

NAVTC - Corvit Systems

Course Name:

- Artificial Intelligence (Robotics)

Python Programming Task:

Check your Python Programming learning progress and take your skills to the next level with Corvit Systems interactive tasks.

We created these Python tasks as a fun way for you to check your learning progress and to test your skills.

- Task No. : 01
- Skill Level : Basic
- Total Questions: 18
- Time Limit: 1 Hour

Python Programming Task covered the following Topics/Concepts

- Style Guide for Python Code
- Python Variables
- Basic Python Data Types
 - Integer, Boolean, Floating, Complex, String
- Basic Python Built-in Methods & Conversions
 - print, type, help, int, bool, float, str
- Basic Python Operators & Expressions
 - Arithmetic Operators
- Complete String Data Type
 - String Variables
 - Basic String Operations (+, *, in)
 - Advance String Operations (Indexing & Slicing)
 - Built-in String Methods
 - String Formatting (str.format, f-string)

In []:

Ready to Task Your Basic Python Programming Fundamentals Skills ...

Question 1:

Write a program to assign different and same values to multiple variables in one line, then display variables and type with one print statement.

```
In [1]: # assign different values to multiple variables in one line
a, b = 5, 4.8
# assign same values to multiple variables in one line
x = y = 20
```

Display Method 1

```
In [2]: print(a, type(a), b, type(b), x, type(x), y, type(y))

5 <class 'int'> 4.8 <class 'float'> 20 <class 'int'> 20 <class 'int'>
```

Display Method 2

```
In [3]: print(f'a = {a} - (type(a)), b = {b} - (type(b)) \n x = {x}, {y} - (type(x)), {type(y)}')

a = 5 - <class 'int'>, b = 4.8 - <class 'float'>
x = y = 20, 20 - <class 'int'>, <class 'int'>
```

Question 2:

Write a program that performs all arithmetic assignment operations on one variable.

```
In [4]: x = 25
```

Method 1 - Arithmetic Operations

```
In [5]: print("Addition : " + str(x + 5))
print("Subtraction : " + str(x - 5))
print("Multiplication : {}".format(x * 5))
print("Division : " + str(x / 5))
print("Floor Division : " + str(x // 5))
print("Modulus : " + str(x % 5))
print("Exponentiation : " + str(x ** 5))
```

Addition : 30
Subtraction : 20
Multiplication : 125
Division : 5.0
Floor Division : 5
Modulus : 0
Exponentiation : 9765625

```
In [6]: print("Addition : {}".format(x + 5))
print("Subtraction : {}".format(x - 5))
print("Multiplication : {}".format(x * 5))
print("Division : {}".format(x / 5))
print("Floor Division : {}".format(x // 5))
print("Modulus : {}".format(x % 5))
print("Exponentiation : {}".format(x ** 5))
```

Addition : 30
Subtraction : 20
Multiplication : 125
Division : 5.0
Floor Division : 5
Modulus : 0
Exponentiation : 9765625

```
In [7]: print(f"Addition : {x + 5}")
print(f"Subtraction : {x - 5}")
print(f"Multiplication : {x * 5}")
print(f"Division : {x / 5}")
print(f"Floor Division : {x // 5}")
print(f"Modulus : {x % 5}")
print(f"Exponentiation : {x ** 5}")
```

Addition : 30
Subtraction : 20
Multiplication : 125
Division : 5.0
Floor Division : 5
Modulus : 0
Exponentiation : 9765625

Method 2 - Arithmetic Operations

```
In [8]: x = 25
x = x + 5
print(f"Addition : {x}")
x = x - 5
print(f"Subtraction : {x}")
x = x * 5
print(f"Multiplication : {x}")
x = x / 5
print(f"Division : {x}")
x = x // 5
print(f"Floor Division : {x}")
x = x % 5
print(f"Modulus : {x}")
x = x ** 5
print(f"Exponentiation : {x}")
```

Addition : 30
Subtraction : 25
Multiplication : 125
Division : 25.0
Floor Division : 5.0
Modulus : 0.0
Exponentiation : 0.0

Method 3 - Arithmetic Assignment Operations

```
In [9]: x = 25
x += 5
print(f"Addition : {x}")
x -= 5
print(f"Subtraction : {x}")
x *= 5
print(f"Multiplication : {x}")
x /= 5
print(f"Division : {x}")
x //= 5
print(f"Floor Division : {x}")
x %= 5
print(f"Modulus : {x}")
x **= 5
print(f"Exponentiation : {x}")
```

Addition : 30
Subtraction : 25
Multiplication : 125
Division : 25.0
Floor Division : 5.0
Modulus : 0.0
Exponentiation : 0.0

Question 3:

Write a program that stored a three-digit number as integer and reverse the number.

```
In [10]: num = 689
```

Method 1

```
In [11]: a = num // 100
num = num % 100
b = num // 10
num = num % 10
```

```
In [12]: print(f"Number in Reverse Order is : {num}{b}{a}")

Number in Reverse Order is : 986
```

Method 2

```
In [13]: n1 = 689
n2 = str(689)
```

```
In [14]: print(f"Number in Reverse Order is : {n2[::-1]}")

Number in Reverse Order is : 986
```

Question 4:

Write a program that stored two integer numbers, swaps these values without using third variable and displays them.

```
In [15]: x = 10
y = 4
```

```
In [16]: print(f"Original value in x = {x} and y = {y}")
x = x + y
y = x - y
x = x - y
print(f"Swapped value in x = {x} and y = {y}")
```

Original value in x = 10 and y = 4
Swapped value in x = 4 and y = 10

Question 5:

Target String: Python Programming Fundamentals.

Expected Result: to rgamn udmnas

```
In [17]: s = 'Python Programming Fundamentals.'

In [18]: print(s[2:22])

to rgamn udmnas

In [19]: print(s[2:len(s):2])

to rgamn udmnas
```

Question 6:

Write a program that concate all basic python data type variables.

```
In [20]: v1 = 45
v2 = 36.8
v3 = True
v4 = 22j
v5 = 'python'

In [21]: v6 = str(v1) + ", " + str(v2) + ", " + str(v3) + ", " + str(v4) + ", " + v5

In [22]: print(v6)

45, 36.8, True, 22j, python
```

Question 7:

Write a program that format() method takes 4 number of arguments, and are placed into the respective placeholders:

```
In [23]: ar1 = "test val 1"
ar2 = "test val 2"
ar3 = "test val 3"
ar4 = "test val 4"

In [24]: print('{}, {}, {}, {}'.format(ar1, ar2, ar3, ar4))

test val 1, test val 2, test val 3, test val 4

In [25]: print('{0}, {1}, {2}, {3}'.format(ar1, ar2, ar3, ar4))

test val 1, test val 2, test val 3, test val 4

In [26]: print('{0:1}, {ar2}, {ar3}, {ar4}'.format(ar1=5, ar2="python", ar3=2.7, ar4=True))

5, python, 2.7, True
```

Question 8:

Target

Book Id: 5
Book Name: Deep Learning using PyTorch
Price: 100

Expected Result - Book Info - Book Id : 5, Book Name : Deep Learning using PyTorch, Price : 100\$

```
In [27]: bookId = 5
bookName = 'Deep Learning using PyTorch'
price = 100

In [28]: print(f'Book Info - Book Id : {bookId}, Book Name : {bookName}, Price : {price}$')

Book Info - Book Id : 5, Book Name : Deep Learning using PyTorch, Price : 100$
```

Question 9:

Target Strings:

machine learning

Expected Result:

mACHINE LEARNING

```
In [29]: s1 = 'machine learning'

In [30]: s2 = s1.upper().split()

In [31]: print(s2)

['MACHINE', 'LEARNING']

In [32]: w1 = s2[0][0].lower() + s2[0][1:]

In [33]: w2 = s2[1][0].lower() + s2[1][1:]

In [34]: print(f'{w1} {w2}')

mACHINE LEARNING
```

Question 10:

Target Value:

15.4567

Expected Result:

```
***15.457***

In [35]: val = 15.4567
```

```
In [36]: # float number with center alignment
print("{0:**12.3f} ".format(val))

***15.457***
```

Question 11:

Difference between python string find and index methods with example.

```
In [37]: # The find() method returns the index of first occurrence of the substring
# (if found). If not found, it returns -1.

# The index() method returns the index of a substring inside the string (if found).
# If the substring is not found, it raises an exception.

# The index() method is similar to find() method for strings.
# The only difference is that find() method returns -1 if the substring is not found,
# whereas index() throws an exception.
```

```
In [38]: s = "Python Programming"

In [39]: s.find('ML')
```

Out[39]: -1

```
In [40]: s.find('Pro')
```

Out[40]: 7

```
In [41]: s.index('ML')

-----
ValueError                                Traceback (most recent call last)
<ipython-input-41-03ecdc4fada6> in <module>
----> 1 s.index('ML')

ValueError: substring not found
```

```
In [42]: s.index('Pro')

7

Out[42]: 7
```

Question 12:

Write a Python program to remove spaces from a given string.

```
In [43]: s1 = "Python Programming Fundamentals"

In [44]: s2 = s1.replace(' ', '')

In [45]: print(s2)

PythonProgrammingFundamentals
```

Question 13:

Write a Python program to check whether a string starts with specified characters.

```
In [46]: s = 'Python Programming Fundamentals'

In [47]: result = s.startswith('P')

In [48]: print(type(result))

<class 'bool'>
```

```
In [49]: print(result)

True
```

Question 14:

Write a program to check if the letter 'r' is present in the word 'Python'.

```
In [50]: s = 'Python'

In [51]: result = 'r' in s

In [52]: print(result)

False
```

Question 15:

Write a program that find all occurrences of "python" in given string ignoring the case.

```
In [53]: s = '''Python is a powerful general-purpose programming language.
Python is simple easy-to-use syntax.'''

In [54]: print(s.lower().count('python'))

2
```

Question 16:

Write a program that stored your full name as a string variabe and displays the abbreviations of the first and middle names except the last name which is displayed as it is. For example, if your name is Engineer Rizwan Mughal, then the output should be E.R.Mughal.

```
In [55]: fullName = 'Engineer Rizwan Mughal'

In [56]: name = fullName.split()

In [57]: print(name)

['Engineer', 'Rizwan', 'Mughal']

In [58]: print(name[0])

Engineer

In [59]: print(name[0][0])

E

In [60]: print(name[1])

Rizwan

In [61]: print(name[1][0])

R

In [62]: print(name[2])

Mughal

In [63]: shortName = name[0][0] + "." + name[1][0] + "." + name[2]

In [64]: print(shortName)

E.R.Mughal
```

Question 17:

Display the following output:

C:\Users\UserName>

```
In [65]: filePath = 'C:\Users\UserName>'

In [66]: print(filePath)

C:\Users\UserName>
```

Question 18:

Target String:

0300-1000001

Expected Result:

```
Network Code: 0300
Actual Code: 1000001

In [67]: cellNo = '0300-1000001'

In [68]: splitCellNo = cellNo.split('-')
```

```
In [69]: print(splitCellNo)

['0300', '1000001']

In [70]: print(f'Network Code: {splitCellNo[0]}')
print(f'Actual Code: {splitCellNo[1]}')
```

Happy Learning 😊