100 DAYS 100 PYTHON PROBLEMS

[Day 4: Functions and Modules]

FUNCTIONS :-

- A function is a block of code that performs a specific task or set of tasks. It can take input, process it, and return an output.
- Functions are defined using the **'def'** keyword in most programming languages.
- They have a name, a parameter list, and a body where the actual code is written.
- You can call a function by its name to execute the code within it.

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

my_function()
```

Arguments :-

- Information can be passed into functions as arguments.
- 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")
```

- If you do not know how many arguments that will be passed into your function, add a ***** before the parameter name in the function definition.
- Python also accepts function recursion, which means a defined function can call itself.

100 DAYS 100 PYTHON PROBLEMS

[Day 4: Functions and Modules]

- This way the function will receive a tuple of arguments, and can access the items accordingly.

```
def my_function(*kids):
  print("The youngest child is " + kids[2])

my_function("Emil", "Tobias", "Linus")
```

Lambda :-

- A lambda function is a small anonymous function. It can take any number of arguments, but can only have one expression.
- Syntax : `lambda arguments : expression`
- The power of lambda is better shown when you use them as an anonymous function inside another function.

```
def myfunc(n):
  return lambda a : a * n

mydoubler = myfunc(2)

print(mydoubler(11))
```

100 DAYS 100 PYTHON PROBLEMS

[Day 4: Functions and Modules]

MODULES :-

- A file containing a set of functions you want to include in your application.
- Write a code of any function, save it in your folder with some name like filename.py
- You can use this module in new python files with **'import'** keyword, like ***'import filename'***.
- Also you can use **'as'** keyword to rename your module.

PRACTICE QUESTIONS :-

1. Write a Python function that takes two numbers as input and returns their sum.
2. Define a function that calculates the factorial of a given positive integer using a loop.
3. Create a Python module called ***math_operations*** with functions for addition, subtraction, multiplication, and division. Import this module and use it to perform these operations.
4. Write a Python function that takes a list as a parameter and returns the maximum value in the list.
5. Define a function that generates a Fibonacci sequence of a given length using recursion.
6. Create a Python module named **utils** that contains a function to check if a number is prime or not. Import this module and use the function to determine if a given number is prime.
7. Write a function that takes a string as input and returns the string reversed.