Tutorial 10

Date: Feb 28, 2021

Email: mohdakram.ansari@ucalgary.ca

Agenda

1. Commenting best practices
2. Functions
3. Main function
4. Exercise

1. Commenting Best Practices

To write a comment in Python, simply put the hash mark # before your desired comment:

```
# This is a comment
```

Python ignores everything after the hash mark and up to the end of the line.
You can insert them anywhere in your code, even inline with other code:

```
print("This will run.")  # This won't run
```

Tip: Comments should be short, sweet, and to the point.

Multiline comments

If you don't want to put a `#` at the start of each line in a multiline comment,
you can enclose your comment lines in tripple quote (single or double).

```
"""
If I really hate pressing `enter` and
typing all those hash marks, I could
just do this instead
"""
```

Indented Comments

Have your comments indented at the same level as your code

```
# Do This
if grade == 4:
    # Perfect grade
    print("Congratulation!")

# Do Not This
if grade == 4:
# Perfect grade
    print("Congratulation!")
```

Commenting Shortcut in PyCharm

Clicking each and every line to comment it out could take a lot of time!
In these cases, you'll want to toggle comments instead.
Simply select the desired code and press `Ctrl` + `/` on PC, or `Cmd` + `/` on Mac:

---

2. Functions

What is a function in Python?

In Python, a function is a group of related statements that perform a specific task.

Syntax of Function

```
def function_name(parameters):
    """docstring"""
    statement(s)...
```

Example:

```
def greet(name):
    """
    This function greets to
    the person passed in as
    a parameter
    """
    print("Hello, " + name + ". Good morning!")
```

How to call a function in python?

Once we have defined a function, we can call it from another function, program,
or even the Python prompt.
To call a function we simply type the function name with appropriate parameters.

Example:

```
def greet(name):
    """
    This function greets to
    the person passed in as
    a parameter
    """
    print("Hello, " + name + ". Good morning!")

# This is how you call a function
greet('Paul')
```

The `return` Statement

The return statement is used to exit a function and go back to the place from where it was called.

Syntax:

```
return [expression_list]
```

Example:

```
def compound_interest(principal, rate, time):
    """This function will return the compound interest on the principal amount"""
    amount = principal * (pow((1 + rate / 100), time))
    interest = amount - principal
    return interest

amount = float(input("Enter the invested amount"))
rate = 14
years = 8
ci = compound_interest(amount, rate, years)
print("{0} will yield an interest of {1:.2f} in {2} years".format(amount, ci, years))
```

We can also return more than 1 value using the return statement

Example:

```
def add(num1, num2):
    total  = num1 + num2
    return 'SUM', total, [num1, num2]

a, b, c = add(5, 6)

print(a)
# SUM
print(b)
# 11
print(c)
# [5, 6]
```

You can also have multiple return statements in your code

Example:

```
def absolute_value(num):
    """This function returns the absolute
    value of the entered number"""
    if num >= 0:
        return num
    else:
        return -num

print(absolute_value(2))
print(absolute_value(-4))
```

Exercise

Convert the program to print all prime numbers to use the function.
Create the function to check if a number is prime or not if needed.

Starting code: `tut_09/example_for_all_primes.py`

---

3. Main Function

It is a best practice in python to start your program from the main function.

Example:

```
def add(a, b):
    return a+b

def main():
    print("Total of 2 and 3 is", add(2,3))

main()
```

4. Exercise

Do not use any inbuilt function

Define a function that accepts 3 numbers are arguments and returns the maximum of the three numbers. Test your function using combination of different values from the main function.