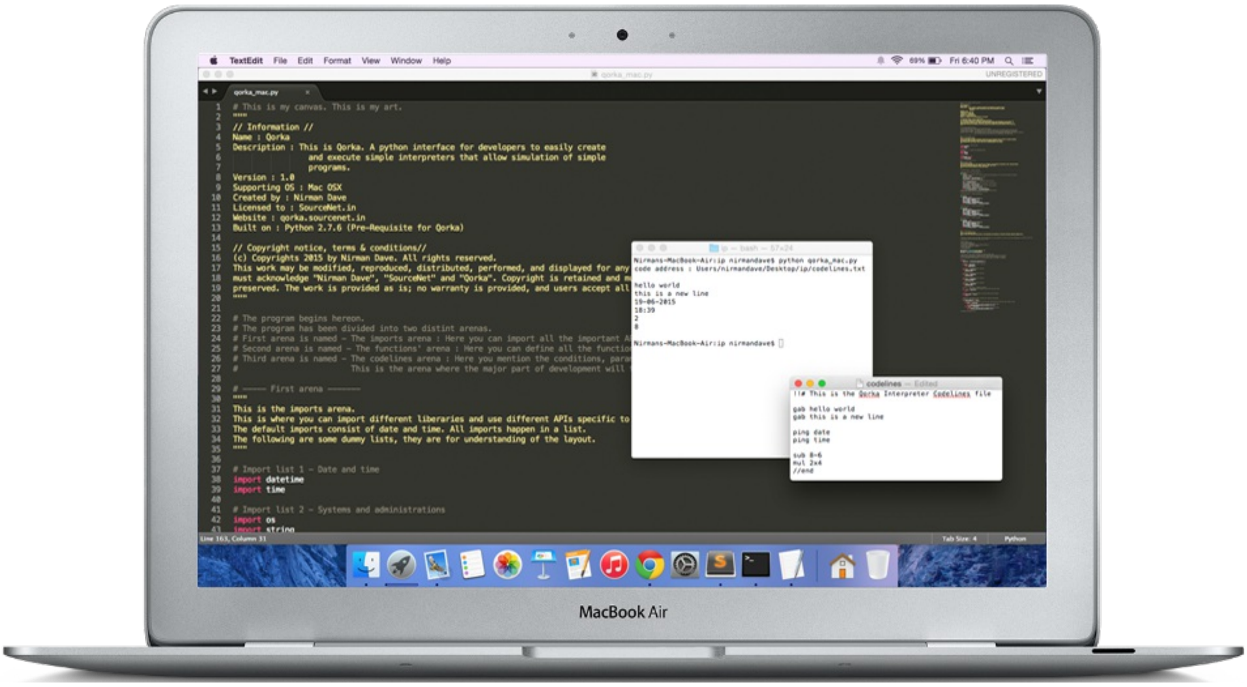
Qorka



```
# This is my canvas. This is my art.
"""
// Information //
Name : Qorka
Description : This is Qorka. A python interface for developers to easily create
              and execute simple interpreters that allow simulation of simple
              programs.
Version : 1.0
Supporting OS : Mac OSX
Created by : Nirman Dave
Licensed to : SourceNet.in
Website : qorka.sourcenet.in
Built on : Python 2.7.6 (Pre-Requisite for Qorka)

// Copyright notice, terms & conditions//
(c) Copyrights 2015 by Nirman Dave. All rights reserved.
This work may be modified, reproduced, distributed, performed, and displayed for any
must acknowledge "Nirman Dave", "SourceNet" and "Qorka". Copyright is retained and m
preserved. The work is provided as is; no warranty is provided, and users accept all
"""

# The program begins hereon.
# The program has been divided into two distint arenas.
# First arena is named - The imports arena : Here you can import all the important A
# Second arena is named - The functions' arena : Here you can define all the functio
# Third arena is named - The codelines arena : Here you mention the conditions, para
#                         This is the arena where the major part of development will

# ----- First arena ------
"""
This is the imports arena.
This is where you can import different liberaries and use different APIs specific to
The default imports consist of date and time. All imports happen in a list.
The following are some dummy lists, they are for understanding of the layout.
"""

# Import list 1 - Date and time
import datetime
import time

# Import list 2 - Systems and administrations
import os
import string
```

```
Nirmans-MacBook-Air:ip nirmandave$ python qorka_mac.py
code address : Users/nirmandave/Desktop/ip/codelines.txt

hello world
this is a new line
19-06-2015
18:39
2
8

Nirmans-MacBook-Air:ip nirmandave$ 
```

```
!!# This is the Qorka Interpreter Codelines file

gab hello world
gab this is a new line

ping date
ping time

sub 8-6
mul 2x4
//end
```

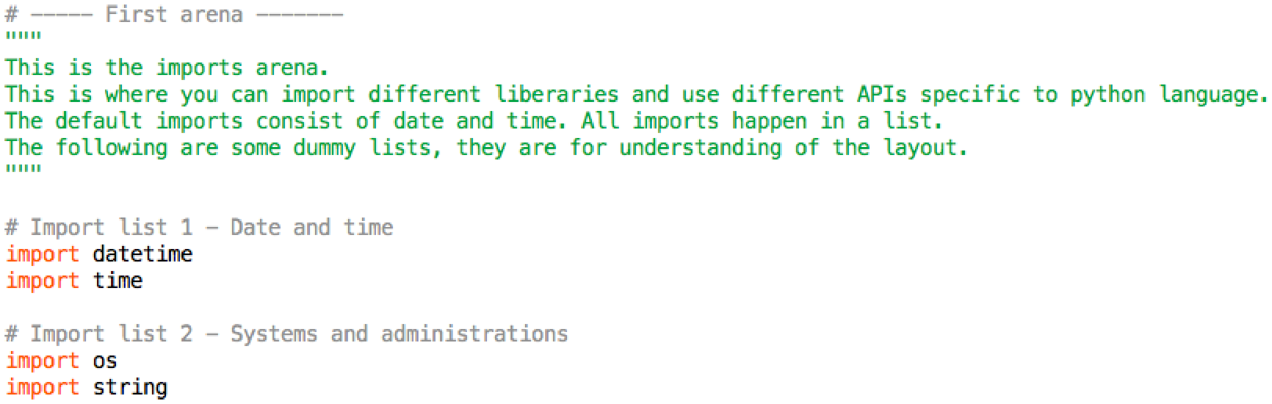
Ever wondered how awesome it would be if you could quickly assemble your own programming language (code interpreter) in minutes? or a program that allows you to code in Hindi, Spanish, French or any other language you ever wanted?

Qorka is an interpreter development platform from SourceNet which allows you to creatively put together your own programming interpreter with ease. This means that I can create my own programming syntax, by defining each function on the Qorka platform. And everytime I write a program with my own programming syntax (programming language) in a text editor and run it on Qorka, it will execute the file code according to the functions defined by you.

Qorka : Environment

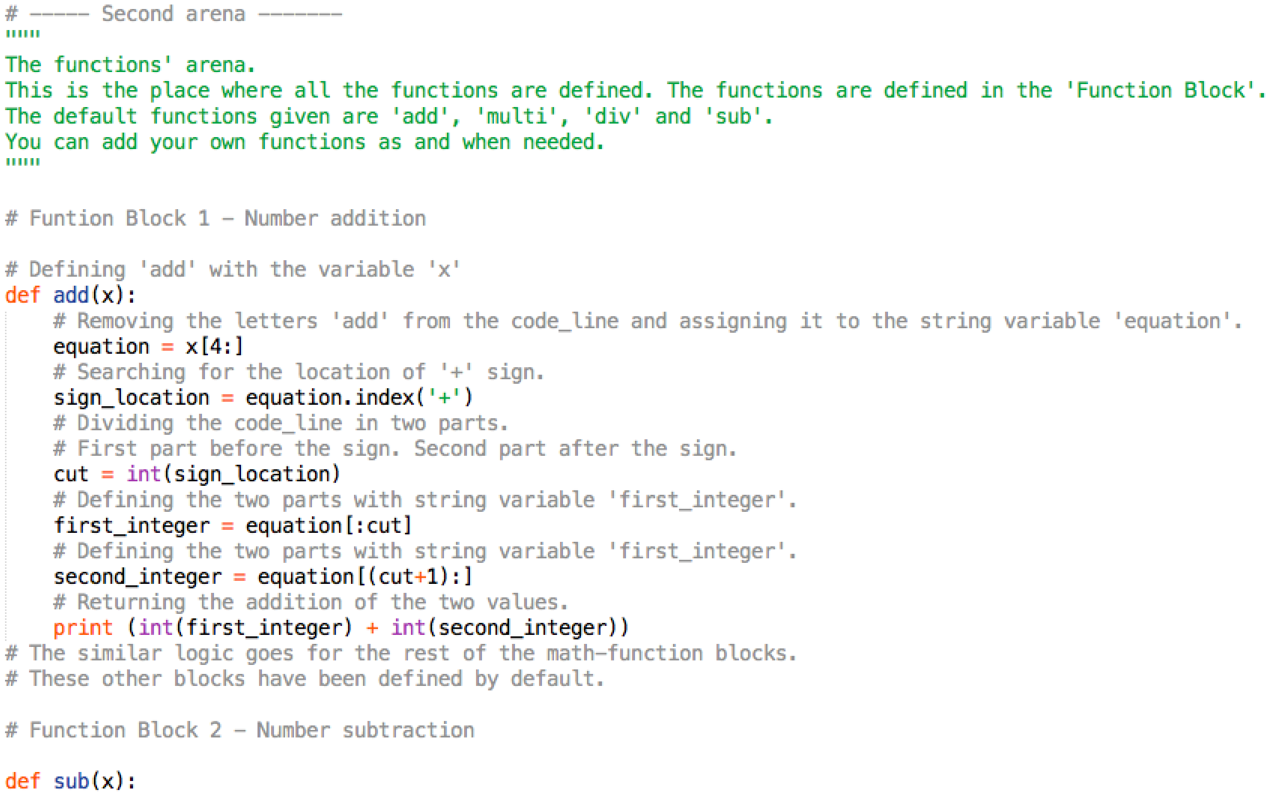
QORKA
interpreter canvas for Python
qorka.sourcenet.in

Qorka is completely written in Python and requires knowledge in Python programming to operate. To get started, first open the Qorka code file, "qorka_mac.py" for Mac and "qorka_pc.py" for Windows. You will see the code file is divided into three distinct parts called the "Arenas".



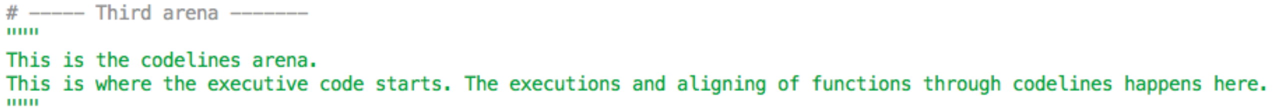
```
# ----- First arena -------
"""
This is the imports arena.
This is where you can import different liberaries and use different APIs specific to python language.
The default imports consist of date and time. All imports happen in a list.
The following are some dummy lists, they are for understanding of the layout.
"""

# Import list 1 - Date and time
import datetime
import time

# Import list 2 - Systems and administrations
import os
import string
```

The first arena is where you can import different libraries and use different APIs specific to python language. The default imports consist of date and time. All imports happen in a list. The following are some dummy lists, they are for understanding of the layout.



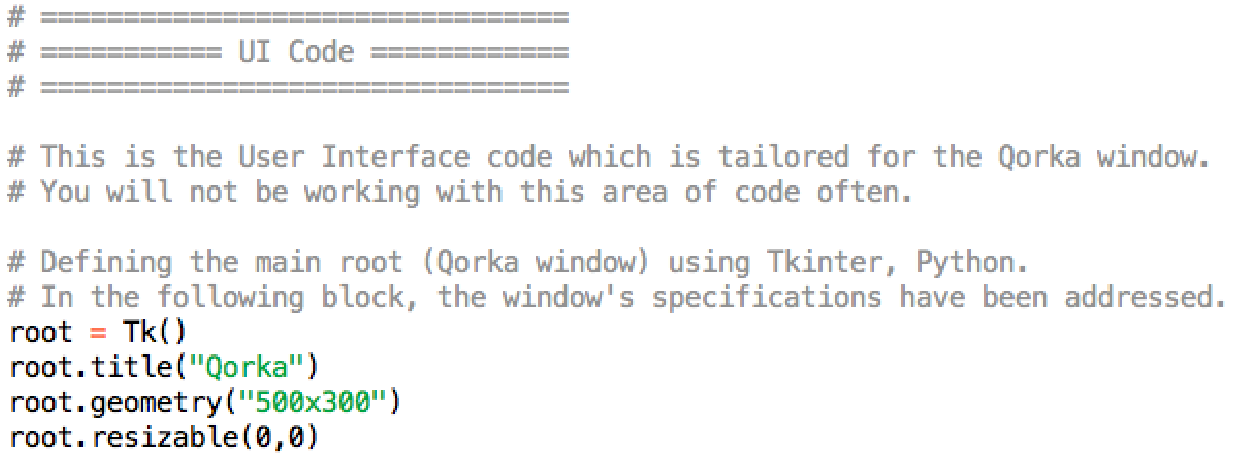
```
# ----- Second arena -------
"""
The functions' arena.
This is the place where all the functions are defined. The functions are defined in the 'Function Block'.
The default functions given are 'add', 'multi', 'div' and 'sub'.
You can add your own functions as and when needed.
"""

# Funtion Block 1 - Number addition

# Defining 'add' with the variable 'x'
def add(x):
    # Removing the letters 'add' from the code_line and assigning it to the string variable 'equation'.
    equation = x[4:]
    # Searching for the location of '+' sign.
    sign_location = equation.index('+')
    # Dividing the code_line in two parts.
    # First part before the sign. Second part after the sign.
    cut = int(sign_location)
    # Defining the two parts with string variable 'first_integer'.
    first_integer = equation[:cut]
    # Defining the two parts with string variable 'first_integer'.
    second_integer = equation[(cut+1):]
    # Returning the addition of the two values.
    print (int(first_integer) + int(second_integer))
# The similar logic goes for the rest of the math-function blocks.
# These other blocks have been defined by default.

# Function Block 2 - Number subtraction

def sub(x):
```

The second arena is the place where all the functions are defined. The functions are defined in the 'Function Block'. The default functions given are 'add', 'multi', 'div' and 'sub'. You can add your own functions while creating your own programming language.



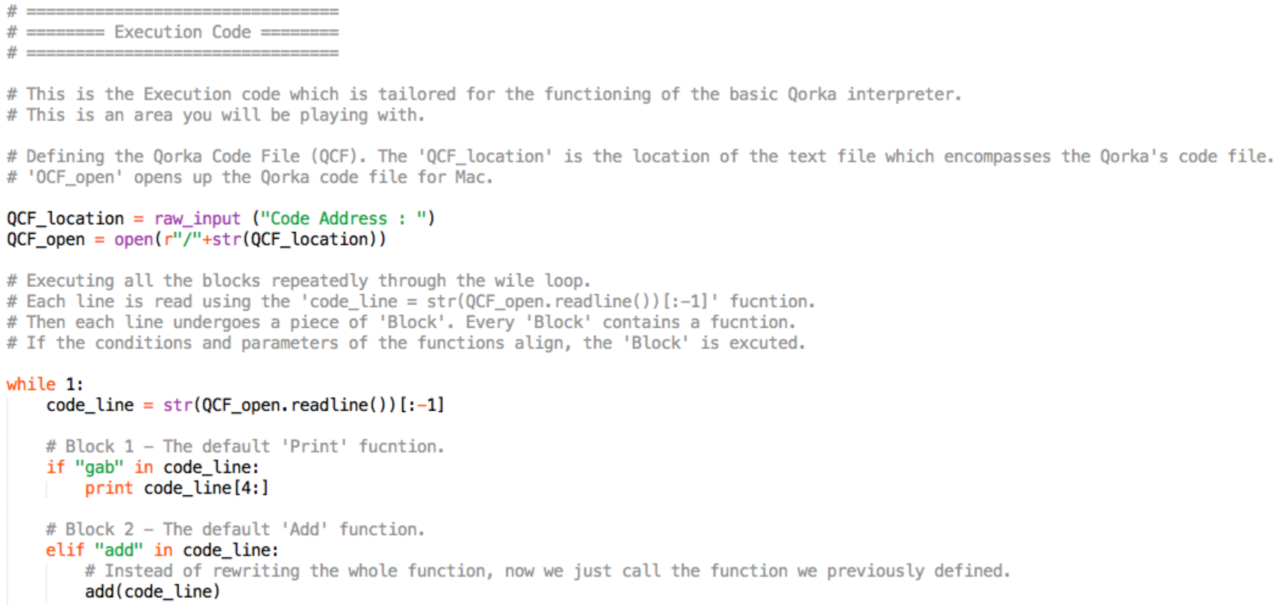
```
# ----- Third arena -------
"""
This is the codelines arena.
This is where the executive code starts. The executions and aligning of functions through codelines happens here.
"""
```

The third arena is made of two parts. The UI code (which we do not need to alter) and The Execution code. The UI code consists of the UI that you see the first time you open Qorka.



```
# ================================
# =========== UI Code ===========
# ================================

# This is the User Interface code which is tailored for the Qorka window.
# You will not be working with this area of code often.

# Defining the main root (Qorka window) using Tkinter, Python.
# In the following block, the window's specifications have been addressed.
root = Tk()
root.title("Qorka")
root.geometry("500x300")
root.resizable(0,0)
```

The Execution code is the main playground. It is tailored to function as the basic Qorka interpreter.



```
# ================================
# ======== Execution Code ========
# ================================

# This is the Execution code which is tailored for the functioning of the basic Qorka interpreter.
# This is an area you will be playing with.

# Defining the Qorka Code File (QCF). The 'QCF_location' is the location of the text file which encompasses the Qorka's code file.
# 'OCF_open' opens up the Qorka code file for Mac.

QCF_location = raw_input ("Code Address : ")
QCF_open = open(r"/"+str(QCF_location))

# Executing all the blocks repeatedly through the wile loop.
# Each line is read using the 'code_line = str(QCF_open.readline())[:-1]' fucntion.
# Then each line undergoes a piece of 'Block'. Every 'Block' contains a fucntion.
# If the conditions and parameters of the functions align, the 'Block' is excuted.

while 1:
    code_line = str(QCF_open.readline())[:-1]

    # Block 1 - The default 'Print' fucntion.
    if "gab" in code_line:
        print code_line[4:]

    # Block 2 - The default 'Add' function.
    elif "add" in code_line:
        # Instead of rewriting the whole function, now we just call the function we previously defined.
        add(code_line)
```

Qorka : Getting started

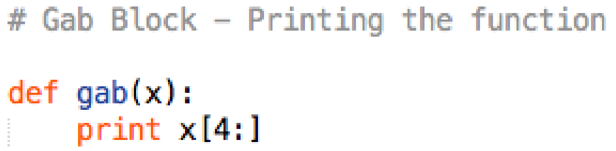
Lets say you want to build a programming language (code interpreter), the first thing you need to add is a "print" function.

Playing : First Arena

For this simple operation we do not need to import any modules into the first arena. So we leave the first arena as is.

Explaining : Second arena

Jumping to the second arena, we now need to explain Qorka about what a print function is and what it will do. So we quickly define a print function. I will name my function as gab! (This means each time I enter "gab Hello, World." the program will print the text after gab.



```
# Gab Block - Printing the function

def gab(x):
    print x[4:]
```

Executing : Third arena

Here you need to define exactly when do you want the function to execute. We want the function to execute each time the word "gab" is found in the codeline.

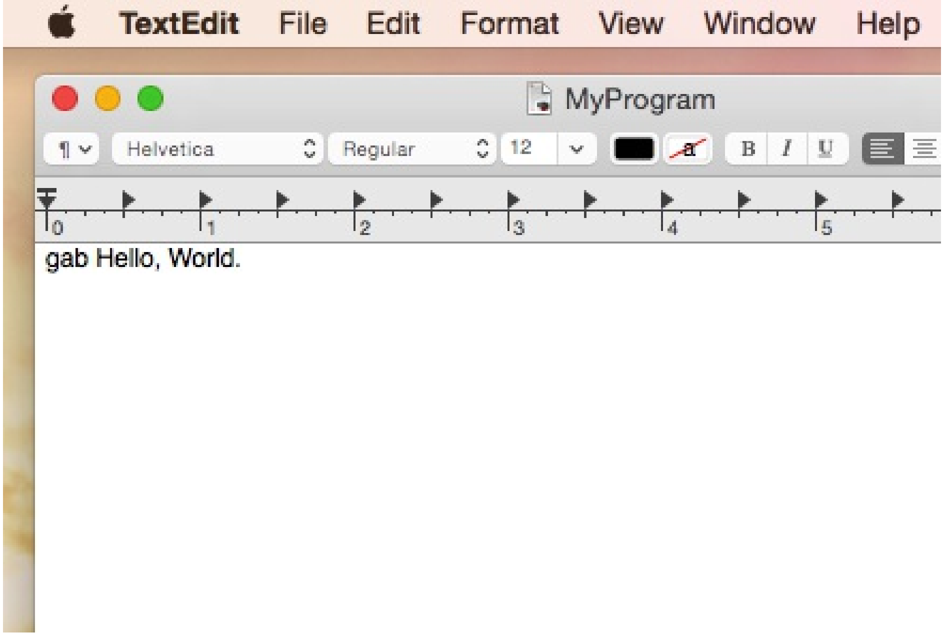
```
# Block 1 - The default 'Print' fucntion.
if "gab" in code_line:
    gab(code_line)
```

Saving your Qorka program

Once you're done with all the three arenas, save the code as "qorka_mac.py" to your desktop.

Operating in your programming language

By saving the Qorka file, you have now created your programming language (code interpreter). Now let us write a simple "Hello, World." program using your programming language.



gab Hello, World.

Open up a text editor, and write down "gab Hello, World". Now save the file as "MyProgram.txt" on your Desktop.

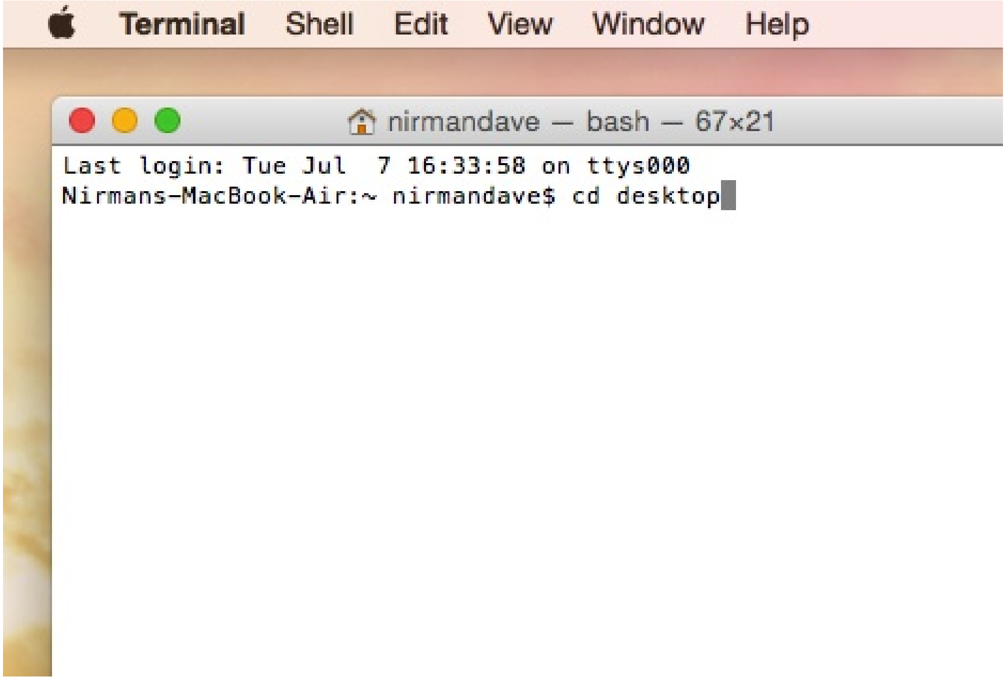
Executing the code

Congratulations you have successfully created your own programming language (code interpreter) and have also written down your first Hello, World. code in your own programming language (code interpreter). Now we just need to run it, and see how things go! To do so, first make sure you have two files posted on your desktop.

- MyProgram.txt
- qorka_mac.py or qorka_pc.py

Now open up the terminal on Mac or command prompt on Windows, and change the directory to Desktop. Use the following command to do so:

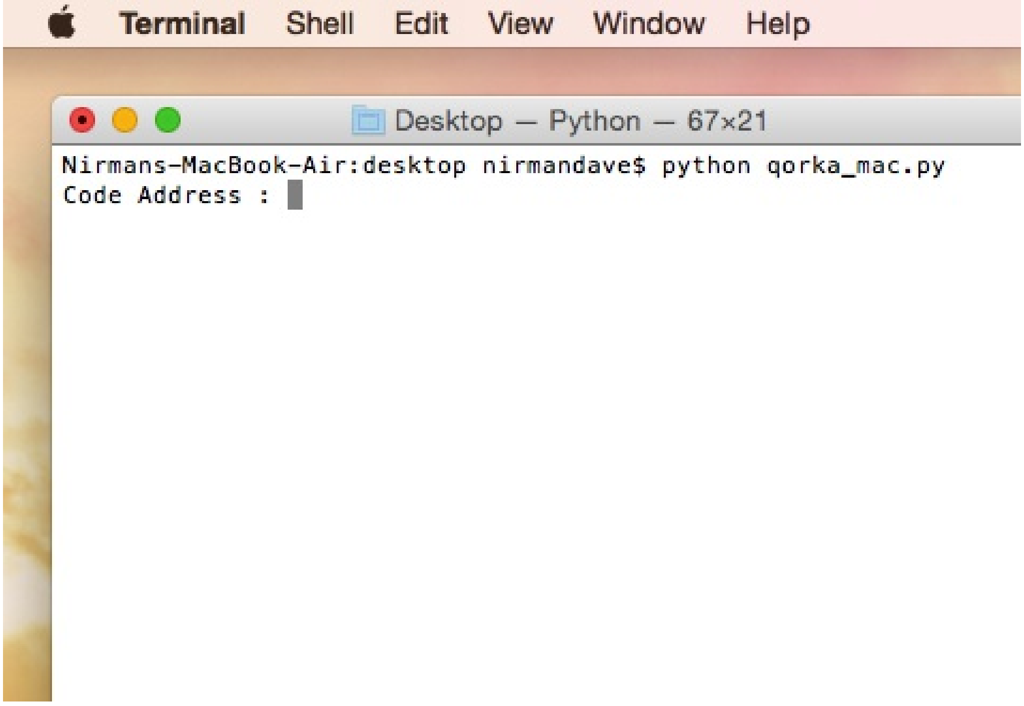
cd desktop



```
Last login: Tue Jul  7 16:33:58 on ttys000
Nirmans-MacBook-Air:~ nirmandave$ cd desktop
```

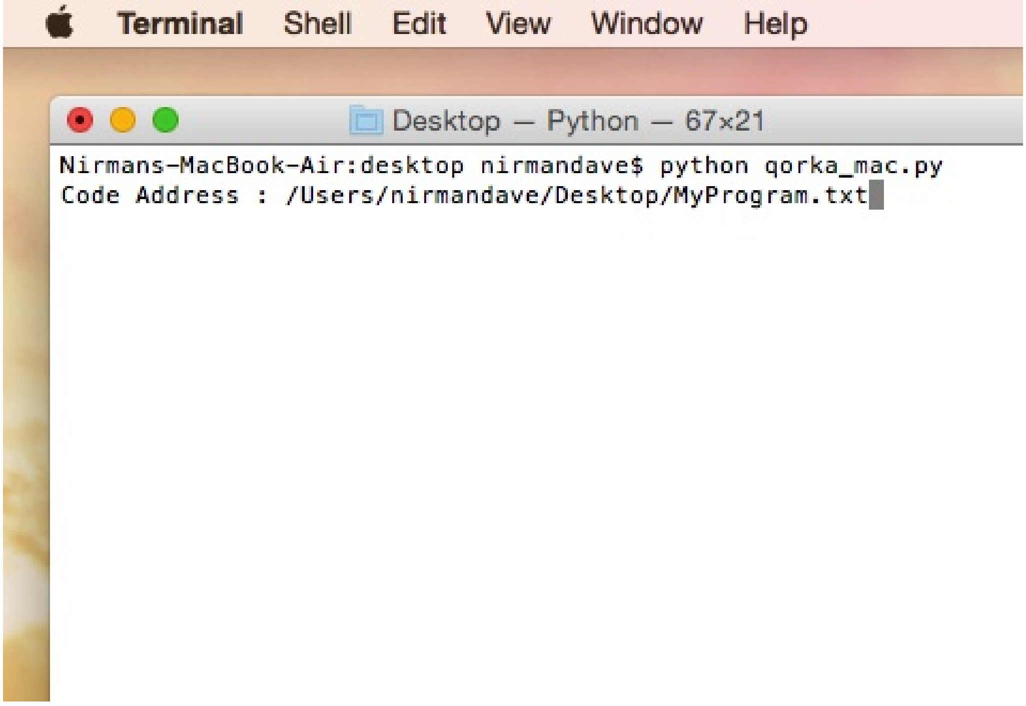
Now, run the Qorka python file. To do so, input the following command:

Python qorka_mac.py



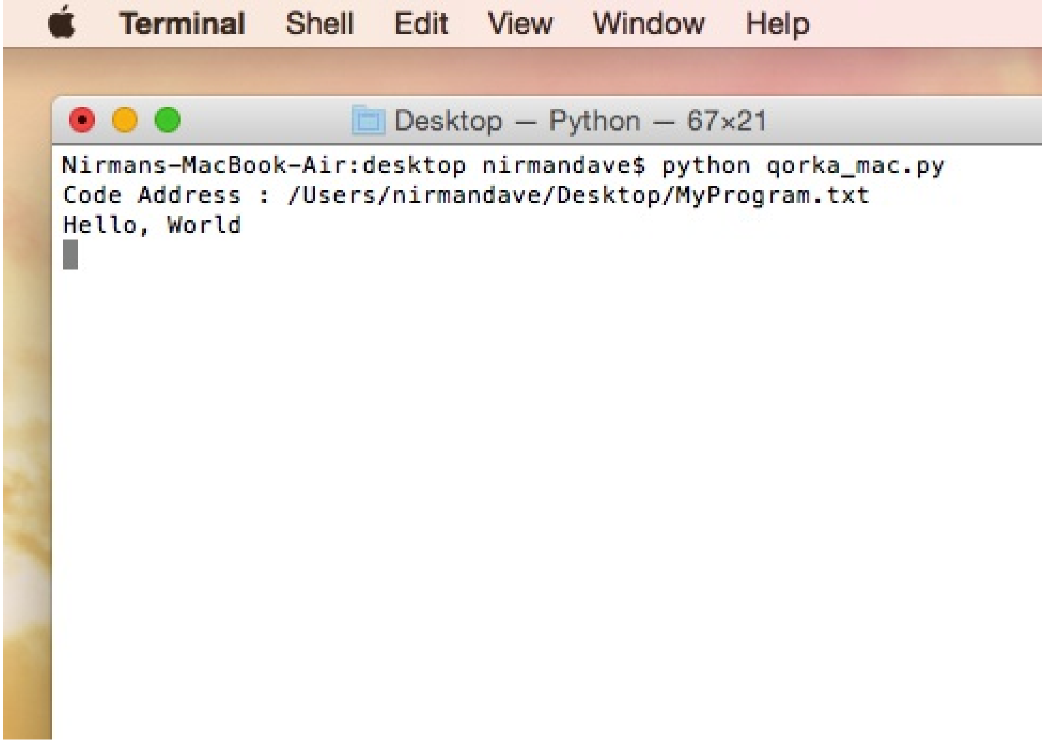
```
Nirmans-MacBook-Air:desktop nirmandave$ python qorka_mac.py
Code Address :
```

Once you hit enter, Qorka will ask you for the location of your code (that is MyProgram.txt).



```
Nirmans-MacBook-Air:desktop nirmandave$ python qorka_mac.py
Code Address : /Users/nirmandave/Desktop/MyProgram.txt
```

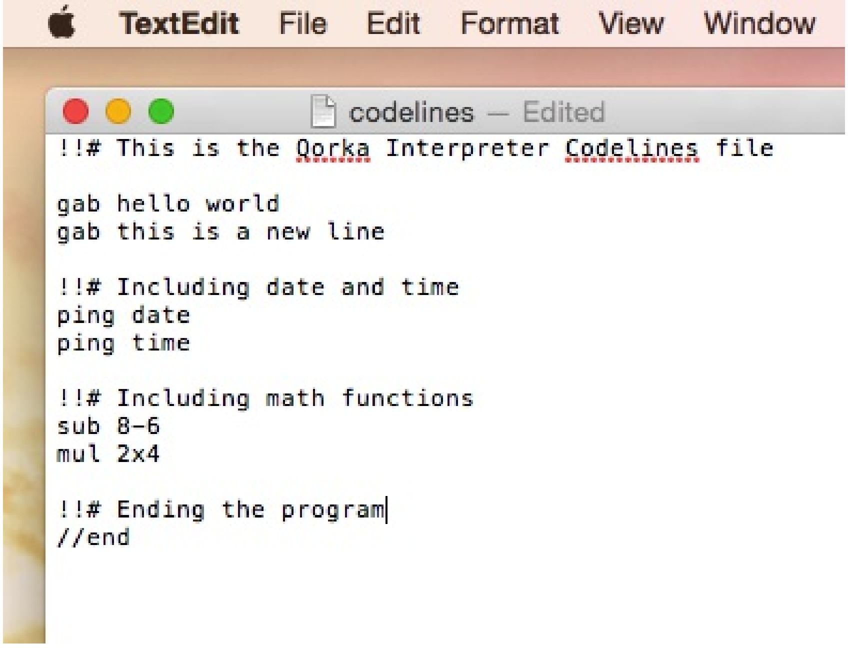
Once you enter the MyProgram.txt file location, your code will run on the programming language (code interpreter) you have created. And you will see the output at follows.



```
Nirmans-MacBook-Air:desktop nirmandave$ python qorka_mac.py
Code Address : /Users/nirmandave/Desktop/MyProgram.txt
Hello, World
```

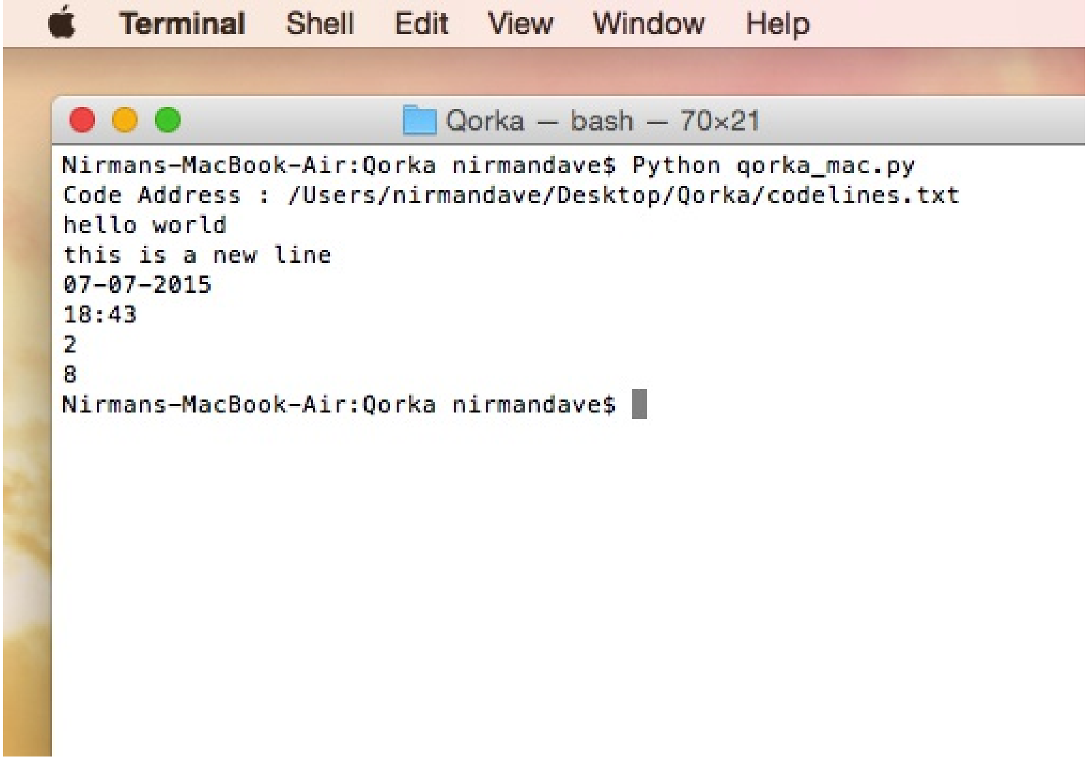
What's more?

This way you can create n number of functions and define n number of operations to create a language that does more than just saying "hello". Check this out, my language also incorporates math functions, commenting facility, and much much more. Check this out, here is my new set of code lines.



```
!!# This is the Qorka Interpreter Codelines file

gab hello world
gab this is a new line

!!# Including date and time
ping date
ping time

!!# Including math functions
sub 8-6
mul 2x4

!!# Ending the program
//end
```

And here is how it functions!



```
Nirmans-MacBook-Air:Qorka nirmandave$ Python qorka_mac.py
Code Address : /Users/nirmandave/Desktop/Qorka/codelines.txt
hello world
this is a new line
07-07-2015
18:43
2
8
Nirmans-MacBook-Air:Qorka nirmandave$
```

Qorka : Program information

Here is some quick information about Qorka and its build.

- Program name : Qorka (http://qorka.sourcenet.in)
- Program description : This is Qorka. A python interface for developers to easily create and execute simple interpreters that allow simulation of simple programs.
- Program version : 1.0

- Supporting OS : Mac OSX & Windows
- Created by : Nirman Dave (http://www.nirmandave.com)
- Licensed to : SourceNet.in (http://www.sourcenet.in)
- Website : qorka.sourcenet.in (http://qorka.sourcenet.in)
- Built on : Python 2.7.6 (Pre-Requisite for Qorka)

Qorka : Copyright notice, terms & conditions