

- Print current date and time in Python
- Print time only

#### **Expected Output**

2021-03-26 17:49:30.070618

17:48:34.820141

 Write a Python program to get the Python version you are using.

A string containing the version number of the Python interpreter plus additional information on the build number and compiler used. This string is displayed when the interactive interpreter is started.

#### **Version info:**

A tuple containing the five components of the version number: major, minor, micro, release level, and serial. All values except release level are integers; the release level is 'alpha', 'beta', 'candidate', or 'final'. The version\_info value corresponding to the Python version 2.0 is (2, 0, 0, 'final', 0). The components can also be accessed by name, so sys.version\_info[0] is equivalent to sys.version\_info. major and so on.

Note: 'sys' module provides access to some variables used or maintained by the interpreter and to functions that interact strongly with the interpreter.

 Print first and last name in reverse order with a space between them



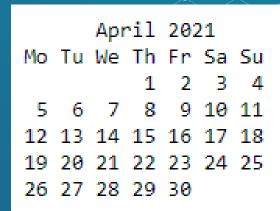
Mostafa Ali

#### **Expected Output**

Ali Mostafa

 Write a Python program to print the calendar of a given month and year.

Note: Use 'calendar' module.



Write a Python programto parse a string to Floator Integer.

#### Input

n = "246.2458"

#### **Expected Output**

246.2458 246

Repeat the word many times

#### Input

"Hello World" 3

#### **Expected Output**

"Hellow World"
"Hellow World"
"Hellow World"

 Write a Python program to get a string made of the first 2 and the last 2 chars from a given a string. If the string length is less than 2, return instead the empty string

#### Input

'w3resource'

'w3'

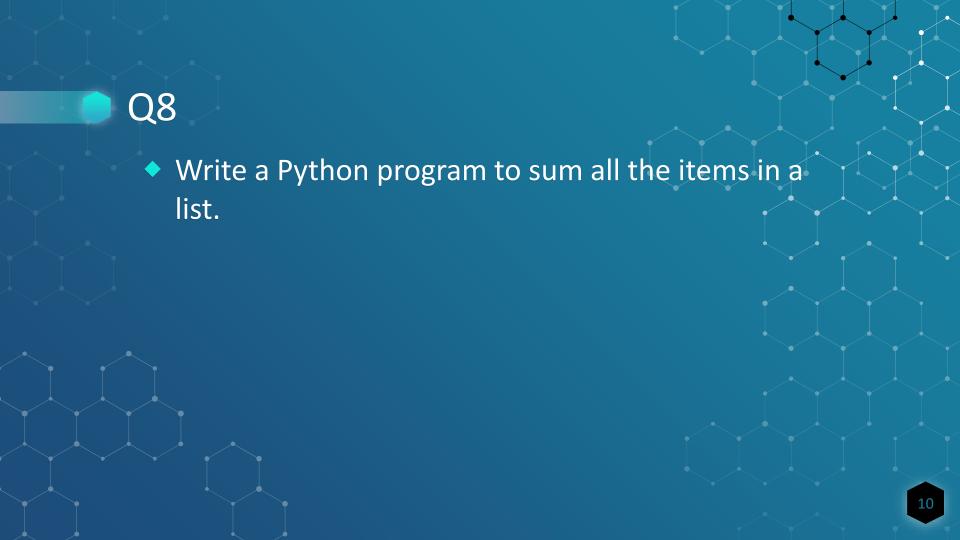
'w'

#### **Expected Output**

'w3ce'

'w3w3'

**Empty String** 





 Write a Python program to calculate the value of equation of second degree

$$y = 3x^2 + 4x - 3$$

## Concatenate two lists index-wise

#### Given:

```
list1 = ["M", "na", "i", "Ke"]
list2 = ["y", "me", "s", "lly"]
```

```
['My', 'name', 'is', 'Kelly']
```

### Given a Python list of numbers. Turn every item of a list into its square

#### Given:

#### **Expected output:**

[1, 4, 9, 16, 25, 36, 49]

```
list1 = ["Mike", "", "Emma", "Kelly", "", "Brad"]
```

#### **Expected output:**

```
["Mike", "Emma", "Kelly", "Brad"]
```

Remove empty strings from the list of strings

```
aTuple = (10, 20, 30, 40, 50)
```

#### Expected output:

(50, 40, 30, 20, 10)

Reverse the tuple

```
aTuple = (10, 20, 30, 40)
```

#### Expected output:

```
aTuple = (10, 20, 30, 40)

# Your code

print(a) # should print 10

print(b) # should print 20

print(c) # should print 30

print(d) # should print 40
```

## Unpack the following tuple into variables

```
tuple1 = (50, 10, 60, 70, 50)
```

#### **Expected output:**

2

# Counts the number of occurrences of item from a tuple

```
set1 = {10, 20, 30, 40, 50}
set2 = {30, 40, 50, 60, 70}
```

#### **Expected output:**

{40, 50, 30}

Return a new set of identical items from a given two set

Returns a new set with all items from both sets by removing duplicate

```
set1 = {10, 20, 30, 40, 50}
set2 = {30, 40, 50, 60, 70}
```

#### **Expected output:**

{70, 40, 10, 50, 20, 60, 30}

Given two Python sets, update the first set with items that exist only in the first set and not in the second set

### Convert two lists int keys = ['Ten', 'Twenty', 'Thirty'] a dictionary

```
values = [10, 20, 30]
```

```
{'Ten': 10, 'Twenty': 20, 'Thirty': 30}
```

Create a new dictionary by extracting the specific keys from a another dictionary

```
sampleDict = {
   "name": "Kelly",
   "age":25,
   "salary": 8000,
   "city": "New york"
}
```

#### Keys to extract

```
keys = ["name", "salary"]
```

```
{'name': 'Kelly', 'salary': 8000}
```

## Delete set of keys from a dictionary

#### Given:

```
sampleDict = {
   "name": "Kelly",
   "age":25,
   "salary": 8000,
   "city": "New york"

}
keysToRemove = ["name", "salary"]
```

```
{'city': 'New york', 'age': 25}
```

### THANKS!

