```
1
```

Essential Python 101

- varibles
- data types
- data studctures
- function
- contro flow
- OOP

```
1 print ("hello wolrd")
2
```

```
hello wolrd
```

```
1 print ("I am learning python")
```

```
I am learning python
```

```
1 # This is just a note
2 print (2+2)
3 print (5*5)
```

```
4
25
```

```
1 # basic calculation
2 1+1
3 2*2
4 print (7/2)
5 print (7//2) #floor division
```

```
3.5
3
```

```
1 pow(5,2)
```

```
25
```

```
1 abs(-666)
```

```
666
```

```
1 # modulo
2 5 % 3
```

```
2
```

```
1 # 5 building blocks
2 # 1.varibles
3 # 2. data types
4 # 3. data studctures
5 # 4. function
6 # 5. contro flow
7 # 6. OOP
```

```
1 # assign a new variavle
2 my_name = "lie"
3 age = 30
4 gpa = 3.14
5 movie_lover = True # False
```

```
1 my_name
```

```
'lie'
```

```
1 #over write a value
2 age = 30
```

```
3 new_age = age - 12
4 print(age, new_age)
```

```
    30 18
```

```
1 s23_price = 30000
2 discount = 0.15
3 new_s23_price = s23_price *(1 -discount)
4 print(new_s23_price)
```

```
    25500.0
```

```
1 # remove variable
2 del new_s23_price
```

```
1  #count variable
2  age = 34
3  age += 1
4  age -= 2
5  age *= 2
6  print(age)
7
```

```
    66
```

```
1 # data types
2 # int float str bool
```

```
1 age = 30
2 gpa = 3.14
3 school = "Swu"
4 movie_lover = True
```

```
1 # check data types
2 print( type(age) )
3 print( type(gpa) )
4 print( type(school) )
5 print( type(movie_lover) )
```

```
    <class 'int'>
    <class 'float'>
    <class 'str'>
    <class 'bool'>
```

```
1 # convert type
2 x = 100
3 x = str(x)
4 print(x, type(x))
```

```
    100 <class 'str'>
```

```
1 y = False # T=1, F=0
2 y = int(y)
3 print(y,type(y))
```

```
    0 <class 'int'>
```

```
1 z = 1
2 z = bool(z)
3 print(z,type(z))
```

```
    True <class 'bool'>
```

```
1 age = 34
2 print(age+age, age*2, age/2)
```

```
    68 68 17.0
```

```
1 text = "I'm learning Python"
2 text2 = ' "hahaha" '
3 print(text, text2)
```

```
    I'm learning Python  "hahaha"
```

```
1 text ="hello"
2 print(text*4)
```

```
hellohellohellohello
```

```
1 #type hint
2 age: int = 30
3 my_name: str = "lie"
```

```
1 #greeting
2 def greeting(name = "Lie", location="Bkk"):
3     print("Hello "+name)
4     print("She is in " +location)
```

```
1 greeting(location = "Japan",
2          name = "Lie")
```

```
Hello Lie
She is in Japan
```

```
1 def add_two_nums(num1, num2):
2     print("Hello world")
3     print("Done!")
4     print("Yes")
5     return num1 + num2
```

```
1 result = add_two_nums(5,10)
2 print(result)
```

```
Hello world
Done!
Yes
15
```

```
1 def add_two_nums(a: int, b: int) ->int:
2     return a+b
```

```
1 add_two_nums(5,6)
```

```
11
```

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

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

```
1 # slicing, index starts with 0
2 text[-1]
```

```
'r'
```

```
1 text
```

```
'a duck walks into a bar'
```

```
1 text[7:]
```

```
'walks into a bar'
```

```
1 # String is immutable
2 name = "Python" # -> Cython
3 name = "C" + name[1:]
4 print(name)
```

```
Cython
```

```
1 text = "a duck walks into a bar"
```

```
1 text.upper()
```

```
'A DUCK WALKS INTO A BAR'
```

```
1 text.title()
```

```
    'A Duck Walks Into A Bar'
```

```
text = text.lower()
text
```

```
    'a duck walks into a bar'
```

```
words = text.split(" ")
```

```
" ". join(words)
```

```
    'a duck walks into a bar'
```

```
"-". join(words)
```

```
    'a-duck-walks-into-a-bar'
```

```
#data structures
# list []
# tuple()
# dictionary{}
# srt {unique}
```

```
shopping_items = ["banana", "egg", "milk"]
shopping_items[0] = "pineapple"
print(shopping_items)
```

```
    ['pineapple', 'egg', 'milk']
```

```
shopping_items.append("tomato")
print(shopping_items)
```

```
    ['pineapple', 'egg', 'milk', 'tomato']
```

```
#sort items
shopping_items.sort(reverse = True)
print(shopping_items)
```

```
    ['tomato', 'pineapple', 'milk', 'egg']
```

```
def mean(scores):
    return sum(scores) / len(scores)
```

```
scores = [90,88,85,92,75]
print(len(scores),sum(scores),
      min(scores), max(scores), mean(scores))
```

```
    5 430 75 92 86.0
```

```
shopping_items.pop()
shopping_items
```

```
    ['tomato', 'pineapple', 'milk']
```

```
shopping_items.remove("milk")
shopping_items
```

```
    ['tomato', 'pineapple']
```

```
shopping_items.append("coke")
shopping_items
```

```
    ['tomato', 'pineapple', 'coke']
```

```
shopping_items.append("coffee")
shopping_items
```

```
    ['tomato', 'pineapple', 'coke', 'coffee']
```

```
#.insert()
shopping_items.insert(1,"orange")
```

```
1 shopping_items
```

```
['tomato', 'orange', 'pineapple', 'coke', 'coffee']
```

```
1 # list + list
2 items1 = ['egg', 'milk']
3 items2 = ['banana', 'bread']
4 print(items1+items2)
```

```
['egg', 'milk', 'banana', 'bread']
```

```
1 #tuple () is immutable
2 tup_items = ('egg', 'bread','coke','coke','coke')
3 tup_items
```

```
('egg', 'bread', 'coke', 'coke', 'coke')
```

```
1 tup_items.count('coke')
```

```
3
```

```
1 # username password
2 # student1, student2
3 s1 = ("id001", "123456")
4 s2 = ("id002", "654321")
5 user_pw = (s1, s2)
6
7 print(user_pw)
```

```
(('id001', '123456'), ('id002', '654321'))
```

```
1 # tuple unpacking
2 username, password = s1
3 s1
```

```
('id001', '123456')
```

```
1 # tuple unpacking
2 name, age, _ = ("lie",30,3.41)
3 print(name,age)
```

```
lie 30
```

```
1 #Set {unique}
2 courses = ["Python", "Python", "R", "SQL","SQL","sql"]
```

```
1 set(courses)
```

```
{'Python', 'R', 'SQL', 'sql'}
```

```
1 #contro flow
2 #if for while
3
```

```
1 # final exam 150 quesetions, pass >=120
2 def grade(score):
3     if score >= 120:
4         return "Excellent"
5     elif score >= 100:
6         return "Good"
7     elif score >= 80:
8         return "Okay"
9     else:
10        return "Need to read more"
11
```

```
1 result = grade(95)
2 print(result)
```

```
Okay
```

```
1 #use and, or in condition
2 # course == data science, score >=80 passed
3 # course == english, score >=70 passed
4 def grade(course, score):
```

```
    if course == "english" and score >= 70:
        return "passed"
    elif course == "data science" and score >= 80:
        return "passed"
    else:
        return "falled"
```

```
grade("english",70)
```

```
'passed'
```

```
# for loop
# if score >= 80, passed
def grading_all(scores):
    new_scores = []
    for score in scores:
        new_scores.append(score+2)
    return new_scores

```

```
grading_all([75,88,90,95,52])
```

```
[77, 90, 92, 97, 54]
```

```
# list comprehension
scores = [77, 90, 92, 97, 54]
```

```
new_scores = [s*2 for s in scores]
new_scores

```

```
[154, 180, 184, 194, 108]
```

```
friends = ["pink","blue","eye","book"]
[f.upper() for f in friends]


```

```
['PINK', 'BLUE', 'EYE', 'BOOK']
```

```
#While loop
count = 0

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

```
hello
hello
hello
hello
hello
```

```
# chatbot for fruit oeder
user_name = input("What is your name?")
```

```
What is your name?Wallow
```

```
def chatbot():
    fruits = []
    while True:
        fruit = input("What fruit do you want to order?")
        fruits.append(fruit)
        if fruit == "exit":
            return fruits

```

```
chatbot()
```

```
What fruit do you want to order?apple
What fruit do you want to order?grape
What fruit do you want to order?coconut
What fruit do you want to order?lemon
What fruit do you want to order?exit
['apple', 'grape', 'coconut', 'lemon', 'exit']
```

```
age = int(input("how old are you"))
```

```
how old are you30
```

```
type(age)
```

```
int
```

```
#OOP - Object oriented programming
# Dog class
```

```
class Dog:
    def __init__(self, name, age, breed):
        self.name = name
        self.age = age
        self.breed = breed
```

```
dog1 = Dog("buay", 2, "chivava")
dog2 = Dog("pepo", 4 , "bulldog")
dog3 = Dog("pepsi", 5, "skt")
```

```
print(dog1.name, dog1.age, dog1.breed)
```

```
buay 2 chivava
```

```
class Employee:
    def __init__(self, id, name, dept, pos):
        self.id = id
        self.name = name
        self.dept = dept
        self.pos = pos
    def hello(self):
        print(f"Hello! my name is {self.name}")
    def work_hours(self, hours):
        print(f"{self.name} works for {hours} hours.")
    def change_dept(self, new_dept):
        self.dept = new_dept
        print(f"{self.name} is now in {self.dept}.")
```

```
emp1 = Employee(1, "John", "Finance", "Financial Analyst")
```

```
print(emp1.name,emp1.dept)
```

```
John Finance
```

```
emp1.hello()
```

```
Hello! my name is John
```

```
emp1.work_hours(10)
```

```
John works for 10 hours.
```

```
emp1.dept
```

```
'Finance'
```

```
emp1.change_dept("Data Science")
```

```
John is now in Data Science.
```

```
#Object: attribute => name, id, dept, pos  #ตัวแปร คุณลักษณะ
#Object: method => hello(), change_dept
```