

## Tutorial Week 1: Python Basics (Questions)

[Click to download the Jupyter Notebook files \(.ipynb\)](#)

### 0.1 Example: Calculate the multiplication and sum of two numbers

Given two integer numbers return their product only if the product is equal to or lower than 1000, else return their sum.

Hint:

- Create a function that will take two numbers as parameters.
- Next, inside the function, multiply two numbers and save their product in a **product** variable.
- Next, use the if condition to check if the **product** > 1000. If yes, return the **product**.
- Otherwise, use the **else** block to calculate the sum of two numbers and return it.

```
[1]: def multiplication_or_sum(num1, num2):  
    ### Start your code here ###  
  
    product = num1 * num2  
    if product <= 1000:  
        return product  
    else:  
        return num1 + num2  
  
    ### End your code here ###  
  
    # first condition  
    result = multiplication_or_sum(20, 30)  
    print("The result is", result)  
  
    # second condition  
    result = multiplication_or_sum(40, 30)  
    print("The result is", result)
```

The result is 600

The result is 70

## 1 Questions

Hint:

- Write **your code** between two comment lines: **### Start/End your code here ###**.
- **Expected output** is shown at the end of each question (directly below the code cell).

### 1.1 Find the largest item from a given list

Hint:

- Use the built-in function `max()` to get the largest number from a list.

```
[2]: x = [4, 6, 8, 24, 12, 2]
    ### Start your code here ###

    ### End your code here ###
```

24

### 1.2 Return multiple values from a function

Write a program to create function `calculation()` such that it can accept two variables and calculate addition and subtraction. Also, it must return both addition and subtraction in a single return call.

Hint:

- Separate return values with a comma.

```
[3]: def calculation(a, b):
    ### Start your code here ###

    ### End your code here ###

    # get result in tuple format
    res = calculation(40, 10)
    print(res)
```

(50, 30)

### 1.3 Print the sum of the current number and the previous number

Write a program to iterate the first 10 numbers and in each iteration, print the sum of the current and previous number.

Hint:

- Create a variable called `previous_num` and assign it to 0.
- Iterate the first 10 numbers one by one using for loop and built-in function `range()`.
- Next, display the current number (`i`), previous number, and the addition of both numbers in each iteration of the loop. At last, change the value previous number to current number (`previous_num = i`).

```
[4]: print("Printing current and previous number and their sum in a range(10)")
    previous_num = 0

    for i in range(1, 11):
        ### Start your code here ###
```

```
### End your code here ###
```

Printing current and previous number and their sum in a range(10)

```
Current Number 1 Previous Number 0 Sum: 1
Current Number 2 Previous Number 1 Sum: 3
Current Number 3 Previous Number 2 Sum: 5
Current Number 4 Previous Number 3 Sum: 7
Current Number 5 Previous Number 4 Sum: 9
Current Number 6 Previous Number 5 Sum: 11
Current Number 7 Previous Number 6 Sum: 13
Current Number 8 Previous Number 7 Sum: 15
Current Number 9 Previous Number 8 Sum: 17
Current Number 10 Previous Number 9 Sum: 19
```

#### 1.4 Display numbers divisible by 5 from a list

Iterate the given list of numbers and print only those numbers which are divisible by 5.

```
[5]: num_list = [10, 20, 33, 46, 55]
print("Given list:", num_list)
print('Divisible by 5:')

### Start your code here ###

### End your code here ###
```

Given list: [10, 20, 33, 46, 55]

Divisible by 5:

```
10
20
55
```

#### 1.5 Check if the first and last number of a list is the same

Write a function to return True if the first and last number of a given list is same. If numbers are different then return False.

```
[6]: def first_last_same(numberList):
    print("Given list:", numberList)

    ### Start your code here ###

    ### End your code here ###

numbers_x = [10, 20, 30, 40, 10]
```

```
print("result is", first_last_same(numbers_x))

numbers_y = [75, 65, 35, 75, 30]
print("result is", first_last_same(numbers_y))
```

Given list: [10, 20, 30, 40, 10]

result is True

Given list: [75, 65, 35, 75, 30]

result is False

## 1.6 Create a new list from two lists using the following condition

Given two lists of numbers, write a program to create a new list such that the new list should contain odd numbers from the first list and even numbers from the second list.

Hint:

- Create an empty list named `result_list`.
- Iterate first list using a for loop.
- In each iteration, check if the current number is odd number using `num % 2 != 0` formula. If the current number is an odd number, add it to the result list.
- Now, Iterate the first list using a loop.
- In each iteration, check if the current number is odd number using `num % 2 == 0` formula. If the current number is an even number, add it to the result list.

```
[7]: def merge_list(list1, list2):
    result_list = []

    ### Start your code here ###

    ### End your code here ###

list1 = [10, 20, 25, 30, 35]
list2 = [40, 45, 60, 75, 90]
print("result list:", merge_list(list1, list2))
```

result list: [25, 35, 40, 60, 90]

## 1.7 Remove first n characters from a string

Write a program to remove characters from a string starting from zero up to `n` and return a new string. For example:

- `remove_chars("pynative", 4)` so output must be `tive`. Here we need to remove first four characters from a string.
- `remove_chars("pynative", 2)` so output must be `native`. Here we need to remove first two characters from a string.

Hint: - Use string slicing to get the substring. For example, to remove the first four characters and the remaining use `s[4:]`.

```
[8]: def remove_chars(word, n):  
    ### Start your code here ###  
  
    ### End your code here ###  
  
    print("Removing characters from a string")  
    print(remove_chars("pynative", 4))  
    print(remove_chars("pynative", 2))
```

Removing characters from a string  
Original string: pynative  
tive  
Original string: pynative  
native

### 1.8 Calculate income tax for the given income by adhering to the below rules

Taxable Income	Rate (in %)
First \$10,000	0
Next \$10,000	10
The remaining	20

Example:

- Suppose the taxable income is 45000 the income tax payable is  $10000 \times 0\% + 10000 \times 10\% + 25000 \times 20\% = \$6000$ .

```
[9]: income = 45000  
    print("Given income", income)  
  
    tax_payable = 0  
  
    ### Start your code here ###  
  
    ### End your code here ###  
  
    print("Total tax to pay is", tax_payable)
```

Given income 45000  
Total tax to pay is 6000.0

### 1.9 Count the occurrence of each element from a list

Write a program to iterate a given list and count the occurrence of each element and create a dictionary to show the count of each element.

```
[10]: sample_list = [11, 45, 8, 11, 23, 45, 23, 45, 89]  
    print("Original list ", sample_list)
```

```
count_dict = dict()

### Start your code here ###

### End your code here ###

print("Printing count of each item ", count_dict)
```

Original list [11, 45, 8, 11, 23, 45, 23, 45, 89]

Printing count of each item {11: 2, 45: 3, 8: 1, 23: 2, 89: 1}

### 1.10 Append new string in the middle of a given string

Given two strings, `s1` and `s2`. Write a program to create a new string `s3` by appending `s2` in the middle of `s1`.

Hint:

- Use built-in function `len(s1)` to get the string length.
- Next, get the middle index number by dividing string length by 2.

```
[11]: def append_middle(s1, s2):
        print("Original Strings are:", s1, s2)

        ### Start your code here ###

        ### End your code here ###

append_middle("Ault", "Kelly")
```

Original Strings are: Ault Kelly

After appending new string in middle: AuKellylt