

Applied Calculus for IT - 501031

Lab 03

1. Python Indentation

Indentation refers to the spaces at the beginning of a code line. The indentation in Python is very important, and it indicate a block of code.

- Example:

```
if 5 > 2:  
    print("Five is greater than two!")
```

- Python will give you an error if you skip the indentation:

```
if 5 > 2:  
print("Five is greater than two!")
```

The content of error is: *IndentationError: expected an indented block*

- The number of spaces is up to you as a programmer, but it has to be at least one.

```
if 5 > 2:  
    print("Five is greater than two!")  
if 5 > 2:  
    print("Five is greater than two!")
```

- You have to use the same number of spaces in the same block of code, otherwise Python will give you an error:

```
if 5 > 2:  
    print("Five is greater than two!")  
        print("Five is greater than two!")
```

The content of error is: *IndentationError: unexpected indent*

2. Python main function

Main function is like the entry point of a program. However, Python interpreter runs the code right from the first line. The execution of the code starts from the starting line and goes line by line. It does not matter where the main function is present or it is present or not.

Since there is no *main()* function in Python, when the command to run a Python program is given to the interpreter, the code that is at **level 0 indentation** is to be executed.

3. Python Functions

- Definitions:

- A function is a block of code which only runs when it is called.
- A function is a group of related statements that performs a specific task.

- Advantages:

- Functions help break our program into smaller and modular chunks. As our program grows larger and larger, functions make it more organized and manageable.
- It avoids repetition and makes the code reusable.

- **To call a function**, use the function name followed by parenthesis, and then data can be passed inside the parentheses.

```
print("sum of a and b:")  
range(1, 10, 3)
```

- A function is defined using the **def** keyword:

```
def my_function():  
    print("Hello from a function")  
  
my_function()
```

```
def myPrint():  
    print("abcdxyz")  
    print("12345")
```

```
myPrint()
```

- You can pass data, known as **parameters/arguments**, into a function. Parameters/Arguments are specified after the function name, inside the parentheses. You can add as many arguments as you want, just separate them with a comma.

```
def my_function(fname):  
    print(fname + " Refsnes")  
  
my_function("Emil")  
my_function("Tobias")  
my_function("Linus")
```

- Positional arguments:

By default, a function must be called with the correct number of arguments. Meaning that if your function expects 2 arguments, you have to call the function with 2 arguments, not more, and not less.

```
def my_function(fname, lname):  
    print(fname + " " + lname)  
  
my_function("Emil", "Refsnes")
```

The above arguments *fname* and *lname* are called positional arguments. Positional arguments must be included in the correct order.

- See more “parameters vs arguments” in:

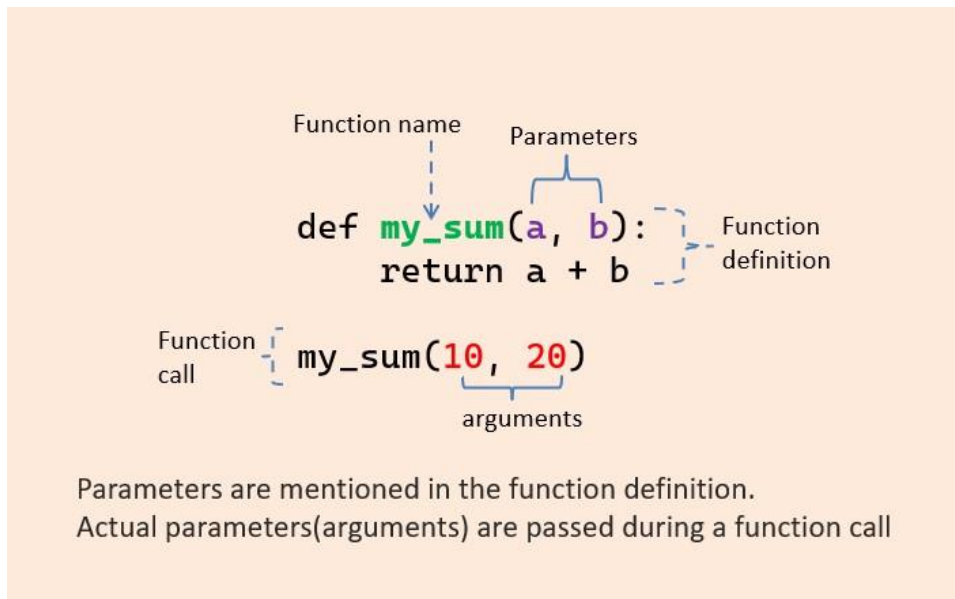
https://www.w3schools.com/python/gloss_python_function_arguments.asp

- To let a function return a value, use the **return** statement:

```
def my_function(x):  
    return 5 * x  
  
print(my_function(3))  
print(my_function(5))  
print(my_function(9))
```

```
def my_sum(a, b):  
    return a + b  
  
my_sum(1, 15)
```

- Summary (<https://pynative.com/python-function-arguments/>):



- Keyword Arguments:

You can also send arguments with the *key = value* syntax. This way the order of the arguments does not matter.

```
def my_function(child3, child2, child1):  
    print("The youngest child is " + child3)  
  
my_function(child1 = "Emil", child2 = "Tobias", child3 = "Linus")
```

- Default arguments:

A default argument is an argument that assumes a default value if a value is not provided in the function call for that argument.

```
# Function definition is here  
def printinfo( name, age = 35 ):  
    "This prints a passed info into this function"  
    print "Name: ", name  
    print "Age ", age  
    return;  
  
# Now you can call printinfo function  
printinfo( age=50, name="miki" )  
printinfo( name="miki" )
```

Non-default argument must not follow the default argument, it means you can't define (**age** = 35, **name**) in the above function “**printinfo**”.

- **Practice examples:**

Define a Python function for each of the following functions:

$$f(x) = \sqrt{x}$$

$$f(x) = \frac{x+1}{x-1}$$

```
import math

def f1_sqrt(x):
    return math.sqrt(x)

def f2_fraction(x):
    res = (x + 1) / (x - 1)
    return res

#main
print( f1_sqrt(4) )
print( f2_fraction(2) )
```

Exercise 1

Define a Python function for each of the following functions:

(a) $f(x) = \sqrt{x}$

(b) $f(x) = \sqrt[3]{x}$

(c) $f(x) = x^{\frac{2}{3}}$

(d) $f(x) = \frac{x^3}{3} - \frac{x^2}{2} - 2x + \frac{1}{3}$

(e) $f(x) = \frac{2x^2 - 3}{7x + 4}$

(f) $f(x) = \frac{5x^2 + 8x - 3}{3x^2 + 2}$

(g) $f(x) = \sin(x)$

(h) $f(x) = \cos(x)$

(i) $f(x) = 3^x$

(j) $f(x) = 10^{-x}$

(k) $f(x) = e^x$

(l) $f(x) = \log_2(x)$

(m) $f(x) = \log_{10}(x)$

(n) $f(x) = \ln(x)$

Exercise 2

Define a Python function for each of the following functions, and then find the range (minimum and maximum values) of the functions:

(a) $f(x) = 2 + \frac{x^2}{x^2 + 4}$ with $x \in [-2, 2]$

(b) $f(x) = \sqrt{5x + 10}$ with $x \in [0, 5]$

(c) $f(x) = \frac{2}{x^2 - 16}$ with $x \in [5, 10]$

(d) $f(x) = x^4 + 3x^2 - 1$ with $x \in [-3, 3]$

(e) $f(x) = \begin{cases} x, & x \geq 0 \\ -x, & x < 0 \end{cases}$ with $x \in [-3, 3]$

Hint:

```
import math
import numpy as np

def fx_2e(x):
    if x >= 0:
        return x
    else:
        return -x

def cau2e():
    for x in np.arange(-3, 3.1, 0.1):
        #find minimum and maximum values of fx_2e
        print( round (fx_2e(x), 5) )
        # ...

#main
cau2e()
```

4. Python Anonymous/Lambda Function

In Python, an anonymous function is a function that is defined [without a name](#).

While normal functions are defined using the [def](#) keyword in Python, anonymous functions are defined using the [lambda](#) keyword. Hence, anonymous functions are also called lambda functions.

- Syntax:

```
lambda arguments: expression
```

Lambda functions can have any number of arguments but only one expression. The expression is evaluated and returned.

```
f1_double = lambda x: x * 2  
print( f1_double(5) )
```

In the above program, `x` is the argument and `x * 2` is the expression that gets evaluated and returned. This function returns a function object which is assigned to the identifier `f1_double`.

The statement

```
f1_double = lambda x: x * 2
```

is nearly the same as:

```
def f1_double(x):  
  
    return x * 2
```

- Other examples:

```
f2 = lambda a, b: (a + b) / (a - b)  
print( f2(5, 1) )  
  
fx = lambda x: x**4 + 3*x**2 - 1  
print( fx(1) )
```

- Use of Lambda Function in python

Lambda functions are used along with built-in functions like `filter()`, `map()` etc.

The `map()` function in Python takes in a function and a list. The function is called with all the items in the list and a new list is returned which contains items returned by that function for each item.

Here is an example use of `map()` function to double all the items in a list.

```
my_list = [1, 5, 4, 6, 8, 11, 3, 12]  
f1_double = lambda x: x * 2  
  
new_list = list( map(f1_double , my_list) )  
print(new_list)
```

Another example, calculate y-values for the function $f = x^2$

```
import numpy as np

f = lambda x: x**2

x = np.arange(-3, 3.1, 1)
y = list( map(f, x) )

print(x)
print(y)
```

For more usages, visit the websites:

- https://www.w3schools.com/python/python_lambda.asp
- <https://www.programiz.com/python-programming/anonymous-function>

Exercise 3

Write a computer program to compute the composites of function. Meanwhile,

$$f_1 = x + 5 \text{ and } f_2 = x^2 - 3.$$

(a) $f_1(f_2(0))$

(c) $f_1(f_1(-5))$

(b) $f_2(f_1(0))$

(d) $f_2(f_2(2))$

5. Python Matplotlib

Matplotlib is a low level graph plotting library in python that serves as a visualization utility.

5.1 Installation of Matplotlib

If you have Python and PIP already installed on a system, then install it using this command:

```
C:\Users\Your Name>pip install matplotlib
```

```
python -m pip install matplotlib
```

If this command fails, then use a python distribution that already has Matplotlib installed, like Anaconda, Spyder etc.

5.2 Matplotlib Pyplot

Most Matplotlib utilities lies under the **pyplot** submodule, and are usually imported under the **plt** alias:

```
import matplotlib.pyplot as plt
```

Now the Pyplot package can be referred to as **plt**.

Plotting x and y points:

The **plot()** function is used to draw points (markers) in a diagram.

By default, the **plot()** function draws a line from point to point.

The function takes parameters for specifying points in the diagram:

- Parameter 1 is an array containing the points on the **x-axis**.
- Parameter 2 is an array containing the points on the **y-axis**.

For ex., plot a line from (1, 3) to (8, 10), we pass two arrays [1, 8] and [3, 10] to the plot function:

```
import matplotlib.pyplot as plt
import numpy as np

xpoints = np.array([1, 8])
ypoints = np.array([3, 10])

plt.plot(xpoints, ypoints)
plt.show()
```

Draw a line in a diagram from position (1, 3) to (2, 8) then to (6, 1) and finally to position (8, 10):

```
import matplotlib.pyplot as plt
import numpy as np

xpoints = np.array([1, 2, 6, 8])
ypoints = np.array([3, 8, 1, 10])

plt.plot(xpoints, ypoints)
plt.show()
```

For more usages, visit the website: https://www.w3schools.com/python/matplotlib_intro.asp

- **Practice examples:**

Plot the function $f(x) = -x^3$ and specify the intervals over which the function is increasing and the intervals where it is decreasing.

```
import matplotlib.pyplot as plt
import numpy as np

fx_4i = lambda x: -x**3
#def fx_4i(x):
#    return -x**3

#main
x_array = np.arange(-10, 10.1, 0.1)
y_array = list( map(fx_4i, x_array) ) #y = f(x)

plt.plot(x_array, y_array, color='red')
plt.grid()
plt.show()
print("4i, f(x) is decreasing as x-values belong to  $(-\infty, +\infty)$ ")
```

Exercise 4

Write a Python program to plot each of the following functions and specify the intervals over which the function is increasing and the intervals where it is decreasing:

(i) $f(x) = -x^3$	(k) $f(x) = -\frac{1}{x}$	(m) $f(x) = \sqrt{ x }$
(j) $f(x) = -\frac{1}{x^2}$	(l) $f(x) = \frac{1}{ x }$	(n) $f(x) = \sqrt{ -x }$

Exercise 5

Write a Python program to plot the functions:

$$f_1(x) = \sqrt{1 - (|x| - 1)^2}, f_2(x) = -3\sqrt{1 - \sqrt{\frac{|x|}{2}}}.$$

f_1 is drawn with magenta color, f_2 is drawn with red color, and $x \in [-2, 2]$.

Guide link: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.plot.html

Exercise 6

Write a Python program to plot the original and shifted graphs together, labeling each graph with its equation in these following cases:

- | | |
|--|--|
| (a) $f(x) = x^2 + k$, for $k = 2, 4, 6, 8, 10, 12$,
and $x \in [-10, 10]$. | (g) $f(x) = \frac{1}{x^2}$, left 2, down 1. |
| (b) $f(x + k) = (x + k)^2$, for $k = 2, 4, 6, 8, 10, 12$,
and $x \in [-10, 10]$. | (h) $f(x) = 1 - x^3$, stretched horizontally
by a factor of 2. |
| (c) $f(x) = k\sqrt{x}$, for $k = \frac{1}{3}, 1, 3, 6$, and $x \in [1, 50]$. | (i) $f(x) = \sqrt{x + 1}$, compressed horizontally
by a factor of 4. |
| (d) $f(x) = x^3$, left 1, down 1. | (j) $f(x) = \sqrt{x + 1}$, stretched vertically
by a factor of 3. |
| (e) $f(x) = x^{2/3}$, right 1, down 1. | |
| (f) $f(x) = \frac{1}{2}(x + 1) + 5$, down 5, right 1. | |

6b is not correct, should be $f(x) = (x + k)^2$

Hint:

#6a

```
def f6_a(x, k):
    return ...

def cau6a():
    k = np.arange(...)
    x = np.arange(...)

    for ki in k:
        y = []
        for xi in x:
            y.append(...)

    plt.plot(x, y, label = "k=" + str(ki) )

    plt.title("Cau 6a")
    plt.legend()
    plt.show()
```

```
#6h: f(x) -->> g(x) = f(x/2)
#6i: f(x) -->> g(x) = f(4x)
#6j: f(x) -->> g(x) = 3 * f(x)
```

Exercise 7

Write a Python program to check whether the function $f(x)$ is one-to-one function or not:

$$(a) \ f(x) = x^3 - \frac{x}{2}$$

$$(b) \ f(x) = x^2 + \frac{x}{2}$$

Guide link: <https://www.cuemath.com/algebra/one-to-one-function/>

6. References

- Python Tutorial on the W3schools website: <https://www.w3schools.com/python/default.asp>
- Python Tutorial on the Tutorials Point website:
<https://www.tutorialspoint.com/python/index.htm>

-- THE END --