

## ✓ Lab 1: Basic Python Programming

### ✓ 1. Basic usage

John Doe is a 29 years-old system engineer who earns ₦41500.00 a month.

Create and assign variables to store this person's information (name, age, position and salary).

```
name = "John Doe"
age = 29
position = "System Engineer"
salary = 41500.00

print(name)
print(age)
print(position)
print("₦{:.2f}".format(salary))

John Doe
29
System Engineer
₦41500.00
```

What is the type of each variables?

```
print(type(name))
print(type(age))
print(type(position))
print(type(salary))

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

The manager decides to give John a 7% raise. Update his salary.

```
raise_salary = salary * 1.07
print("Updated Salary: ₦{:.2f}".format(raise_salary))

Updated Salary: ₦44405.00
```

Prints his information again with his new salary.

```
print(name)
print(age)
print(position)
print("₦{:.2f}".format(raise_salary))

John Doe
29
System Engineer
₦44405.00
```

Now, he decides to resign. Delete his information from the system.

```
del name
del age
del position
del raise_salary
```

## ✓ 2. Variable and Expression

**2.1** Write a code to convert temperature unit from celcius to other units

C = 34.5

### Fahrenheit

$$✓ \quad \frac{C}{5} = \frac{F-32}{9}$$

F = 94.1

```
celcius = 34.5
fahrenheit = (celcius*9/5)+32
```

```
print(fahrenheit)
```

```
94.1
```

Double-click (or enter) to edit

### Kelvin

$$K = C + 273.15$$

K = 307.65

```
kelvin = (celcius+273.15)
```

```
print(kelvin)
```

```
307.65
```

### Rømer

$$✓ \quad Ro = \frac{C \times 21}{40} + 7.5$$

Ro = 25.6125

```
romer = (celcius*21/40)+7.5
```

```
print(romer)
```

```
25.6125
```

## ✓ 3. Multi-item variables

### List

```
names = ['Thomas', 'Kate', 'Mike', 'Amelia', 'James', 'Megan']
```

Create new variable call `new_name` which takes input name of the user.

```
new_name = input('Enter your name: ')
```

```
Enter your name: Nonpitch
```

Insert `new_name` into `names` list.

```
names.append(new_name)
```

```
print(names)
```

```
['Thomas', 'Kate', 'Mike', 'Amelia', 'James', 'Megan', 'Nonpitch', 'Nonpitch']
```

Select your name from the list

```
print(names[7:])
```

```
['Nonpitch']
```

Merge another\_names into names.

```
another_names = ['Peter', 'Steve', 'Sam', 'Charlotte']
```

```
print(names + another_names)
```

```
['Thomas', 'Kate', 'Mike', 'Amelia', 'James', 'Megan', 'Nonpitch', 'Nonpitch', 'Peter', 'Steve', 'Sam', 'Charlotte']
```

Change Amelia's name to Amy

```
names[3] = "Amy"
```

```
print(names)
```

```
['Thomas', 'Kate', 'Mike', 'Amy', 'James', 'Megan', 'Nonpitch', 'Nonpitch']
```

## ▼ Dictionary

```
capital_city = {'England': 'London',
                'Spain': 'Madrid',
                'Japan': 'Tokyo',
                'Australia': 'Sydney',
                'Germany': 'Berlin',
                }
```

Add a record Thailand and it's capital city to this dictionary

```
capital_city['Thailand'] = 'Bangkok'
```

```
print(capital_city)
```

```
{'England': 'London', 'Spain': 'Madrid', 'Japan': 'Tokyo', 'Australia': 'Sydney', 'Germany': 'Berlin', 'Thailand': 'Bangkok'}
```

You may notice that the capital city of Australia is wrong. It should be Canberra. Correct this mistake.

```
capital_city['Australia'] = 'Canberra'
```

```
print(capital_city)
```

```
{'England': 'London', 'Spain': 'Madrid', 'Japan': 'Tokyo', 'Australia': 'Canberra', 'Germany': 'Berlin', 'Thailand': 'Bangkok'}
```

## ▼ 4. Control Flows and conditional statements

### ▼ if...elif...else

1. Define a variable to get input age from user.

```
age = int(input("Your age: "))
```

```
Your age: 19
```

Write a series of if...elif...else statement that categorize input age into following groups:

```
Babies: 0-2 years old
Children: 3-12 years old
Teenager: 13-19 years old
Young Adults: 20-29 years old
Middle-aged Adults: 30-45 years old
Old Adult: 46-59 years old
Elderly: Above 60 years old
```

```
if 0<=age<=2:
    group = "Babies"
elif 3<=age<=12:
    group = "Children"
elif 13<=age<=19:
    group = "Teenager"
elif 20<=age<=29:
    group = "Young Adults"
elif 30<=age<=45:
    group = "Middle-aged Adults"
elif 46<=age<=59:
    group = "Old Adult"
elif age>=60:
    group = "Elderly"
```

```
print("You are:", group)
```

```
You are: Teenager
```

## ✓ Looping

1. Write a code to create a multiplication table of an input number (multiplier from 1-12).

```
number = int(input("Input a number: "))
```

```
for i in range(1, 13):
    ans = number*i
    print(f"{number} * {i} = {ans}")
```

```
Input a number: 4
4 * 1 = 4
4 * 2 = 8
4 * 3 = 12
4 * 4 = 16
4 * 5 = 20
4 * 6 = 24
4 * 7 = 28
4 * 8 = 32
4 * 9 = 36
4 * 10 = 40
4 * 11 = 44
4 * 12 = 48
```

2. Write a code that construct the following pattern.

Unsupported Cell Type. Double-Click to inspect/edit the content.

```
rows = int(input("input: "))
```

```
print("output:")
for i in range(1, rows+1):
    print("*" * i)
```

```
input: 5
output:
```

```

*
**
***
****
*****

```

3. Creates a loop to print I love <programming language>! except for Assembly, print Not you, Assembly.

```
languages = ['C/C++', 'Python', 'R', 'Java', 'SQLs', 'Assembly', 'Go', 'Rust', 'Kotlin']
```

```

for code_lang in languages:
    if code_lang == 'Assembly':
        print("Not you, Assembly")
    else:
        print("I love", code_lang)

```

```

I love C/C++
I love Python
I love R
I love Java
I love SQLs
Not you, Assembly
I love Go
I love Rust
I love Kotlin

```

4. Write a code to print every number from 1 to 25 except the one that is divisible by 3.

```

for i in range(1, 26):
    if i%3 == 0:
        continue
    else:
        print(i)

```

```

1
2
4
5
7
8
10
11
13
14
16
17
19
20
22
23
25

```

5. Write a code that finds the number that is divisible by 7 in a given range.

```

lower_bound = 1
upper_bound = 100
divisor = 7

```

```
result = []
```

```

for i in range(1, 101):
    if i%7 == 0:
        print(i)

```

```

7
14
21
28
35
42
49
56
63
70
77

```

84  
91  
98

6. Write a code that construct the following pattern.

```
input: 5
output:
#####
*#####
**#####
***#####
****###
*****#
```

```
input: 10
output:
#####
*#####
**#####
***#####
****#####
*****#####
*****#####
*****#####
*****#####
*****#####
*****#####
*****#####
```

```
num = int(input("input: "))
```

```
for i in range(1, num+1):
    print(" "*i + "#"*num)
    num = num-1
```

```
input: 5
#####
*#####
**#####
***#####
****###
*****#
```

## ✓ 5. Functions

1. Define a function `average` that takes arbitrary number of arguments and calculate the mean of input.

```
def average(numbers):
    return sum(numbers) / len(numbers) if numbers else None
```

```
user_input = input("input numbers by space: ")
numbers = [float(num) for num in user_input.split()]
mean = average(numbers)
print("average: ", mean)
```

```
input numbers by space: 2 5 6 7
average: 5.0
```

2. Define a function `sumproduct` that takes 2 equal-sized lists and calculate sum of the products of two lists.

It should look like this:

```
sumproduct([1,2,3],[4,5,6])
output: 32
```

$(1 * 4) + (2 * 5) + (3 * 6) = 32$

```
def sumproduct(list1, list2):
    if len(list1) != len(list2):
        return None
    else:
        return sum(x*y for x,y in zip(list1, list2))

ans = sumproduct([1,2,3],[4,5,6])
print("output: ", ans)

output: 32
```

3. Define a function `fibonacci` that returns Fibonacci number at `n` position.

A Fibonacci number at position `n` is defined by  $F(n) = F(n-1) + F(n-2)$ . Where  $F(0) = 0$  and  $F(1) = 1$

**Example:** 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ...

```
def fibonacci(n):
    if n == 0:
        return 0
    elif n == 1:
        return 1
    else:
        return fibonacci(n-1) + fibonacci(n-2)

position = int(input("position: "))
ans = fibonacci(position)
print(f"Fibonacci number at {position} position:", ans)

position: 8
Fibonacci number at 8 position: 21
```

4. Define a function `is_palindrome` that takes input string and check whether it is a palindrome or not.

A string is a palindrome if it reads the same forward and backwards.

**Example:** madam, race car, borrow or rob, amore roma, never odd or even

Do not consider whitespace. Use `str.replace(' ', '')` to remove whitespace from your string.

Case-insensitive. You can turn everything into lower or uppercase using `str.lower()` or `str.upper()`

**Hint:** you can reverse the string using `[::-1]` slice.

```
str1 = "radar" # palindrome
str2 = "rotator" # palindrome
str3 = "lemon" # not palindrome

def is_palindrome(input_str):
    removed_str = input_str.replace(' ', '').lower()
    return removed_str == removed_str[::-1]

user_input = input("input a string: ")
result = is_palindrome(user_input)

print(f"{user_input} {'is' if result else 'is not'} a palindrome")

input a string: 404
404 is a palindrome
```

5. An anagram is a word or phrase formed by rearranging the letters of a different word or phrase.

Define a function `is_anagram` that takes in 2 strings and check whether it is possible to compose a second string using letters in the first string or not.

**Example:** Tom Marvolo Riddle can be rearranged into I am Lord Voldemort

Meaning of Life can be rearranged into Engine of a Film

Do not consider whitespace. Use `str.replace(' ', '')` to remove whitespace from your string.

Case-insensitive. You can turn everything into lower or uppercase using `str.lower()` or `str.upper()`

Returns only True or False

```
# Write your code here
str1 = "Meaning of Life"
str2 = "Engine of a Film"

def is_anagram(str1, str2):
    removed_str1 = str1.replace(' ', '').lower()
```

```
removed_str1 = str1.replace(' ', '').lower()
removed_str2 = str2.replace(' ', '').lower()
return set(removed_str1) == set(removed_str2)
```

```
ans = is_anagram(str1, str2)
print(ans)
```

```
True
```

---