Hello world

Basic python for data analyst (Beginner)

##code

- variable
- data type
- data structure
- control flow
- function

In [12]:
```
## basic calculation

print(2-5)
print(6*3)
print(5/2)
print(5//2) ## floor division
print(5%2) ## modulo
print(5**2) ## pow(5,2)
```

```
-3
18
2.5
2
1
25
```

In [13]:
```
## variable

x = 100
y = 50
print(x+y)
```

```
150
```

In [15]:
```
my_university_name = "TBS"
print(my_university_name)
del my_university_name ## delete variable
```

```
TBS
```

In [24]:
```
x,y,z = 1,2,3
print(x,y,z)
```

```
1 2 3
```

In [ ]:
```
## R for small data, prototyping
## Python for larger data, software, data science, ai, application
## R is statistical langauge vs. Python is general langauge
```

In [26]:
```
my_name = "euro"
age = 25
friends = ["fluk", "bird", "Taung"] ## list: you can update data(mutable)
fav_food = ("hotdog", "sushi", "sandwich") ## tuple: you can't(immutable)
```

In [27]:
```
friends[0]
```

Out[27]:
```
'fluk'
```

In [30]:
```
friends[0] = "Suparit"
friends
```

Out[30]:
```
['Suparit', 'bird', 'Taung']
```

In [31]:
```
fav_food[0] = "chicken" ##can't not edit data inside tuple or immutable
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-31-211dc43aa6df> in <module>
----> 1 fav_food[0] = "chicken"

TypeError: 'tuple' object does not support item assignment
```

In [32]:
```
## data types
## int, float, str, bool

age = 25 #int
gpa = 3.03 #float
name = "euro" #str
type(gpa)
```

Out[32]:
```
float
```

In [36]:
```
## convert type
str(age)
```

Out[36]:
```
'25'
```

In [42]:
```
print(type(age))
print(type(str(age)))
```

```
<class 'int'>
<class 'str'>
```

Working with string

In [47]:
```
## fstring aka format string

name = "euro"
gpa = 3.03
```

```
text = f"{name} graduated from tbs with gpa {gpa}."
print(text)
```

```
euro graduated from tbs with gpa 3.03.
```

In [50]:

```
## long string

long_str = """
I abcdefght
abcde
abcde
"""
print(long_str)
```

```
I abcdefght
abcde
abcde
```

In [69]:

```
## function vs method

text = "a duck walks into a bar"
print(len(text))
print(text)
```

```
23
a duck walks into a bar
```

In [70]:

```
## method is a function created specificially to an object

## string method

print(text.upper())
print(text.replace("duck", "lion"))
print(text.count("d"))
print(text.split(" "))
```

```
A DUCK WALKS INTO A BAR
a lion walks into a bar
1
['a', 'duck', 'walks', 'into', 'a', 'bar']
```

In [71]:

```
## replace new value

text = text.replace("duck","lion")
print(text)
print(text.count("d"))
```

```
a lion walks into a bar
0
```

In [72]:

```
result = text.split(" ")
print(result)
print(" ".join(result))
```

```
['a', 'lion', 'walks', 'into', 'a', 'bar']
a lion walks into a bar
```

In [87]:
```
## index start from 0

text = "python"
print(len(text))
print(text[0:2])
print(text[0:5:2])
```

```
6
py
pto
```

In [83]:
```
## text + text

"Python" + " is shit" + " wahaha"*3
```

Out[83]:
```
'Python is shit wahaha wahaha wahaha'
```

In [88]:
```
## string is immutable

text = "python"
text[0] = "c"
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-88-4553ce5935f7> in <module>
      2
      3 text = "python"
----> 4 text[0] = "c"

TypeError: 'str' object does not support item assignment
```

In [89]:
```
print("c"+ text[1:])
```

```
cython
```

data structure

1. list
2. tuple
3. dictionary
4. set

In [150…
```
## list, order, mutable object

shopping_list = ["egg","milk","chocolate","yoghurt"]

shopping_list[0] = "cabbage"
print(shopping_list)
```

```
['cabbage', 'milk', 'chocolate', 'yoghurt']
```

In [151…
```
## list method

shopping_list.append("cheese") ##fill the last one
print(shopping_list)
```
```
['cabbage', 'milk', 'chocolate', 'yoghurt', 'cheese']
```

In [152…
```
shopping_list.remove("yoghurt")
print(shopping_list)
```
```
['cabbage', 'milk', 'chocolate', 'cheese']
```

In [153…
```
shopping_list.pop() ##del the last positon in the list
print(shopping_list)
```
```
['cabbage', 'milk', 'chocolate']
```

In [154…
```
shopping_list.count("cabbage")
```

Out[154…
```
1
```

In [155…
```
shopping_list.sort(reverse=True) #sort z to a
print(shopping_list)
```
```
['milk', 'chocolate', 'cabbage']
```

In [156…
```
shopping_list.insert(1,"grapes")
print(shopping_list)
```
```
['milk', 'grapes', 'chocolate', 'cabbage']
```

In [157…
```
## list + list

["item1"] + ["item2","item3"]
```

Out[157…
```
['item1', 'item2', 'item3']
```

In [159…
```
## loop through shopping list

for item in shopping_list:
    print(item)
```
```
milk
grapes
chocolate
cabbage
```

In [162…
```
for item in shopping_list:
    print("I want to buy "+item)
```
```
I want to buy milk
I want to buy grapes
```

```
I want to buy chocolate
I want to buy cabbage
```

In [163…

```
for item in shopping_list:
    print(item.upper())
```

```
MILK
GRAPES
CHOCOLATE
CABBAGE
```

In [171…

```
for item in shopping_list:
    if len(item) >= 8:
        continue
    else:
        print("I need to buy "+item)
```

```
I need to buy milk
I need to buy grapes
I need to buy cabbage
```

In [172…

```
## average revenue per user

spending = [500,1200,800,300]

for spend in spending:
    if spend >=900:
        print("high spender")
    else:
        print("low spender")
```

```
low spender
high spender
low spender
low spender
```

In [173…

```
scores = [80,95,85,65,48,78]

for score in scores:
    if score >=80:
        print(score,"passed")
    else:
        print(score, "failed")
```

```
80 passed
95 passed
85 passed
65 failed
48 failed
78 failed
```

In [176…

```
## list comprehension

scores = [80,95,85,65,48,78]

grades = [score+5
          for score in scores]
print(grades)
```

```
[85, 100, 90, 70, 53, 83]
```

In [178…
```
grades = ["P" if score >=80 else "F"
          for score in scores]
print(grades)
```

```
['P', 'P', 'P', 'F', 'F', 'F']
```

In [179…
```
## tuple, order, immutable

toy, jane, ann = 35,45,36
print(toy, jane, ann)
```

```
35 45 36
```

In [185…
```
names = ("euro","fluk","bird") ##cannot change, immutable
names[0] = "Tuang"
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-185-4a401b78ae7e> in <module>
      1 names = ("euro","fluk","bird") ##cannot change, immutable
----> 2 names[0] = "Tuang"

TypeError: 'tuple' object does not support item assignment
```

In [186…
```
print(names.count("euro"))
print(names.index("bird"))
```

```
1
2
```

In [189…
```
for name in names:
    print(f"hello! {name.capitalize()}")
```

```
hello! Euro
hello! Fluk
hello! Bird
```

In [191…
```
## recap list

complex_list = [
    25, "Naruto",
    [1,2,3,4,5],
    ("hello","cava","moshi")
]
```

In [193…
```
complex_list
```

Out[193…
```
[25, 'Naruto', [1, 2, 3, 4, 5], ('hello', 'cava', 'moshi')]
```

In [194…
```
complex_list[3][1]
```

Out[194…
```
'cava'
```

In [195…
```
## dictionary; key-vlue pair

movie = {
    "title": "Inception",
    "release_year": 2010,
    "genre": ["Action", "Adventure", "Sci-Fi"],
    "director": "Christopher Nolan"
}
```

In [196…
```
customer_01 = {
    "name": "Naruto",
    "age" : 25,
    "fav_movies" : ["AA","BB","XYZ"],
    "gpa" : 3.03
}

customer_01
```

Out[196…
```
{'name': 'Naruto', 'age': 25, 'fav_movies': ['AA', 'BB', 'XYZ'], 'gpa': 3.03}
```

In [197…
```
##dictionary is unordered, mutable
customer_01[1] ##จะดึงให้ดึง key แทน
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-197-d83d4d4433d0> in <module>
      1 ##dictionary is unordered, mutable
----> 2 customer_01[1]

KeyError: 1
```

In [198…
```
customer_01["age"]
```

Out[198…
```
25
```

In [199…
```
## dictionary method

customer_01.keys()
```

Out[199…
```
dict_keys(['name', 'age', 'fav_movies', 'gpa'])
```

In [200…
```
list(customer_01.keys())
```

Out[200…
```
['name', 'age', 'fav_movies', 'gpa']
```

In [203…
```
print(customer_01.values())
print(customer_01.items()) ## เป็นคู่ๆ
```

```
dict_values(['Naruto', 25, ['AA', 'BB', 'XYZ'], 3.03])
dict_items([('name', 'Naruto'), ('age', 25), ('fav_movies', ['AA', 'BB', 'XYZ']), ('gp
a', 3.03)])
```

In [206…

```
## create new key

customer = customer_01
customer["city"] = "Bangkok"
customer["nationality"] = "Thai"

print(customer)
```

```
{'name': 'Naruto', 'age': 25, 'fav_movies': ['AA', 'BB', 'XYZ'], 'gpa': 3.03, 'city': 'B
angkok', 'nationality': 'Thai'}
```

In [233…

```
## remove gpa key
del customer["gpa"]

## use method
customer.pop("city")
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-233-790f9070c20b> in <module>
      1 ## remove gpa key
----> 2 del customer["gpa"]
      3
      4 ## use method
      5 customer.pop("city")

KeyError: 'gpa'
```

In [221…

```
customer
```

Out[221…

```
{'name': 'Naruto',
 'age': 25,
 'fav_movies': ['AA', 'BB', 'XYZ'],
 'nationality': 'Thai'}
```

In [234…

```
## update value

customer["name"] : "Michael"
```

In [235…

```
customer ## ค่อยกลับมาเช็ค
```

Out[235…

```
{'name': 'Naruto',
 'age': 25,
 'fav_movies': ['AA', 'BB', 'XYZ'],
 'nationality': 'Thai'}
```

In [236…

```
## set is for find distinct or unique value

set([1,1,2,3,4])
```

Out[236…

```
{1, 2, 3, 4}
```

In [237…
```
##set operation; union and intersection

mary = {"orange","apple"}
euro = {"apple","mango"}

mary & euro
```

Out[237…
```
{'apple'}
```

Recap data structure

1. list
2. tuple
3. dictionary
4. set

Function

users defined fucntion.

it is reuseable

In [239…
```
def hello():
    print("hello world")

hello()
```

```
hello world
```

In [245…
```
##default argument

def hello2(name="euro"):
    print("hello " + name)

hello2()
hello2("Fluk")
```

```
hello euro
hello Fluk
```

In [249…
```
def greeting():
    username = input("What's your name: ")
    result= f"Hi {username}!"
    print(result)
    action = input("What are you going to do: ")
    print(f"You're going to {action}")
```

In [250…
```
greeting()
```

```
What's your name: Naphon
Hi Naphon!
```

```
What are you going to do: having lunch
You're going to having lunch
```

In [253…
```
input("How old are you: ")
```

```
How old are you: 25
```

Out[253…
```
'25'
```

In [255…
```
user_age = int(input("How old are you: "))
```

```
How old are you: 25
```

In [259…
```
## function is able to have more than one parameter

def my_power(base=2, power=3):
    return(base**power)
```

In [262…
```
result = my_power()
print(result)
```

```
8
```

In [264…
```
result = my_power(power=5)
print(result)
```

```
32
```

Control flow

1. if
2. for
3. while

In [267…
```
## regular function

def double(num):
    return num*2

double(10)
```

Out[267…
```
20
```

In [269…
```
##lamda function

double = lambda num : num*2

double(10)
```

Out[269…
```
20
```

In [275…
```
hello = lambda: print("hello world!")
hello()
```

```
hello world!
```

In [280…
```
hello = lambda name: print("hello, name")
hello(name="euro") ##ค่อยกลับมาดู งงไอสัด
```

```
hello, name
```

In [285…
```
def grading(score):

    """
    input: score is a numeric number
    output: grade passed or failed
   """

    if score >=80:
        return "Passed"
    else:
        return "failed"

grading(85)
```

Out[285… 'Passed'

In [292…
```
## multiple if else

def full_grading(score):
    if score >= 80:
        return "A"
    elif score >= 70:
        return "B"
    elif score >= 60:
        return "C"
    else:
        return "F"

print(full_grading(62))
print(full_grading(59))
print(full_grading(75))
print(full_grading(92))
```

```
C
F
B
A
```

In [298…
```
## if multiple conditions
time = "morning"
day = "weekday"

if time == "morning"and day == "weekday":
    print("I am eating cereal")
elif time == "morning" and day == "weekend":
    print("I am eating hamburger")
```

```
else:
    print("No eating")
```

```
I am eating cereal
```

In [299…

```
## recap for

for item in [ "grapes","orange","banana"]:
    print(item.upper())
```

```
GRAPES
ORANGE
BANANA
```

In [305…

```
for item in [ "grapes","orange","asian banana"]:
    if(len(item) <=6):
        print(item.upper())
```

```
GRAPES
ORANGE
```

In [308…

```
## while loop

count = 0
while count < 5:
    print("hello world")
    count = count+1
```

```
hello world
hello world
hello world
hello world
hello world
```

HW

1. review methods (list.string)
2. pao ying chub

In [311…

```
import random

def pao_ying_chub():
    print("let's play game")
    hands = ["hammer", "scissors", "paper"]
    a_hand = random.choice(hands)
    print(a_hand)

pao_ying_chub()
```

```
let's play game
paper
```