More than 100 Python exercises have solutions (sample code)

More than 100 Python code examples are shared by guy zhiwehu on Github, however, the solution of this series is written on the old Python version. Following TipsMake.com will be Vietnameseized and edited to suit Python 3.x to help you learn and practice Python.

More than 100 Python code examples are shared by guy zhiwehu on Github, however, the sample code of this series is written on the old Python version. Following TipsMake.com will be Vietnameseized and edited to suit Python 3.x to help you learn and practice Python.

Note: The sample code in the article is written in Python 3.6.2, if you are using Python version 2.5 or below it may not run the code because in the new Python version many commands, the function has been changed.

This Python exercise will be divided into 3 levels for beginners to learn programming, who have learned programming but who have just learned Python and who want to improve Python proficiency. Each exercise has a full set of requirements, lesson suggestions, and solution (sample code), which is a sample Python code for your reference. Now invite you to go into the specific content.

Practical Python exercises with sample code

1. 1. Describe the Python level
2. 2. Python exercise structure
3. 3. Python level 1 exercises
4. 4. Python exercises level 2
5. 5. Python level 3 exercises
6. 6. Other Python exercises

1. Describe the Python level

Level 1: The person who has just taken an overview of Python, can solve some problems with 1, 2 classes or Python functions. Exercises of this level can be found in regular textbooks and instructional materials.

Level 2: New learner of Python but already has a relatively powerful programming platform that can solve problems related to 3 classes or Python functions. These exercises are often not found in textbooks.

Level 3: Enhance, use Python to solve more complex problems by using rich functions, data structures and algorithms. At this level you can solve problems using several standard Python packages and advanced

programming techniques.

2. Python exercise structure

Each Python exercise in this page will include the following 3 sections:

1. Question.
2. Suggestions.
3. Sample code.

I will leave the original form like this, you can see the questions and suggestions then practice yourself before pulling down to see the sample code.

3. Python level 1 exercises

Lesson 01:

Question:

Write a program to find all numbers divisible by 7 but not multiples of 5, located in paragraphs 2000 and 3200 (including 2000 and 3200). The obtained numbers will be printed in series on one line, separated by commas.

Suggestions:

1. Use range (#begin, #end)

Sample code:

```
j=[]
for i in range(2000, 3201):
 if (i%7==0) and (i%5!=0):
 j.append(str(i))
print (','.join(j))
```

Lesson 2:

Question:

Writing a program can calculate the factorial of a given number. The result is printed in series on a line, separated by commas. For example, the given number is 8, the output must be 40320.

Suggestions:

1. In case the input data is provided, please choose a way for the user to enter the number.

Sample code:

```
x=int(input("Nh?p s? c?n tính giai th?a:"))
def fact(x):
 if x == 0:
```

```
  return 1
  return x * fact(x - 1)
 print (fact(x))
```

Lesson 03:

Question:

For a given integer n, write a program to create a dictionary containing (i, i * i) as an integer between 1 and n (including 1 and n) and then print this dictionary. Example: Suppose the number n is 8, the output will be: {1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64}.

Suggestions:

1. Write a command that requires an integer n.

Sample code:

```
 n=int(input("Nh?p vào m?t s?:"))
 d=dict()
 for i in range(1,n+1):
  d[i]=i*i

 print (d)
```

Lesson 04:

Question:

Write a program that accepts a sequence of numbers, separated by commas from the console, creates a list and a tuple containing all numbers.

Example: The input provided is 34,67,55,33,12,98, the output is:

['34', '67', '55', '33', '12', '98']
('34', '67', '55', '33', '12', '98')

Suggestions:

1. Write a command to enter the values ??and then use the data type conversion rule to complete.

Sample code:

```
 values=input("Nh?p vào các giá tr?:")
 l=values.split(",")
 t=tuple(l)
 print (l)
 print (t)
```

Lesson 05:

Question:

Define a class with at least 2 methods:

getString: to receive a string entered by the user from the console.

printString: print the string just entered in uppercase.

Add simple risk functions to test the method of the class.

Example: The input string is quantrimang.com, the output must be: QUANTRIMANG.COM

Suggestions:

1. Use __init__ to build parameters.

Sample code:

```
class InputOutString(object):
 def __init__(self):
 self.s = ""

 def getString(self):
 self.s = input("Nh?p chu?i:")
 # Code by TipsMake.com
 def printString(self):
 print (self.s.upper())

strObj = InputOutString()
strObj.getString()
strObj.printString()
```

Lesson 06:

Question:

Write a method that calculates the square value of a number.

Suggestions:

1. Use operator **.

Sample code:

```
x=int(input("Nh?p m?t s?:")) #nh?p s? c?n tính bình ph??ng t? giao di?n
def square(num):  #??nh ngh?a bình ph??ng c?a m?t s?
 return num ** 2
 # Code by TipsMake.com
print (square(2)) #in bình ph??ng c?a 2
print (square(3)) #in bình ph??ng c?a 3
print (square(x)) #in bình ph??ng c?a x
```

Since the problem does not specifically require you to calculate the number of squares available or the number entered, I use both.

Lesson 07:

Question:

Python has many built-in functions, if you don't know how to use it, you can read the documentation online or find some books. But Python also has function documentation available for every Python built-in function. The requirement of this exercise is to write a program to print documents on some of the built-in Python functions such as abs (), int (), input () and add documents for the function you define yourself.

Suggestions:

1. Use __doc__

Sample code:

```
print (abs.__doc__)
print (int.__doc__)
print (input.__doc__)
 # Code by TipsMake.com
def square(num):
 '''Tr? l?i giá tr? bình ph??ng c?a s? ???c nh?p vào.

 S? nh?p vào ph?i là s? nguyên.
 '''
 return num ** 2

print (square.__doc__)
```

Lesson 08:

Question:

Defining a class includes class parameters and has the same instance parameter

Suggestions:

1. When defining an instance parameter, it needs to be added to __init__
2. You can initialize an object with the start parameter or set the value later.

Sample code:

```
class Person:
 # ??nh ngh?a l?p "name"
 name = "Person"
 # Code by TipsMake.com
 def __init__(self, name = None):
 # self.name là bi?n instance
 self.name = name

jeffrey = Person("Jeffrey")
print ("%s name is %s" % (Person.name, jeffrey.name))

nico = Person()
```

```
nico.name = "Nico"
print ("%s name is %s" % (Person.name, nico.name))
```

4. Python exercises level 2

Lesson 09:

Question:

Write a program and print the value according to the given formula: Q = ? ([(2 * C * D) / H]) (in words: Q equals the square root of [(2 kernels C kernel D) divided H] With a fixed value of C of 50, H is 30. D is a range of custom values, entered from the user interface, the values ??of D are separated by commas.

Example: Assuming the value chain of D entered is 100,150,180, the output will be 18,22,24.

Suggestions:

1. If the output received is a decimal number, you need to round it to the nearest value, for example 26.0 will be printed as 26.
2. In case the input data is provided for the question, it is assumed that the input is entered by the user from the console.

Sample code:

```
#!/usr/bin/env python
import math
c=50
h=30
value = []
items=[x for x in input("Nh?p giá tr? c?a d: ").split(',')]
for d in items:
 value.append(str(int(round(math.sqrt(2*c*float(d)/h)))))
 # Code by TipsMake.com
print (','.join(value))
```

Lesson 10:

Question:

Write a 2-digit, X, Y program that gets the value from the input and creates a 2-dimensional array. The element value in the ith row and jth column of the array must be i * j.

Note: i = 0.1, ., X-1; j = 0.1, ., Y-1.

Example: Input X and Y value is 3.5 then output is: [[0, 0, 0, 0, 0], [0, 1, 2, 3, 4], [0, 2, 4 , 6, 8]]

Suggestions:

1. Write a command to get X, Y values ??from the user-input console.

Sample code:

```
input_str = input("Nh?p X, Y: ")
dimensions=[int(x) for x in input_str.split(',')]
rowNum=dimensions[0]
colNum=dimensions[1]
multilist = [[0 for col in range(colNum)] for row in range(rowNum)]
 # Code by TipsMake.com
for row in range(rowNum):
 for col in range(colNum):
 multilist[row][col]= row*col

print (multilist)
```

Lesson 11:

Question:

Write a program that accepts a string of words entered by users, separated by commas, and prints those words in sequence in alphabetical order, separated by commas.

Suppose the input is: without, hello, bag, world, then the output will be: bag, hello, without, world.

Suggestions:

In case the input data is entered into the program it should be assumed that the data is entered by the user from the console.

Sample code:

```
items=[x for x in input("Nh?p m?t chu?i: ").split(',')]
items.sort()
print (','.join(items))
```

Lesson 12:

Question:

Write a program that accepts strings as input lines, converts these lines into capital letters and prints them to the screen. Suppose the input is:

Hello world
Practice c?n Perfect

The output will be:

HELLO WORLD
PRACTICE MAKES PERFECT

Suggestions:

In case the input data is entered into the program it should be assumed that the data is entered by the user from the console.

Sample code:

```
lines = []
while True:
 s = input()
 if s:
 lines.append(s.upper())
 else:
 break;
 # Lesson Python 12, Code by TipsMake.com
for sentence in lines:
 print (sentence)
```

Lesson 13:

Question:

Write a program that accepts input as a string of words separated by spaces, removing duplicate words, alphabetically ordering, and printing them.

Suppose the input is: hello world and practice perfect and hello world again

The output is: again and hello makes perfect practice world

Suggestions:

1. In case the input data is entered into the program it should be assumed that the data is entered by the user from the console.
2. Use set to remove automatic duplicate data and use sorted () to sort data.

Sample code:

```
s = input("Nh?p chu?i c?a b?n: ")
words = [word for word in s.split(" ")]
print (" ".join(sorted(list(set(words)))))
```

Lesson 14:

Question:

Write a program that accepts input as a series of 4-digit binary numbers, separated by commas, to check if they are divisible by 5. Then print the divisors by 5 into a sequence separated by commas.

The input example is: 0100,0011,1010,1001

The output will be: 1010

Suggestions:

In case the input data is entered into the program it should be assumed that the data is entered by the user from the console.

Sample code:

```
value = []
items=[x for x in input("Nh?p các s? nh? phân: ").split(',')]
for p in items:
 intp = int(p, 2)
 if not intp%5:
 value.append(p)
 # Exercise Python 14, Code by TipsMake.com
print (','.join(value))
```

Lesson 15:

Question:

Write a program to find all the numbers in paragraphs 1000 and 3000 (including both numbers) so that all the digits in the number are even numbers. Print the numbers found in a string separated by commas, on a line.

Suggestions:

In case the input data is entered into the program it should be assumed that the data is entered by the user from the console.

Sample code:

```
values = []
for i in range(1000, 3001):
 s = str(i)

if (int(s[0])%2==0) and (int(s[1])%2==0) and (int(s[2])%2==0) and (int(s[3])%2==

 values.append(s)
# Bài t?p Python 15, Code by TipsMake.com
print (",".join(values))
```

Lesson 16:

Question:

Write a program that accepts input as a sentence, counting the number of letters and numbers in that sentence. Suppose the following input is provided for the program: hello world! 123

The output will be:

Number of letters is: 10
Number of digits is: 3

Suggestions:

In case the input data is entered into the program it should be assumed that the data is entered by the user from the console.

Sample code:

```
s = input("Nh?p câu c?a b?n: ")
# Bài t?p Python 16, Code by TipsMake.com
d={"DIGITS":0, "LETTERS":0}
for c in s:
 if c.isdigit():
 d["DIGITS"]+=1
 elif c.isalpha():
 d["LETTERS"]+=1
 else:
 pass
print ("S? ch? cái là:", d["LETTERS"])
print ("S? ch? s? là:", d["DIGITS"])
```

Lesson 17:

Question:

Write a program that accepts input as a sentence, counting uppercase and lowercase letters.

Suppose the input is: TipsMake.com

The output is:

Uppercase: 3

Lowercase: 8

Suggestions:

In case the input data is entered into the program it should be assumed that the data is entered by the user from the console.

Sample code:

```
s = input("Nh?p câu c?a b?n: ")
d={"UPPER CASE":0, "LOWER CASE":0}
# Code by TipsMake.com
for c in s:
  if c.isupper():
    d["UPPER CASE"]+=1
  elif c.islower():
    d["LOWER CASE"]+=1
  else:
    pass
print ("Ch? hoa:", d["UPPER CASE"])
print ("Ch? th??ng:", d["LOWER CASE"])
```

Lesson 18:

Question:

Write a program that calculates the value of a + aa + aaa + aaaa with a being the number entered by the user.

Assuming a input is 1, the output will be: 1234

Suggestions:

In case the input data is entered into the program it should be assumed that the data is entered by the user from the console.

Sample code:

```
a = input("Nh?p s? a: ")
n1 = int( "%s" % a )
n2 = int( "%s%s" % (a,a) )
n3 = int( "%s%s%s" % (a,a,a) )
n4 = int( "%s%s%s%s" % (a,a,a,a) )
# Bài t?p Python 18, Code by TipsMake.com
print ("T?ng c?n tính là: ",n1+n2+n3+n4)
```

Lesson 19:

Question:

Use a list to filter odd numbers from the list entered by the user.

Assuming the input is: 1,2,3,4,5,6,7,8,9, the output must be: 1,3,5,7,9

Suggestions:

In case the input data is entered into the program it should be assumed that the data is entered by the user from the console.

Sample code:

```
values = input("Nh?p dãy s? c?a b?n, cách nhau b?i d?u ph?y: ")
numbers = [x for x in values.split(",") if int(x)%2!=0]
print (",".join(numbers))
```

Lesson 20:

Question:

Write a program that calculates the actual amount of a bank account based on the transaction log entered from the console.

The log format is displayed as follows:

D 100
W 200

(D is deposit, W is withdrawn money).

Suppose the input provided is:

D 300

D 300

W 200

D 100

The output will be:

500

Suggestions:

In case the input data is entered into the program it should be assumed that the data is entered by the user from the console.

Sample code:

```
import sys
netAmount = 0
# Bài t?p Python 20, Code by TipsMake.com
while True:
  s = input("Nh?p nh?t ký giao d?ch: ")
  if not s:
    break
  values = s.split(" ")
  operation = values[0]
  amount = int(values[1])
  if operation=="D":
     netAmount+=amount
  elif operation=="W":
    netAmount-=amount
  else:
    pass
print (netAmount)
```

5. Python level 3 exercises

Lesson 21:

Question:

A website requires users to enter their username and password to register. Write a program to check the validity of the password users enter.

Criteria for checking passwords include:

1. At least 1 letter is in [az]
2. At least 1 number is in [0-9]
3. At least 1 character is in [AZ]
4. At least 1 character is in [$ # @]
5. Minimum password length: 6
6. Maximum password length: 12

The program must accept a comma-separated password string and check if they meet the above criteria. Valid passwords will be printed, each password separated by commas.

For example, the password entered into the program is: ABd1234 @ 1, a F1 #, 2w3E *, 2We3345

The output will be: ABd1234 @ 1

Suggestions:

In case the input data is entered into the program it should be assumed that the data is entered by the user from the console.

Sample code:

```
import re
value = []
items=[x for x in input("Nh?p m?t kh?u: ").split(',')]
# Bài t?p Python 21, Code by TipsMake.com
for p in items:
  if len(p)12:
    continue
  else:
    pass
  if not re.search("[az]",p):
    continue
  elif not re.search("[0-9]",p):
    continue
  elif not re.search("[AZ]",p):
    continue
  elif not re.search("[$#@]",p):
    continue
  elif re.search("s",p):
    continue
  else:
    pass
  value.append(p)
print (",".join(value))
```

Lesson 22:

Question:

Write a program to sort tuple (name, age, score) in ascending order, name is string, age and height are numbers. Tuple is entered by the user. Sorting criteria are:

Sort by name then sorted by age, then sorted by score. Priority is name> age> point.

If the input is:

Tom, 19.80
John, 20.90
Jony, 17.91
Jony, 17.93
Json, 21.85

The output will be:

[('John', '20', '90'), ('Jony', '17', '91'), ('Jony', '17', '93'), ('Json', '21 ',' 85 '), (' Tom ',' 19 ',' 80 ')]

Suggestions:

In case the input data is entered into the program it should be assumed that the data is entered by the user from the console.

Use itemgetter to accept multiple key arrangements.

Sample code:

```
from operator import itemgetter, attrgetter
 # Python Exercises 22 Code by TipsMake.com
l = []
while True:
  s = input()
  if not s:
    break
  l.append(tuple(s.split(",")))

print (sorted(l, key=itemgetter(0,1,2)))
```

Lesson 23:

Question:

Determine whether a class with generator can repeat numbers between 0 and n, and divide it by 7.

Suggestions:

Use yield.

Sample code:

```
def putNumbers(n):
  i = 0
  while i
    j=i
    i=i+1
    if j%7==0:
```

```
      yield j
# Bài t?p Python 23 Code by TipsMake.com
for i in putNumbers (100):
   print (i)  code="">   while i
    j=i
    i=i+1
    if j%7==0:
      yield j
# Bài t?p Python 23 Code by TipsMake.com
for i in putNumbers (100):
   print (i)
```

Lesson 24:

Question:

A robot moves in the plane starting from the first point (0,0). Robots can move in the direction of UP, DOWN, LEFT and RIGHT with certain steps. The moving mark of the robot is shown as follows:

UP 5

DOWN 3

LEFT 3

RIGHT 3

The following numbers behind the direction of movement are the number of steps. Write a program to calculate the distance from the current position to the first position, after the robot has moved a distance. If the distance is a decimal, just print the nearest integer.

For example: If the following tuple is the program input:

UP 5
DOWN 3
LEFT 3
RIGHT 2

then the output will be 2.

Suggestions:

In case the input data is entered into the program it should be assumed that the data is entered by the user from the console.

Sample code:

```
import math
pos = [0,0]
while True:
 s = input()
 if not s:
```

```
  break
  movement = s.split(" ")
  direction = movement[0]
  steps = int(movement[1])
  if direction=="UP":
  pos[0]+=steps
  elif direction=="DOWN":
  pos[0]-=steps
  elif direction=="LEFT":
  pos[1]-=steps
  elif direction=="RIGHT":
  pos[1]+=steps
  else:
  pass
  # Exercise Python 24 Code by TipsMake.com
 print (int(round(math.sqrt(pos[1]**2+pos[0]**2))))
```

Lesson 25:

Question:

Write a program that calculates the frequency of words from input. Output is output after alphabetical arrangement.

Suppose the input is: New to Python or choose between Python 2 and Python 3? Read Python 2 or Python 3.

Then the output must be:

2: 2
3.:1
3?: 1
New: 1
Python: 5
Read: 1
and: 1
between: 1
Choosing: 1
or: 2
Big: 1

Suggestions:

In case the input data is provided for the question, it must be assumed that an input is entered from the console.

Sample code:

```
  freq = {} # frequency of words in text
  line = input ()
  for word in line.split ():
  freq [word] = freq.get (word, 0) +1
  # Exercise Python 25 Code by TipsMake.com
  words = sorted (freq.keys ())
```

```
cho w trong t?:
print ("% s:% d"% (w, freq [w]))
```

6. Other Python exercises

Lesson 26:

Question:

Definition of a function can sum two numbers.

Suggestions:

Defining a function with 2 numbers is an argument. You can calculate the sum in the function and return the value.

Sample code:

```
def SumFunction(number1, number2): #??nh ngh?a hàm tính t?ng
 return number1+number2
print (SumFunction(5,7)) #in t?ng 2 s? 5 và 7
```

Lesson 27:

Question:

Defining a function can convert integers into strings and print it to the console

Suggestions:

Use str () to convert a number into a string.

Sample code:

```
def printValue(n):
 print (str(n))
printValue(3)
```

Lesson 28:

Question:

Function definitions can take two integers in the form of string and calculate their sum, then print the console output.

Suggestions:

Use int () to convert a string to an integer.

Sample code:

```
def printValue(s1,s2):
 print (int(s1)+int(s2))
printValue("3","4") #K?t qu? là 7
```

Lesson 29:

Question:

Function definitions can take two strings from input and connect them then print the console

Suggestions:

Use + to join strings.

Sample code:

```
def printValue(s1,s2):
 print (s1+s2)
printValue("3","4") #K?t qu? là 34
```

Lesson 30:

Question:

Define a function with input of 2 strings and print a string of greater length in the console. If two strings are the same length, print all the strings in line.

Suggestions:

Use the len function () to get the length of a string

Sample code:

```
def printValue(s1,s2):
 # Python 30 Code by TipsMake.com
 len1 = len(s1)
len2 = len(s2)
if len1>len2:
print (s1)
elif len2>len1:
print (s2)
else:
print(s1)
print (s2)
printValue("one","three")
```

Lesson 31:

Question:

The function definition can accept input as an integer and print "This is an even number" if it is even and print "This is an odd number" if it is an odd number.

Suggestions:

Use the% operator to check whether the number is even or odd.

Sample code:

```
def checkValue(n):
 if n%2 == 0:
print ("?ây là m?t s? ch?n")
 else:
 print ("?ây là m?t s? l?")
checkValue(7)
```

Lesson 32:

Question:

Defining a function can be printed in dictionary containing the key numbers 1 to 3 (including both numbers) and their square values.

Suggestions:

1. Use dict [key] = value to import entries into dictionary.
2. Use the word ** to get the square of a number.

Sample code:

```
def printDict():
d=dict()
d[1]=1
d[2]=2**2
d[3]=3**2
print (d)
# Bài t?p Python 32, Code by TipsMake.com
printDict()
```

Running the above code will result in a dictionary as follows: {1: 1, 2: 4, 3: 9}. If you don't understand this dictionary data type well, please read it again: Python data type: string, number, list, tuple, set and dictionary

Lesson 33:

Question:

Defining a function can print dictionary containing the keys from 1 to 20 (including 1 and 20) and their square values.

Suggestions:

1. Use dict [key] = value to import entries into dictionary.
2. Use the word ** to get the square of a number.
3. Use dujnng range () for loops.

Sample code:

```
def printDict():
 d=dict()
 for i in range(1,21):
 d[i]=i**2
 print (d)
# Bài t?p Python 33, Code by TipsMake.com
printDict()
```

The results when running the above code are: {1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100, 11 : 121, 12: 144, 13: 169, 14: 196, 15: 225, 16: 256, 17: 289, 18: 324, 19: 361, 20: 400}

Lesson 34 :

Question:

Define a function that can create a dictionary, containing the key numbers from 1 to 20 (including 1 and 20) and their square values. The function only prints values.

Suggestions:

1. Use dict [key] = value to import entries into dictionary.
2. Use the word ** to get the square of a number.
3. Use range () for loops.
4. Use keys () to iterate the keys in the dictionary. You can use item () to get key / value pairs.

Sample code:

```
def printDict():
 d=dict()
 for i in range(1,21):
 d[i]=i**2
for (k,v) in d.items():
 print (v)
# Bài t?p Python 34, Code by TipsMake.com
printDict()
```

The results you get when running the above code are the square values ??of numbers from 1 to 20.

Lesson 35:

Question:

Defining a function can create a dictionary containing the key numbers 1 through 20 (including 1 and 20) and the square values ??of the key. The function only needs to print the keys.

Suggestions:

Similar to lesson 34.

Sample code:

```
def printDict():
d=dict()
for i in range(1,21):
d[i]=i**2
for k in d.keys():
print (k)
# Bài Python 35, Code by TipsMake.com
printDict()
```

Running the above code you will get the keys in the dictionary, which are numbers from 1 to 20.

Lesson 36:

Question:

Defining a function can create and print list containing square values ??of numbers from 1 to 20 (including 1 and 20).

Suggestions:

1. Use the ** operator to get the square value.
2. Use range () for the loop.
3. Use list.append () to add values ??to the list.

Sample code:

```
def printList ():
li = list ()
for i in range (1.21):
li.append (i ** 2)
print (li)
# Lesson Python 36, Code by TipsMake.com
printList ()
```

Running the above code you will get a list containing the square values ??of the numbers from 1 to 20.

Lesson 37:

Question:

Defining a function can create lists containing square values ??of numbers from 1 to 20 (including 1 and 20) and print the first 5 items in the list.

Suggestions:

1. Use the ** operator to get the square value.
2. Use range () for the loop.
3. Use list.append () to add values ??to the list.
4. Use [n1: n2] to cut the list

Sample code:

```
def printList():
 li=list()
 for i in range(1,21):
 li.append(i**2)
print (li[:5])
# Bài Python 37, Code by TipsMake.com
printList()
```

Running the above code you will get a list containing the square value of the numbers from 1 to 5.

Lesson 38:

Question:

Defining a function can create a list containing the square values ??of the numbers from 1 to 20 (including 1 and 20), and then print the last 5 items in the list.

Suggestions:

Similar to lesson 37.

Sample code:

```
def printList():
li=list()
for i in range(1,21):
li.append(i**2)
print li[-5:]
# Bài Python 38, Code by TipsMake.com
printList()
```

When running the above code, you will get a list containing squared values ??of 16, 17, 18, 19, 20.

Lesson 39:

Question:

Defining a function can create a list containing the square value of the numbers from 1 to 20 (including 1 and 20). Then print all list values, except the first 5 items.

Suggestions:

Similar articles 37, 38.

Sample code:

```
def printList():
li=list()
for i in range(1,21):
li.append(i**2)
print (li[5:])
printList()
```

Result:

```
[36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 289, 324, 361, 400]
```

Lesson 40:

Question:

Defining a function can create and print a tuple containing square values ??of numbers from 1 to 20 (including 1 and 20).

Suggestions:

1. Use the ** operator to get the square value.
2. Use range () for the loop.
3. Use list.append () to add values ??to the list.
4. Use tuple () to get the tuple price from the list.

Sample code:

```
def printTuple():
li=list()
for i in range(1,21):
li.append(i**2)
print (tuple(li))

printTuple()
```

Result:

```
(1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 289, 324, 3
```

Lesson 41:

Question:

With a given tuple (1,2,3,4,5,6,7,8,9,10), write a program that prints half of the first value in 1 line and half of the last value in 1 current.

Suggestions:

Use [n1: n2] to get a part from tuple.

Sample code:

```
tp=(1,2,3,4,5,6,7,8,9,10)
tp1=tp[:5]
tp2=tp[5:]
print (tp1)
print (tp2)
```

Result:

```
(1, 2, 3, 4, 5)
(6, 7, 8, 9, 10)
```

Lesson 42:

Question:

Write a program to create another tuple, containing values ??that are even in the tuple (1,2,3,4,5,6,7,8,9,10) given.

Suggestions:

1. Use for to loop tuple.
2. Use tuple () to create a tuple from the list.

Sample code:

```
tp=(1,2,3,4,5,6,7,8,9,10)
li=list()
for i in tp:
if tp[-i]%2==0:
li.append(tp[i])

tp2=tuple(li)
print (tp2)
```

Result:

```
(2, 4, 6, 8, 10)
```

Lesson 43:

Question:

Writing a program to create and print tuple contains even numbers taken from tuple (1,2,3,4,5,6,7,8,9,10).

Suggestions:

1. Use "for" to repeat tuple.
2. Use tuple () to create a tuple from a list.

Sample code:

```
tp=(1,2,3,4,5,6,7,8,9,10)
li=list()
for i in tp:
if tp[i-1]%2==0:
   li.append(tp[i-1])
      tp2=tuple(li)
print tp2
```

Result:

(2, 4, 6, 8, 10)

Lesson 44:

Request:

Write a Python program that receives the string entered by the user, print "Yes" if the string is "yes" or "YES" or "Yes", otherwise print "No".

Suggestions:

1. Use the if command to check conditions.

Sample code:

```
s = input ()
if s == "yes" or s == "YES" or s == "Yes":
  print ("Yes")
else:
  print ("No")
```

Lesson 45:

Request:

Writing Python programs can filter even numbers in the list using the filter function. The list is [1,2,3,4,5,6,7,8,9,10].

Suggestions:

1. Use filter () to filter elements in a list.
2. Use lambda for unknown function definitions.

Sample code:

```
li = [1,2,3,4,5,6,7,8,9,10]
evenNumbers = list(filter (lambda x: x% 2 == 0, li))
print (evenNumbers)
```

Result:

[2, 4, 6, 8, 10]

Note: In previous Python versions, you only need to use the filter function to return the output as a list. But from Python 3, you must use list (filter ()), then the result will be list. This also applies to map ().

Lesson 46:

Request:

Write a Python program using map () to create a list containing square values ??of numbers in [1,2,3,4,5,6,7,8,9,10].

Suggestions:

1. Use map () to create the list.
2. Use lambda for unknown function definitions.

Sample code:

```
li = [1,2,3,4,5,6,7,8,9,10]
squaredNumbers = list(map (lambda x: x ** 2, li))
print (squaredNumbers)
```

Result:

[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]

Lesson 47:

Request:

Write a Python program using map () and filter () to create a list containing the square value of even numbers in [1,2,3,4,5,6,7,8,9,10].

Suggestions:

1. Use map () to create the list.
2. Use filter () to filter elements in the list.
3. Use lambda to define unknown functions.

Sample code:

```
li = [1,2,3,4,5,6,7,8,9,10]

squareOfEvenNumbers = list (map (lambda x: x ** 2, filter (lambda x: x% 2 == 0,

print (squareOfEvenNumbers)
```

Result:

[4, 16, 36, 64, 100]

Lesson 48:

Request:

Write a program Python using filter () to create a list containing even numbers in paragraph [1.20].

Suggestions:

1. Lesson 45.

Sample code:

```
evenNumbers = list(filter (lambda x: x% 2 == 0, range (1,21)))
print (evenNumbers)
```

Result:

[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]

Lesson 49:

Request:

Write a Python program that uses map () to create a list containing the square value of the numbers in paragraph [1.20].

Suggestions:

1. Lesson 46.

Sample code:

```
squaredNumbers = list(map(lambda x: x ** 2, range (1,21)))
print (squaredNumbers)
```

Result:

[1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 289, 324, 361, 400]

Lesson 50:

Request:

Define a class named Vietnam, with the static method is printNationality.

Suggestions:

Use @staticmethod to define class with static method.

Sample code:

```
class Vietnam (object):
  @staticmethod
  def printNationality ():
    print ("Vietnam")
 # Lesson Python 50, Code by TipsMake.com
VietnamVodich = Vietnam ()
VietnamVodich.printNationality ()
Vietnam.printNationality ()
```

Lesson 51:

Request:

Defining a class named Vietnam and its subclass is Hanoi.

Suggestions:

1. Use `Subclass(ParentClass)` to define a subclass.

Sample code:

```
class Vietnam(object):
  pass

 class Hanoi(Vietnam):
  pass
  # Lesson Python 51, Code by TipsMake.com
 VietnamVodich = Vietnam()
  div="">
```

You finished reading the article "**More than 100 Python exercises have solutions (sample code)**" edited by the **TipsMake** team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.

© 2019 TipsMake.com