Birthday to you!
Happy Birthday to you!
Happy Birthday, dear Jack .
Happy Birthday to you!
```

Function


```
1   def passVal(myVal):
2       myVal = 2
3       print("Inside function: ", myVal)
4
5   myVal = 1
6   passVal(myVal)
7   print("Outside Function ", myVal)
8
9
```

```
l0101

C:\Users\Siriluck\AppData\Local\Programs\
Inside function:  2
Outside Function  1
```

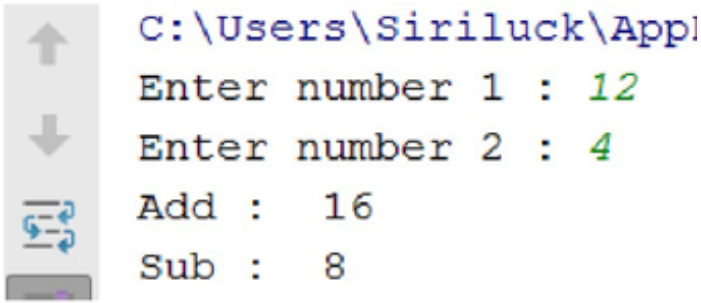
Function


```
def add(x,y) :
    z = x+y
    return z

a = int(input("Enter number 1 : "))
b = int(input("Enter number 2 : "))

c = add(a,b)
print ("Add : ",c)
```

```
C:\Users\Siriluck\Ap
Enter number 1 : 12
Enter number 2 : 3
Add :  15
```

Function


```
def add_sub(x,y) :
    z = x+y
    e = x-y
    return z,e

a = int(input("Enter number 1 : "))
b = int(input("Enter number 2 : "))

c,d = add_sub(a,b)
print ("Add : ",c)
print ("Sub : ",d)
```



Question

- ให้เขียนฟังก์ชันสำหรับ การบวก การลบ การคูณ และการหาร ค่า 2 ค่า จากนั้นส่งผลลัพธ์กลับมาแสดงผล (เขียนรวมในฟังก์ชันเดียว หรือเขียนแยกรายฟังก์ชันก็ได้)

```
Enter number 1 : 12
Enter number 2 : 3
Plus :  15
Minus :  9
Multiply :  36
Divide :  4.0
```