



AL NAFI,  
A company with a focus on education,  
wellbeing and renewable energy.

# Python Alpha 000

An intro to Python Alpha course coverage

# Dua of the day

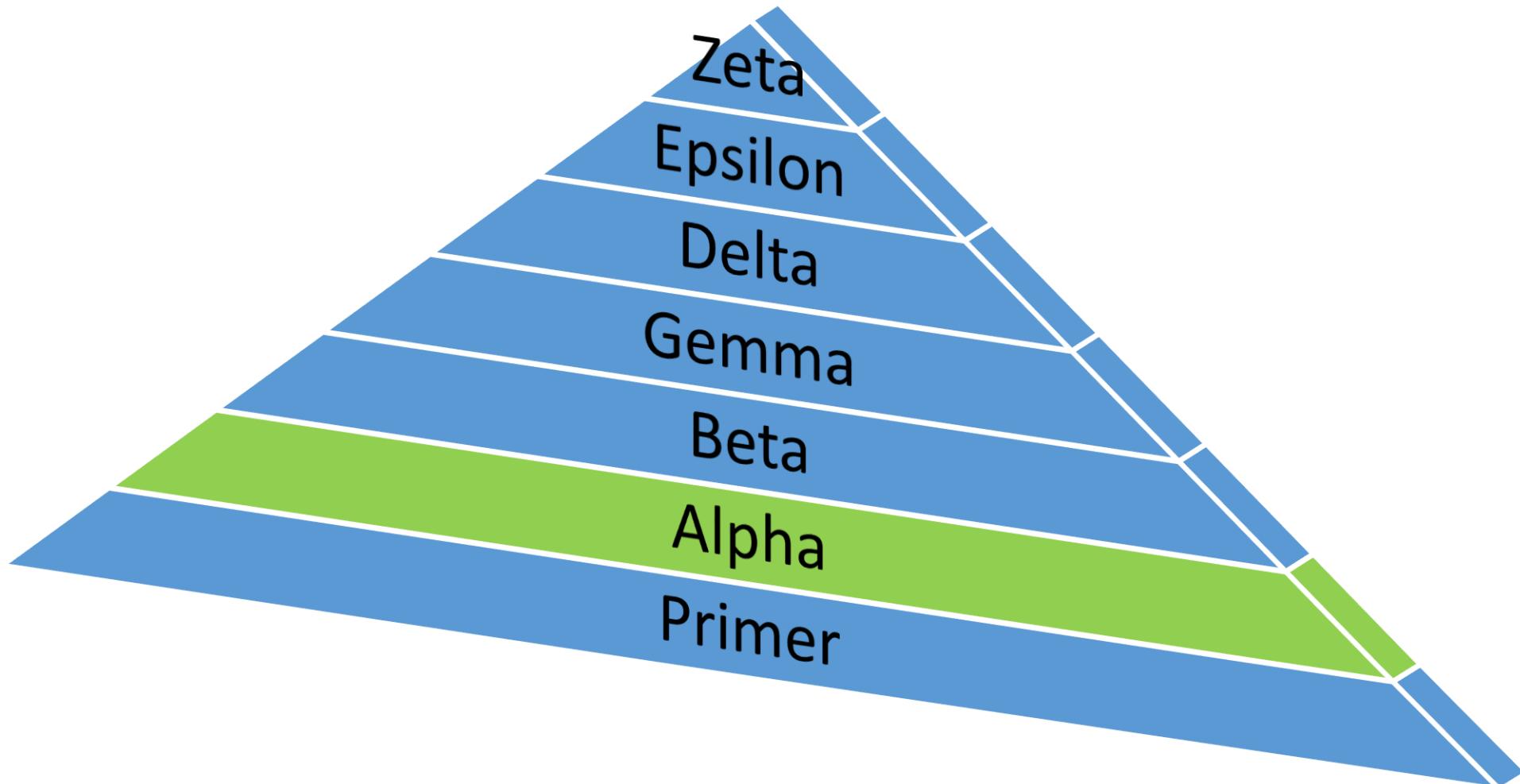
إِلَهِي غَارَتِ النُّجُومُ، وَنَامَتِ الْعَيْنُونُ وَغَلَقَتِ الْمُلُوكُ أَبْوَابَهَا، وَبَابُكَ مَفْتُوحٌ، وَخَلَى كُلُّ حَبِيبٍ بِحَبِيبِهِ، وَهَذَا مَقَامِي بَيْنَ يَدَيْكَ

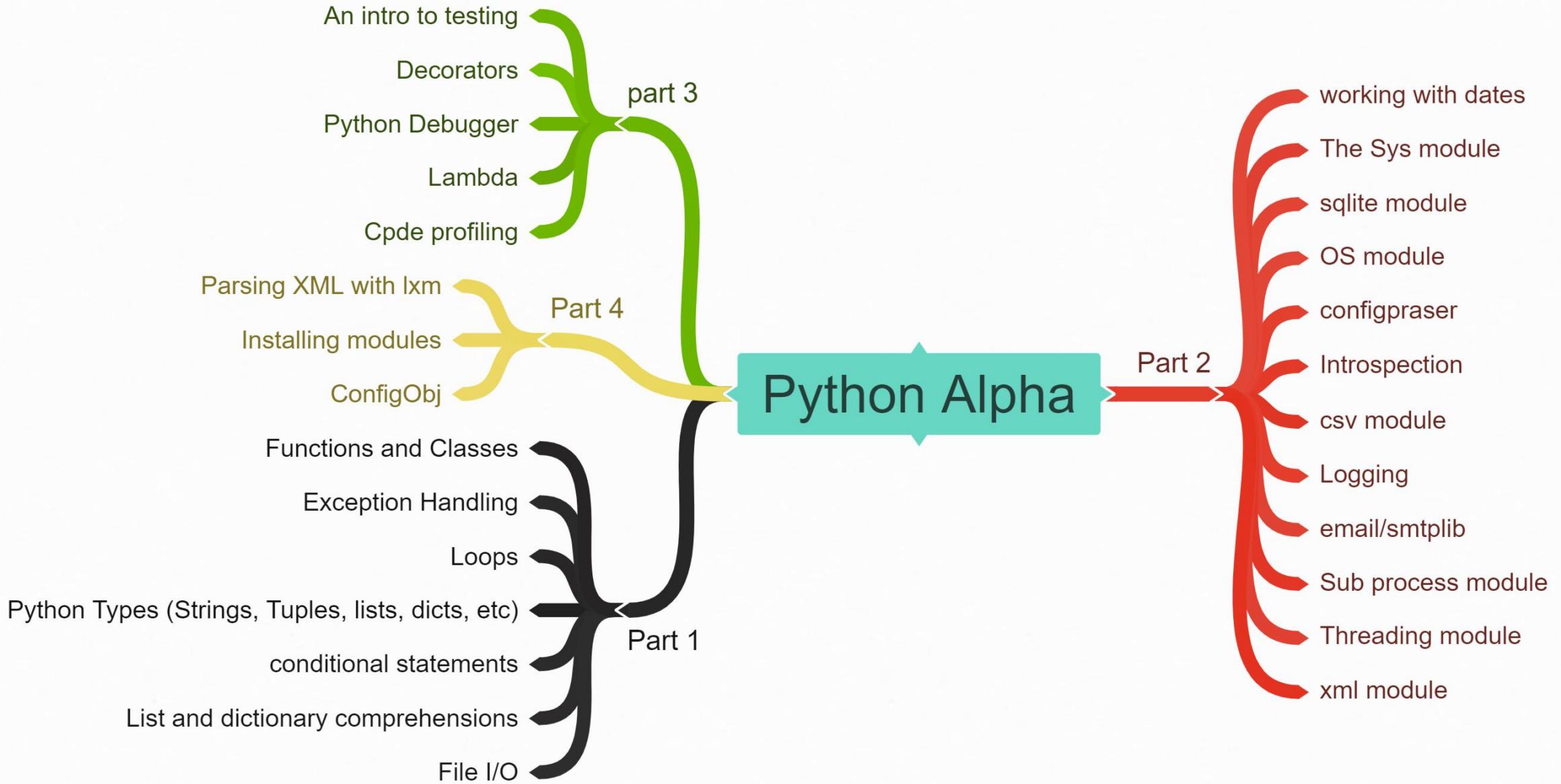
O Allah, the stars have vanished, the eyes have slept, the kings have locked their doors, and Your door remains open. Every lover has found privacy with their beloved, and here I am standing before You.”

# Study, Rinse and Repeat

- Please subscribe to our [YouTube Channel](#) to be on top of your studies.
- Please logon to our website <https://alnafi.com/login/>
- Use your username and password to logon
- Please keep an eye on [zone@alnafi.com](mailto:zone@alnafi.com) emails
- Please review the videos of 30 minutes daily
- Please review the notes daily.
- Please take time for clearing up your mind and reflect on how things are proceeding in your studies.

# Python Roadmap





جزاك الله

Please send us your questions at [zone@alnafi.com](mailto:zone@alnafi.com) We will only answer our Nafi Members. So please quote your membership number within the email.

Soon we will completely move to the portal to answer any further questions and will remove the email answering fuction.



AL NAFI,  
A company with a focus on education,  
wellbeing and renewable energy.

# Python Alpha 100

An intro to Python Alpha course coverage

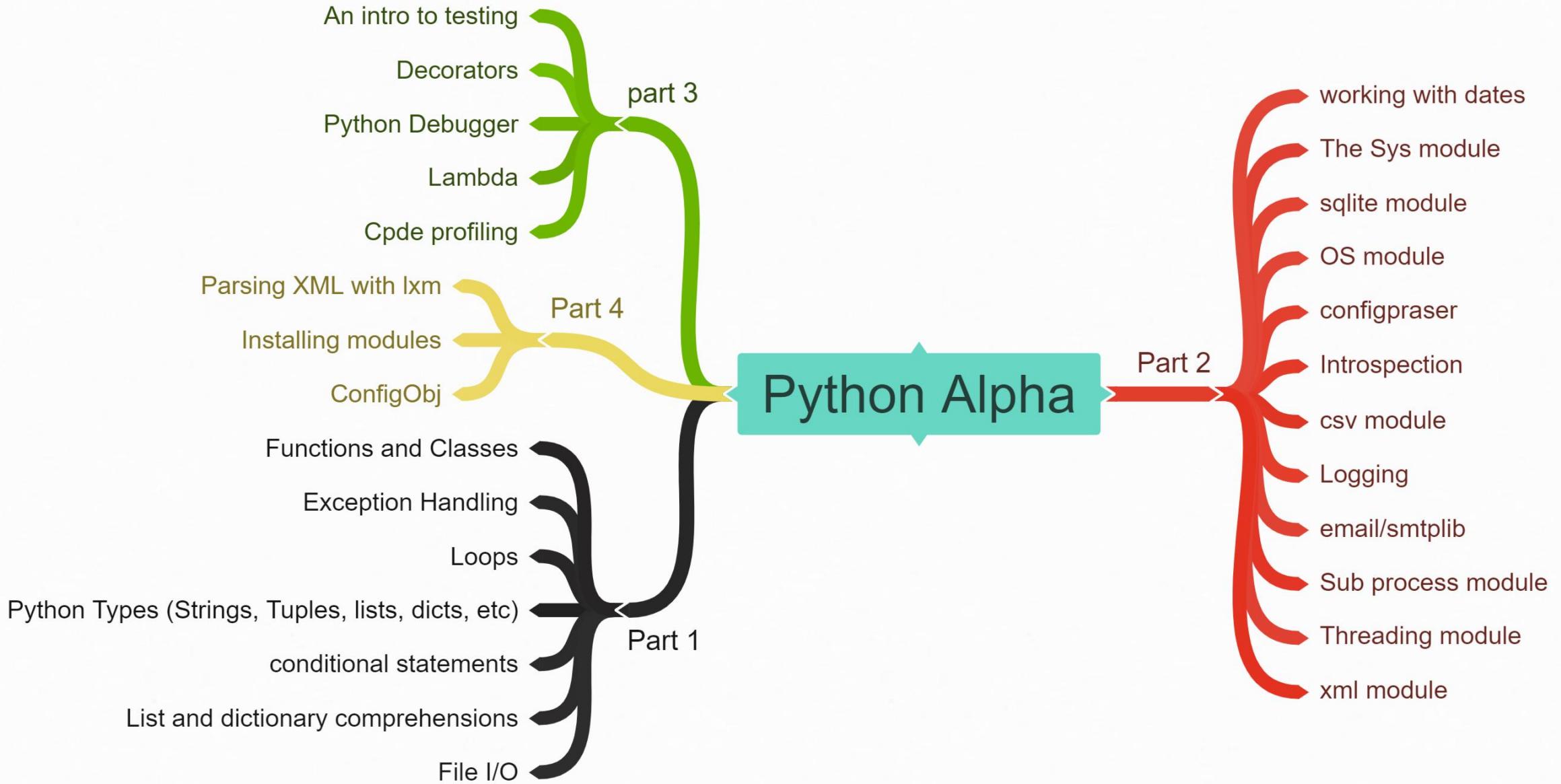
# Dua of the day

إِلَهِي غَارَتِ النُّجُومُ، وَنَامَتِ الْعَيْنُونُ وَغَلَقَتِ الْمُلُوكُ أَبْوَابَهَا، وَبَابُكَ مَفْتُوحٌ، وَخَلَى كُلُّ حَبِيبٍ بِحَبِيبِهِ، وَهَذَا مَقَامِي بَيْنَ يَدَيْكَ

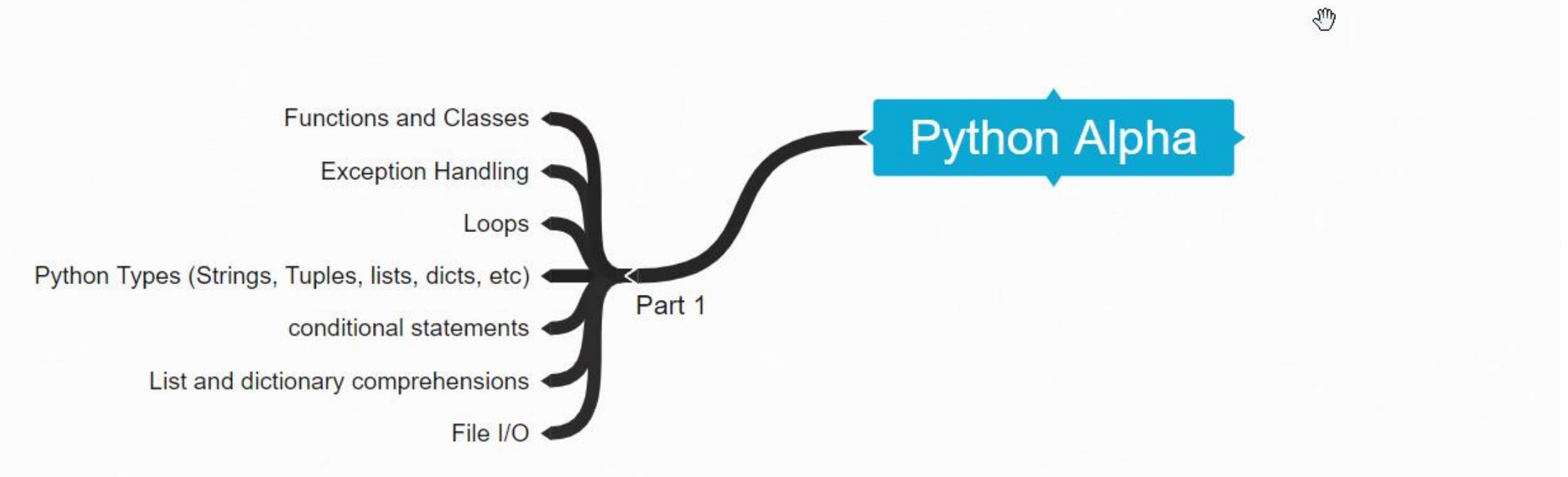
O Allah, the stars have vanished, the eyes have slept, the kings have locked their doors, and Your door remains open. Every lover has found privacy with their beloved, and here I am standing before You.”

# Study, Rinse and Repeat

- Please subscribe to our [YouTube Channel](#) to be on top of your studies.
- Please logon to our website <https://alnafi.com/login/>
- Use your username and password to logon
- Please keep an eye on [zone@alnafi.com](mailto:zone@alnafi.com) emails
- Please review the videos of 30 minutes daily
- Please review the notes daily.
- Please take time for clearing up your mind and reflect on how things are proceeding in your studies.



# Learning the Basics



# Learning the Basics continued...

We will cover the following topics in the next few lectures:

- IDLE
- Strings
- List, Dictionaries and Tuples
- Conditional Statements
- Loops
- Comprehensions

# Learning the Basics continued...

We will cover the following topics in the next few lectures:

- Comprehensions
- Exceptions Handling
- File I/O
- Importing modules and packages
- Functions
- Classes

# IDLE

- No relation to Eric IDLE and who can forget John Clesse



# Python IDLE

- It's a code editor
- Also known as IDE (Integrated Development Environment)
- It provides color highlighting of the key words in the python language
- Provides auto-complete
- A debugger and lots of other fun things ☺

# IDLE = IDE (sort of)

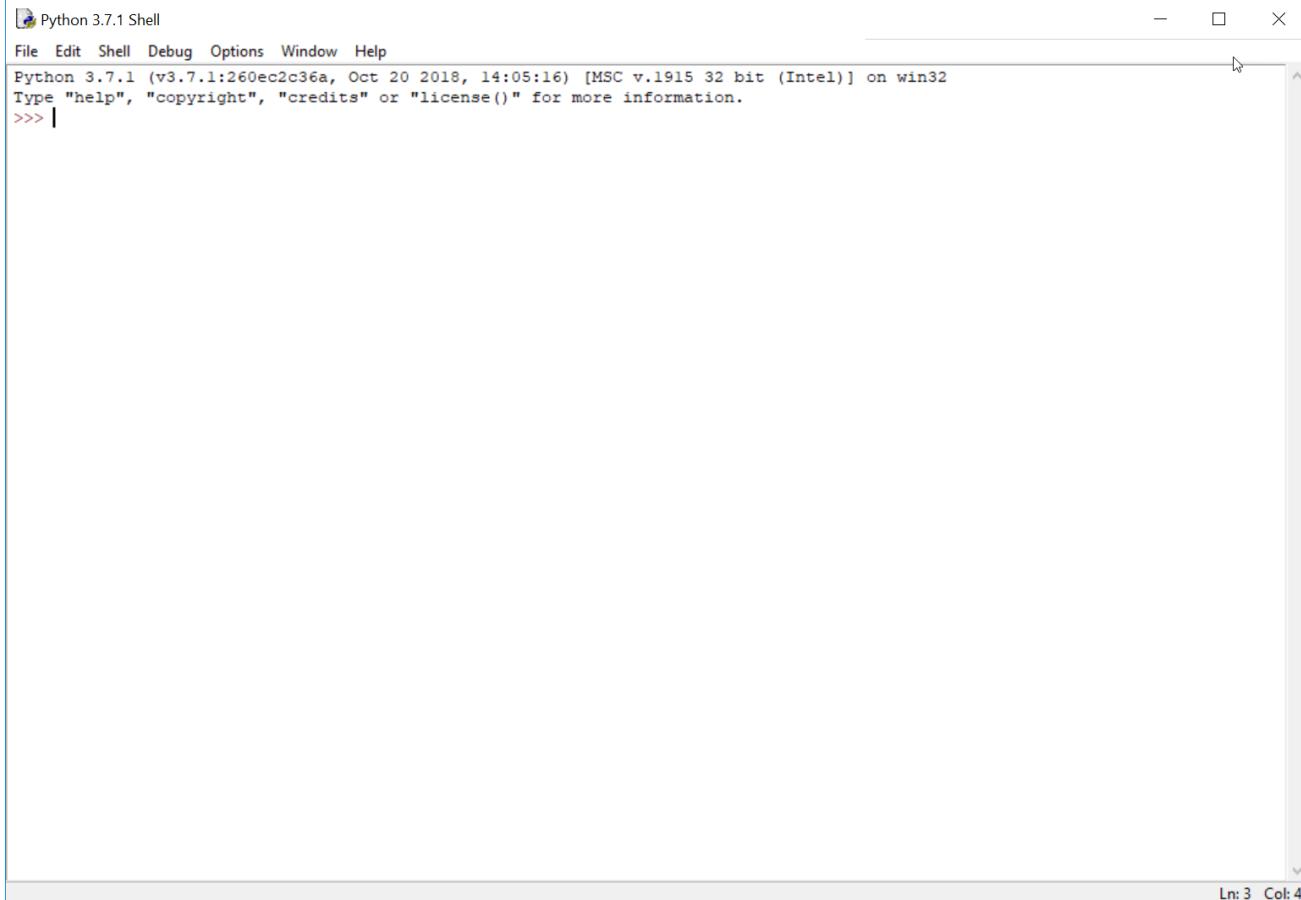
- IDLE is kind of lite IDS
- Which means that it doesn't have all the features of the full blown IDE
- Also known as lite debugger
- Provides basic features

# Installing Python good old fashion way

Watch the video separately via the following link.

<https://youtu.be/CFiVrOVt8W8>

# Walkthrough of IDLE

A screenshot of the Python 3.7.1 Shell window in IDLE. The window title is "Python 3.7.1 Shell". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main text area displays the Python version information: "Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:05:16) [MSC v.1915 32 bit (Intel)] on win32" and the instruction "Type "help", "copyright", "credits" or "license()" for more information. A cursor is visible at the start of a new line of code, indicated by the text ">>> |". The status bar at the bottom shows "Ln: 3 Col: 4".

It's a Python shell where you can type short scripts and see their output and even interact with the code in real-time.

There is no compiling of the code as Python is an interpretive language runs via the interpreter.

جزاك الله

To ask questions, please logon to the portal <https://alnafi.com/login/> and use your username and password. From within the portal you can ask questions. We will only answer questions if they are coming through the portal and not on email.

For any other queries please reach out on [info@alnafi.com](mailto:info@alnafi.com)



AL NAFI,  
A company with a focus on education,  
wellbeing and renewable energy.

# Python Alpha 101 a

## Strings

# Dua of the day to recite after Takbeer in Salah

اللَّهُمَّ بَاعِدْ بَيْنِي وَ بَيْنَ حَطَائِيَّاً كَمَا  
بَاعَدْتَ بَيْنَ الْمَشْرِقِ وَ الْمَغْرِبِ،  
اللَّهُمَّ نَقِّنِي مِنْ حَطَائِيَّاً كَمَا يُنَقِّي التَّوْبَةُ  
الْأَيْضُ مِنَ الدَّنَسِ، اللَّهُمَّ اغْسِلْنِي مِنْ  
حَطَائِيَّاً بِالثَّلْجِ وَ الْمَاءِ وَ الْبَرَدِ

O Allah , separate me from my sins as You have separated the East from the West. O Allah, cleanse me of my transgressions as the white garment is cleansed of stains . O Allah , wash away my sins with ice and water and frost.

Al-Bukhari 1/181, Muslim 1/419

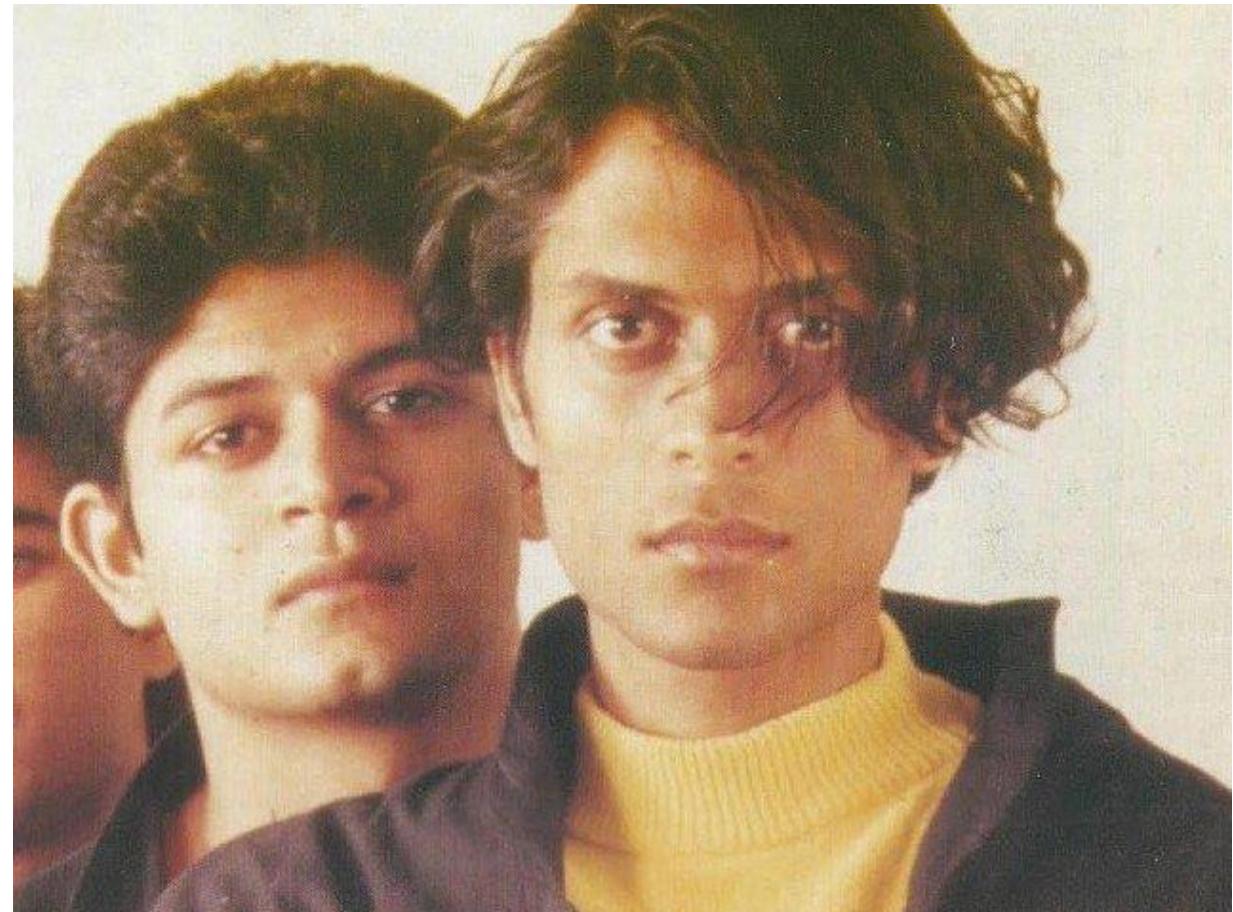
# Study, Rinse and Repeat

- Please subscribe to our [YouTube Channel](#) to be on top of your studies.
- Please logon to our website <https://alnafi.com/login/>
- Use your username and password to logon
- Please keep an eye on [zone@alnafi.com](mailto:zone@alnafi.com) emails
- Please review the videos of 30 minutes daily
- Please review the notes daily.
- Please take time for clearing up your mind and reflect on how things are proceeding in your studies.

# Python Strings not the other strings

In Python we will find many data types. As you have seen from Python Primer. So we will repeat them again to re-enforce concepts 😊

In Python we have string, integer, float, list, dict and tuple. Lets talk about string first.



# So what is python strings

A *string* is simply a series of characters. Anything inside quotes is considered

a string in Python, and you can use single or double quotes around your strings like this:

“This is a string”

“Muhammad Faisal, Rehan-ul-Haque, Muhammad Amir Majeed and Muhammad Ali, Ishraque Ansari are a string”

This flexibility allows you to use quotes and apostrophes within your strings:

'I told my friend, "Python is my favorite language!"'

"The language 'Python' is named after Monty Python, not the snake."

"One of Python's strengths is its diverse and supportive community."

# What to cover about strings 😊

- How to create strings
- String concatenation
- String methods
- String slicing
- String substitution

# Creating a string

Strings are created in three ways. We can use:

- Single quote
- Double quotes
- Triple quotes

# String examples

```
my_string = "Welcome to Python!"
```

```
another_string = 'The bright red fox jumped the fence.'
```

```
a_long_string = """This is a  
multi-line string.  
It covers more than one line""
```

# More string examples

```
my_string = "I'm a Python programmer!"
```

```
otherString = 'The word "python" usually refers to a snake'
```

```
tripleString = """Here's another way to embed "quotes" in a string"""
```

# str method aka casting

```
>>> my_number = 123
```

```
>>> my_string = str(my_number)
```

If you type the code above into your interpreter, you'll find that you have transformed the integer value into a string and assigned the string to the variable `my_string`. This is known as **casting**

But you'll also find that you can't always do the reverse, such as casting a string like 'ABC' into an integer. If you do that, you'll end up with an error like the one in the following example:

```
int('ABC')
```

This throw out an error message.

What this means is that we cannot convert a literal into an integer 😊

However if we do this way

```
x=int("123")
```

Then it would work fine

**Note:**

String in python is immutable type. What this means is that you cannot change a string content after creation.

```
my_string = "abc"
```

```
my_string[0] = "d"
```

The above will throw out an error message.

# String Concatenation

Concatenation means combining two things together.

Or in our case adding two strings together.

```
string_one="sharfoo is a good"  
string_two=" guy"  
string_three=string_one + string_two  
print (string_three)
```

The '+' operator concatenates the two strings into one.

# String Methods

String is an object in Python.

Everything is an object in Python.

Which means Python is an object oriented language ☺

Don't sweat as we will cover this later inshAllah.

# String Methods continued...

For now lets talk about strings more.

For example

```
my_string = "This is a string"
```

What if we need the above string in upper or lower case ☺

# Strings in upper and lower case...

```
my_string = "This is a string"
```

```
my_string.upper()
```

```
my_string = "This is a string"
```

```
my_string.lower()
```

# All the string commands

```
dir(my_string)
```

# Asking for help in python

```
help(my_string.upper)
```

Help is the build in function within Python which will come very handy.

# Introspection means ??

Introspection meaning from dictionary perspective:

- the examination or observation of one's own mental and emotional processes.
- "quiet introspection can be extremely valuable"

In python introspection allows us to ask Python about itself ☺

```
type(my_string)
```

جزاك الله

To ask questions, please logon to the portal <https://alnafi.com/login/> and use your username and password. From within the portal you can ask questions. We will only answer questions if they are coming through the portal and not on email.

For any other queries please reach out on [info@alnafi.com](mailto:info@alnafi.com)



AL NAFI,  
A company with a focus on education,  
wellbeing and renewable energy.

# Python Alpha 101 b

## Strings

# Dua of the day to recite after Takbeer in Salah

اللَّهُمَّ بَاعِدْ بَيْنِي وَ بَيْنَ حَطَائِيَّاً كَمَا  
بَاعَدْتَ بَيْنَ الْمَشْرِقِ وَ الْمَغْرِبِ،  
اللَّهُمَّ نَقِّنِي مِنْ حَطَائِيَّاً كَمَا يُنَقِّي التَّوْبَةُ  
الْأَيْضُ مِنَ الدَّنَسِ، اللَّهُمَّ اغْسِلْنِي مِنْ  
حَطَائِيَّاً بِالثَّلْجِ وَ الْمَاءِ وَ الْبَرَدِ

O Allah , separate me from my sins as You have separated the East from the West. O Allah, cleanse me of my transgressions as the white garment is cleansed of stains . O Allah , wash away my sins with ice and water and frost.

Al-Bukhari 1/181, Muslim 1/419

# Study, Rinse and Repeat

- Please subscribe to our [YouTube Channel](#) to be on top of your studies.
- Please logon to our website <https://alnafi.com/login/>
- Use your username and password to logon
- Please keep an eye on [zone@alnafi.com](mailto:zone@alnafi.com) emails
- Please review the videos of 30 minutes daily
- Please review the notes daily.
- Please take time for clearing up your mind and reflect on how things are proceeding in your studies.

# String Slicing

String Slicing is used a lot in real life. Within Machine Learning, Deep Learning, Data Wrangling, AI, website, Games you name it!!

```
my_string = "My Country Pakistan"  
my_string[0:0]
```

We can even use

```
my_string = "My Country Pakistan"  
my_string[0:-3]
```

# Slicing down to a single character

```
my_string = "My Country Pakistan"  
print(my_string[0])
```

# String Slicing has real usage

As mentioned string slicing is used in ML, DL, AI, DS and big data for parsing fixed width records etc along with other use cases.

Especially text file processing

Now we are increasing our understanding and moving slowly towards ML, DL, AI and other emerging fields using Python ☺

# String Formatting AKA substitution

String formatting (AKA substitution) is the topic of substituting values into a base string. Most of the time, you will be inserting strings within strings;

However you will also find yourself inserting integers and floats into strings quite often as well. There are two different ways to accomplish this task. We'll start with the old way of doing things and then move on to the new.

# The old way of string substitution step 1

```
my_string = "I like %s" % "Python"
```

```
my_string
```

# The old way of string substitution step 2

```
var = "cookies"
```

```
newString = "I like %s" % var
```

```
newString
```

## Note:

The **%s** is the important piece in the code as it tells Python that we may be inserting text soon. If we follow the string with a percent sign and another string or variable, then Python will attempt to insert it into the string. We can insert multiple strings by putting multiple instances of **%s** inside our string. You'll see that in the last example. Just note that when you insert more than one string, you have to enclose the strings that you're going to insert with parentheses.

# The old way of string substitution step 3

```
var = "cookies"
```

```
another_string = "I like %s and %s" % ("Python", var)  
another_string
```

## Note:

The **%s** is the important piece in the code as it tells Python that we may need to insert text soon. If we follow the string with a percent sign and another string or variable, then Python will attempt to insert it into the string. We can insert multiple strings by putting multiple instances of **%s** inside our string. You'll see that in the last example. Just note that when you insert more than one string, you have to enclose the strings that you're going to insert with parentheses.

# Examples with integers

```
my_string = "%i + %i = %i" % (1,2,3)
```

```
my_string
```

# Examples with float

```
float_string = "%f" % (1.23)
```

```
float_string
```

Or another float example

```
float_string2 = "%.2f" % (1.23)
```

```
float_string2
```

# Templates and the New String Formatting Methodology

Another way of formatting

```
print("%(lang)s is fun!" % {"lang":"Python"})
```

This probably looks pretty weird, but basically we just changed our %s into %(lang)s, which is basically the %s with a variable inside it. The second part is actually called a Python dictionary that we will be studying in the next class.

Basically it's a key:value pair, so when Python sees the key "lang" in the string AND in the key of the dictionary that is passed in, it replaces that key with its value.

# More Examples with text

```
print("%(value)s %(value)s %(value)s !" % {"value":"Sharfoo"})
```

# More examples with number

```
print("%(x)i + %(y)i = %(z)i" % {"x":1, "y":2, "z":3})
```

# Another example

```
"Python is as simple as {0}, {1}, {2}".format("a", "b", "c")
```

# Another example

```
xy = {"x":0, "y":10}  
print("Graph a point at where x={x} and y={y}".format(**xy))
```

# Python Official Documentation

<https://docs.python.org/3/>

String Formatting <https://docs.python.org/3/library/string.html#string-formatting>

More on String Formatting

<https://docs.python.org/3/library/string.html#formatexamples>

جزاك الله

To ask questions, please logon to the portal <https://alnafi.com/login/> and use your username and password. From within the portal you can ask questions. We will only answer questions if they are coming through the portal and not on email.

For any other queries please reach out on [info@alnafi.com](mailto:info@alnafi.com)



AL NAIFI,  
A company with a focus on education,  
wellbeing and renewable energy.

# Python Alpha 200

## Lists, Tuples and Dictionaries

# Dua of the day to recite after Takbeer in Salah

اللَّهُمَّ بَاعِدْ بَيْنِي وَ بَيْنَ حَطَائِيَّاً كَمَا  
بَاعَدْتَ بَيْنَ الْمَشْرِقِ وَ الْمَغْرِبِ،  
اللَّهُمَّ نَقِّنِي مِنْ حَطَائِيَّاً كَمَا يُنَقِّي التَّوْبَةُ  
الْأَيْضُ مِنَ الدَّنَسِ، اللَّهُمَّ اغْسِلْنِي مِنْ  
حَطَائِيَّاً بِالثَّلْجِ وَ الْمَاءِ وَ الْبَرَدِ

O Allah , separate me from my sins as You have separated the East from the West. O Allah, cleanse me of my transgressions as the white garment is cleansed of stains . O Allah , wash away my sins with ice and water and frost.

Al-Bukhari 1/181, Muslim 1/419

# Study, Rinse and Repeat

- Please subscribe to our [YouTube Channel](#) to be on top of your studies.
- Please logon to our website <https://alnafi.com/login/>
- Use your username and password to logon
- Please keep an eye on [zone@alnafi.com](mailto:zone@alnafi.com) emails
- Please review the videos of 30 minutes daily
- Please review the notes daily.
- Please take time for clearing up your mind and reflect on how things are proceeding in your studies.

# List, Tuples and Dictionaries

Python has several other important data types like Lists Tuples and Dictionaries, that we'll probably use every day down the line in using ML, DL, AI, Cyber Security and even software development

The 200 lecture aim is to get you all acquainted with each of these data types.

Once you have mastered these three data types plus the string data type from the previous lecture, you will be quite a ways along in your education of Python. You'll be using these four building blocks in 99% of all the applications you will write.

# Lists

A Python list is similar to displaying results or a range.

In Python, an empty list can be created in the following ways.

```
my_list = []
```

or

```
my_list = list()
```

As you can see, you can create the list using square brackets or by using the Python built-in, list. A list contains a list of elements, such as strings, integers, objects or a mixture of types.



# List examples

Let's take a look at some examples:

```
my_list = [1, 2, 3]
```

Or

```
my_list2 = ["a", "b", "c"]
```

Or

```
my_list3 = ["a", 1, "Python", 5]
```

# Nested list

You can also create lists of lists like this:

Occasionally we will combine two or three lists together.

```
my_list = [1, 2, 3]
```

```
my_list2 = ["a", "b", "c"]
```

```
my_list3 = ["a", 1, "Python", 5]
```

```
my_nested_list = [my_list, my_list2, my_list3]
```

```
print(my_nested_list)
```

# List extend method

```
combo_list = []
```

```
one_list = [4, 5]
```

```
combo_list.extend(one_list)
```

```
combo_list
```

# A slightly easier method to add two lists

```
my_list = [1, 2, 3]
```

```
my_list2 = ["a", "b", "c"]
```

```
combo_list = my_list + my_list2
```

```
combo_list
```

# Sorting out a list

```
alpha_list = [34, 23, 67, 100, 88, 2]
```

```
alpha_list.sort()
```

```
alpha_list
```

# A none list

```
alpha_list = [34, 23, 67, 100, 88, 2]  
sorted_list = alpha_list.sort()  
sorted_list  
print(sorted_list)
```

In this example, if we try to assign the sorted list to a variable.

However, when you call the sort() method on a list, it sorts the list in-place. So if we try to assign the result to another variable, then we'll find out that we'll get a None object, which is like a Null in other languages.

Thus when we want to sort something, just remember that we sort them in-place and we cannot assign it to a different variable.

# Slicing a list

```
alpha_list = [34, 23, 67, 100, 88, 2]
```

```
alpha_list[0:3]
```

# Tuples

A tuple is similar to a list, but you create them with parentheses instead of square brackets.

You can also use the tuple built-in. The main difference is that a tuple is immutable (Cannot be changed) while the list is mutable(can be changed).

How I remember Tuples ☺

Tuples are like couples which means they cannot be changed ☺

# Tuple example

```
my_tuple=(1,2,3,4,5)
```

```
my_tuple[0:3]
```

Or an empty

```
another_tuple=tuple()
```

# Tuple casting which means a list inside a tuple

```
abc=tuple([1,2,3])
```

This example has three elements inside it. Notice that it has a list inside it. This is an example of casting. We can change or cast an item from one data type to another. In this case we cast a list into a tuple.

If we want to change the abc tuple back into a list, we can do the following:

```
abc_list=list(abc)
```

# Dictionaries

Using a dictionary to associate value

Lists are *great*, but they are not always the best data structure for every situation.

# Dictionary example

Muhammad Ali, | 2002-6-17,2:58,2.58,2:39,2-25,2-55,2:54,2.18,2:55,2:55,2:22,2-21,2.22



Let's take a look at Muhammad Ali's data:

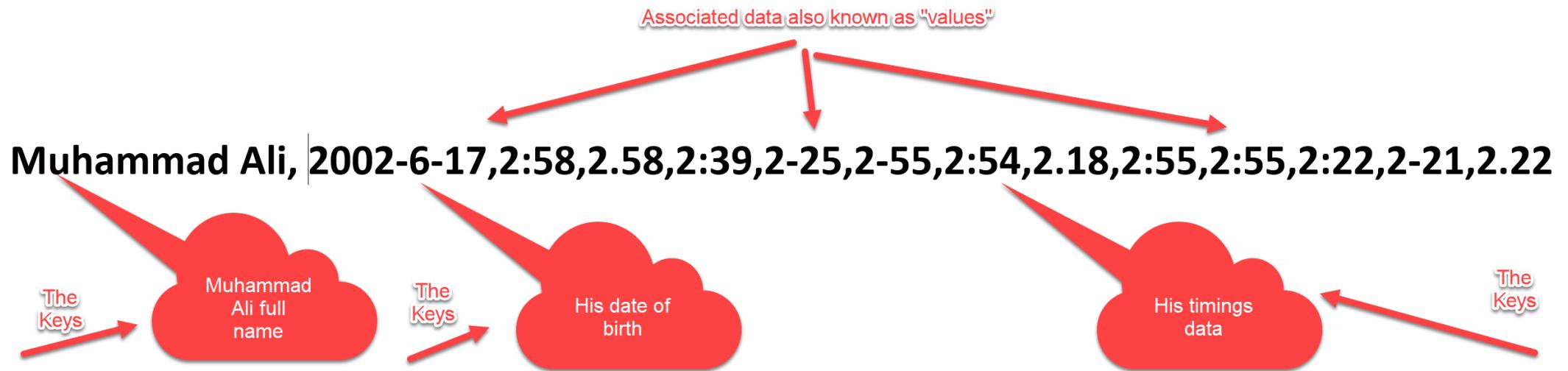
There's a definite **structure** here: the athlete's name, the date of birth, and then the list of times.

Let's continue to use a list for the timing data, because that still makes sense.

But let's make the timing data part of *another* data structure, which associates all the data for an athlete with a single variable.

For that we use a Python **dictionary**, which *associates data values with keys*:

# Another closer look at the data



# Python Dictionary revisited

Python dictionary is a hash table with indexed keys which can be immutable type. For example a string or number can be a key.

We need to be aware that dictionary is an unordered set of key:value pairs and the keys must be unique.

We can also call a dictionary instance's **key** to check if dictionary has a key using **in**

# Empty Dictionary code examples

Empty dictionary

```
my_dict = {}
```

Or

```
another_dict = dict()
```

# Dictionary with values

```
my_other_dict = {"one":1, "two":2, "three":3}  
my_other_dict
```

## Note:

All dictionaries are enclosed with curly braces. The last line is printed out so you can see how unordered a dictionary is. Now lets find out how to access a value in a dictionary.

# Accessing a value within a dictionary

```
my_dict = {"name": "Sharfoo", "address": "123 Gulshan"}  
my_dict["name"]
```

# Adding true and false statements

"name" in my\_dict

True

"state" in my\_dict

False

Boolean True and false is something we will cover later ☺ this is just a place holder for the upcoming lecture.

# Checking all the keys in a dictionary

```
my_dict = {"name": "Sharfoo", "address": "123 Gulshan"}
```

```
my_dict.keys()
```

To check values we use

```
my_dict.values()
```

# The importance of `in`

```
"name" in my_dict      # this is good
```

```
"name" in my_dict.keys() # this works too, but is slower
```

While this probably won't matter much to us right now, in a real job situation, seconds matter. When we have thousands of files to process, these little tricks can save us a lot of time in the long run!

جذب الله

To ask questions, please logon to the portal <https://alnafi.com/login/> and use your username and password. From within the portal you can ask questions. We will only answer questions if they are coming through the portal and not on email.

For any other queries please reach out on [info@alnafi.com](mailto:info@alnafi.com)



AL NAIFI,  
A company with a focus on education,  
wellbeing and renewable energy.

# Python Alpha 200

## Lists, Tuples and Dictionaries

# Dua of the day to recite after Takbeer in Salah

اللَّهُمَّ بَاعِدْ بَيْنِي وَ بَيْنَ حَطَائِيَّاً كَمَا  
بَاعَدْتَ بَيْنَ الْمَشْرِقِ وَ الْمَغْرِبِ،  
اللَّهُمَّ نَقِّنِي مِنْ حَطَائِيَّاً كَمَا يُنَقِّي التَّوْبَةُ  
الْأَيْضُ مِنَ الدَّنَسِ، اللَّهُمَّ اغْسِلْنِي مِنْ  
حَطَائِيَّاً بِالثَّلْجِ وَ الْمَاءِ وَ الْبَرَدِ

O Allah , separate me from my sins as You have separated the East from the West. O Allah, cleanse me of my transgressions as the white garment is cleansed of stains . O Allah , wash away my sins with ice and water and frost.

Al-Bukhari 1/181, Muslim 1/419

# Study, Rinse and Repeat

- Please subscribe to our [YouTube Channel](#) to be on top of your studies.
- Please logon to our website <https://alnafi.com/login/>
- Use your username and password to logon
- Please keep an eye on [zone@alnafi.com](mailto:zone@alnafi.com) emails
- Please review the videos of 30 minutes daily
- Please review the notes daily.
- Please take time for clearing up your mind and reflect on how things are proceeding in your studies.

# List, Tuples and Dictionaries

Python has several other important data types like Lists Tuples and Dictionaries, that we'll probably use every day down the line in using ML, DL, AI, Cyber Security and even software development

The 200 lecture aim is to get you all acquainted with each of these data types.

Once you have mastered these three data types plus the string data type from the previous lecture, you will be quite a ways along in your education of Python. You'll be using these four building blocks in 99% of all the applications you will write.

# Lists

A Python list is similar to displaying results or a range.

In Python, an empty list can be created in the following ways.

```
my_list = []
```

or

```
my_list = list()
```

As you can see, you can create the list using square brackets or by using the Python built-in, list. A list contains a list of elements, such as strings, integers, objects or a mixture of types.



# List examples

Let's take a look at some examples:

```
my_list = [1, 2, 3]
```

Or

```
my_list2 = ["a", "b", "c"]
```

Or

```
my_list3 = ["a", 1, "Python", 5]
```

# Nested list

You can also create lists of lists like this:

Occasionally we will combine two or three lists together.

```
my_list = [1, 2, 3]
```

```
my_list2 = ["a", "b", "c"]
```

```
my_list3 = ["a", 1, "Python", 5]
```

```
my_nested_list = [my_list, my_list2, my_list3]
```

```
print(my_nested_list)
```

# List extend method

```
combo_list = []
```

```
one_list = [4, 5]
```

```
combo_list.extend(one_list)
```

```
combo_list
```

# A slightly easier method to add two lists

```
my_list = [1, 2, 3]
```

```
my_list2 = ["a", "b", "c"]
```

```
combo_list = my_list + my_list2
```

```
combo_list
```

# Sorting out a list

```
alpha_list = [34, 23, 67, 100, 88, 2]
```

```
alpha_list.sort()
```

```
alpha_list
```

# A none list

```
alpha_list = [34, 23, 67, 100, 88, 2]  
sorted_list = alpha_list.sort()  
sorted_list  
print(sorted_list)
```

In this example, if we try to assign the sorted list to a variable.

However, when you call the sort() method on a list, it sorts the list in-place. So if we try to assign the result to another variable, then we'll find out that we'll get a None object, which is like a Null in other languages.

Thus when we want to sort something, just remember that we sort them in-place and we cannot assign it to a different variable.

# Slicing a list

```
alpha_list = [34, 23, 67, 100, 88, 2]
```

```
alpha_list[0:3]
```

# Tuples

A tuple is similar to a list, but you create them with parentheses instead of square brackets.

You can also use the tuple built-in. The main difference is that a tuple is immutable (Cannot be changed) while the list is mutable(can be changed).

How I remember Tuples ☺

Tuples are like couples which means they cannot be changed ☺

# Tuple example

```
my_tuple=(1,2,3,4,5)
```

```
my_tuple[0:3]
```

Or an empty

```
another_tuple=tuple()
```

# Tuple casting which means a list inside a tuple

```
abc=tuple([1,2,3])
```

This example has three elements inside it. Notice that it has a list inside it. This is an example of casting. We can change or cast an item from one data type to another. In this case we cast a list into a tuple.

If we want to change the abc tuple back into a list, we can do the following:

```
abc_list=list(abc)
```

# Dictionaries

Using a dictionary to associate value

Lists are *great*, but they are not always the best data structure for every situation.

# Dictionary example

Muhammad Ali, | 2002-6-17,2:58,2.58,2:39,2-25,2-55,2:54,2.18,2:55,2:55,2:22,2-21,2.22



Let's take a look at Muhammad Ali's data:

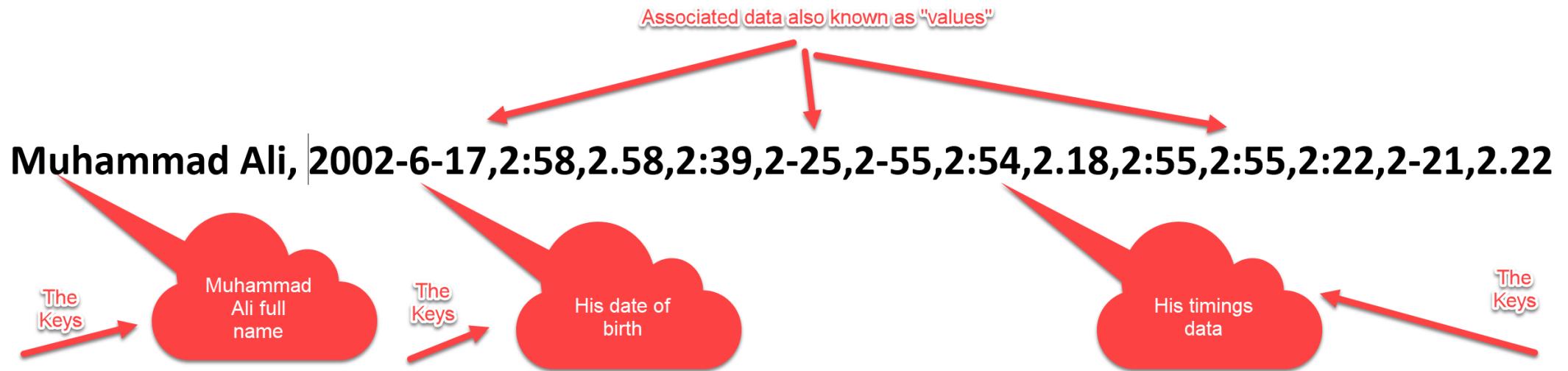
There's a definite **structure** here: the athlete's name, the date of birth, and then the list of times.

Let's continue to use a list for the timing data, because that still makes sense.

But let's make the timing data part of *another* data structure, which associates all the data for an athlete with a single variable.

For that we use a Python **dictionary**, which *associates data values with keys*:

# Another closer look at the data



# Python Dictionary revisited

Python dictionary is a hash table with indexed keys which can be immutable type. For example a string or number can be a key.

We need to be aware that dictionary is an unordered set of key:value pairs and the keys must be unique.

We can also call a dictionary instance's **key** to check if dictionary has a key using **in**

# Empty Dictionary code examples

Empty dictionary

```
my_dict = {}
```

Or

```
another_dict = dict()
```

# Dictionary with values

```
my_other_dict = {"one":1, "two":2, "three":3}  
my_other_dict
```

## Note:

All dictionaries are enclosed with curly braces. The last line is printed out so you can see how unordered a dictionary is. Now lets find out how to access a value in a dictionary.

# Accessing a value within a dictionary

```
my_dict = {"name": "Sharfoo", "address": "123 Gulshan"}  
my_dict["name"]
```

# Adding true and false statements

"name" in my\_dict

True

"state" in my\_dict

False

Boolean True and false is something we will cover later ☺ this is just a place holder for the upcoming lecture.

# Checking all the keys in a dictionary

```
my_dict = {"name": "Sharfoo", "address": "123 Gulshan"}
```

```
my_dict.keys()
```

To check values we use

```
my_dict.values()
```

# The importance of `in`

```
"name" in my_dict      # this is good
```

```
"name" in my_dict.keys() # this works too, but is slower
```

While this probably won't matter much to us right now, in a real job situation, seconds matter. When we have thousands of files to process, these little tricks can save us a lot of time in the long run!

جذب الله

To ask questions, please logon to the portal <https://alnafi.com/login/> and use your username and password. From within the portal you can ask questions. We will only answer questions if they are coming through the portal and not on email.

For any other queries please reach out on [info@alnafi.com](mailto:info@alnafi.com)



AL NAIFI,  
A company with a focus on education,  
wellbeing and renewable energy.

# Python Alpha 300

## Conditional Statements if/else

# Dua of the day to recite after Takbeer in Salah

اللَّهُمَّ بَاعِدْ بَيْنِي وَ بَيْنَ حَطَائِيَّاً كَمَا  
بَاعَدْتَ بَيْنَ الْمَشْرِقِ وَ الْمَغْرِبِ،  
اللَّهُمَّ نَقِّنِي مِنْ حَطَائِيَّاً كَمَا يُنَقِّي التَّوْبَةُ  
الْأَيْضُ مِنَ الدَّنَسِ، اللَّهُمَّ اغْسِلْنِي مِنْ  
حَطَائِيَّاً بِالثَّلْجِ وَ الْمَاءِ وَ الْبَرَدِ

O Allah , separate me from my sins as You have separated the East from the West. O Allah, cleanse me of my transgressions as the white garment is cleansed of stains . O Allah , wash away my sins with ice and water and frost.

Al-Bukhari 1/181, Muslim 1/419

# Study, Rinse and Repeat

- Please subscribe to our [YouTube Channel](#) to be on top of your studies.
- Please logon to our website <https://alnafi.com/login/>
- Use your username and password to logon
- Please keep an eye on [zone@alnafi.com](mailto:zone@alnafi.com) emails
- Please review the videos of 30 minutes daily
- Please review the notes daily.
- Please take time for clearing up your mind and reflect on how things are proceeding in your studies.

# Conditional statements

Like any other computer language python has a conditional statement.

The conditional statement checks to see if a statement is True or False. However we will also look at the following Boolean operations. **and**, **or** and **not**.

These operations can change the behavior of the conditional in simple and complex ways, depending on our project.



# The if statement

```
if 2 > 1:  
    print("This is a True statement!")
```

This conditional tests the "truthfulness" of the following statement:  $2 > 1$ .

Since this statement evaluates to True, it will cause the last line in the example to print to the screen or standard out (stdout).

# Python Cares About Space

In Python space is important. Either tab or space the indent space is the key.

If the indent is not correct then python code will not run.

Do not mix tab or spaces together!

**The recommended number of space for a block of code is 4. This rule is followed by all the developers in the world as a de-facto standard.**

Though we can indent our code with as many spaces as we like as long as its consistent for a block of code.

# Another examples

```
var1 = 1
```

```
var2 = 3
```

```
if var1 > var2:
```

```
    print("This is also True")
```

# Example with else

```
var1 = 5  
var2 = 3  
if var1 > var2:  
    print("This is also True")  
else:  
    print("That was False!")
```

# Example with elif

```
value = 8  
if value < 10:  
    print("That's a great deal!")  
elif 10 <= value <= 20:  
    print("I'd still pay that...")  
else:  
    print("Wow! That's too much!")
```

# Boolean Operations

**or** means that if any conditional that is "ored" together is True, then the following statement runs

**and** means that all statements must be True for the following statement to run

**not** means that if the conditional evaluates to False, it is True. This is the most confusing, in my opinion.

or example

x = 10

y = 20

if x < 10 or y > 15:

```
    print("This statement was True!")
```

# and example

```
x = 10
```

```
y = 10
```

```
if x == 10 and y == 15:
```

```
    print("This statement was True")
```

```
else:
```

```
    print("The statement was False!")
```

# A word of wisdom from myself

You can use **or** and **and** in more than two statements together.  
However, I would not recommend that as that the more statements  
that you combine, the harder it can be to understand and debug so  
**Simplicity** is the key.

# KISS methodology.

As the famous modified saying that I use is **KISS**.

**Keep it Simple Sharfoo**



# Not example

```
my_list = [1, 2, 3, 4]
```

```
x = 10
```

```
if x not in my_list:
```

```
    print("'x' is not in the list, so this is True!")
```

# Another not example

```
x = 10
```

```
if x != 11:
```

```
    print("x is not equal to 11!")
```

Symbol	Definition
==	Equal to
!=	Not equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

# A complex example

```
my_list = [1, 2, 3, 4]
```

```
x = 10
```

```
z = 11
```

```
if x not in my_list and z != 10:
```

```
    print("This is True!")
```

# Checking for nothing

Because we are talking about statements that evaluate to True, we probably need to cover what evaluates to False.

Python has the keyword **False** which has been mentioned a few times.

However an empty string, tuple or list also evaluates to False.

# None continued....

There is also another keyword that basically evaluates to False which is called None.

The None value is used to represent the absence of value. It's kind of analogous to Null, which you find in databases.

# None examples

```
empty_list = []
```

```
empty_tuple = ()
```

```
empty_string = ""
```

```
nothing = None
```

# None examples for list, tuple, string and nothing

```
if empty_list == []:
    print("It's an empty list!")
```

```
if empty_tuple:
    print("It's not an empty tuple!")
```

```
if not empty_string:
    print("This is an empty string!")
```

```
if not nothing:
    print("Then it's nothing!")
```

# More examples for none...

```
if empty_string == "":  
    print("This is an empty string!")
```

# When nothing is printed ☺

```
empty_string = "something"  
if empty_string == "":  
    print("This is an empty string!")
```

# False is the output

`empty_list == empty_string`

`empty_string == nothing`

# Special Characters

Strings can contain special characters, like tabs or new lines. We need to be aware of those as they can sometimes crop up and cause problems.

For example, the new line character is defined as "n", while the tab character is defined as "t". Let's see a couple of examples so you will better understand what these do:

# Special characters n

```
print("I have a \n new line in the middle")
```

# Special character t

```
print("This sentence is \ttabbed!")
```

# Special character \

```
print("This is a backslash \\")
```

جزاك الله

To ask questions, please logon to the portal <https://alnafi.com/login/> and use your username and password. From within the portal you can ask questions. We will only answer questions if they are coming through the portal and not on email.

For any other queries please reach out on [info@alnafi.com](mailto:info@alnafi.com)



AL NAIFI,  
A company with a focus on education,  
wellbeing and renewable energy.

# Python Alpha 400a

## Loops **for** and **while**

# Dua of the day to recite after Takbeer in Salah

اللَّهُمَّ بَاعِدْ بَيْنِي وَ بَيْنَ حَطَائِيَّاً كَمَا  
بَاعَدْتَ بَيْنَ الْمَشْرِقِ وَ الْمَغْرِبِ،  
اللَّهُمَّ نَقِّنِي مِنْ حَطَائِيَّاً كَمَا يُنَقِّي التَّوْبَةُ  
الْأَيْضُ مِنَ الدَّنَسِ، اللَّهُمَّ اغْسِلْنِي مِنْ  
حَطَائِيَّاً بِالثَّلْجِ وَ الْمَاءِ وَ الْبَرَدِ

O Allah , separate me from my sins as You have separated the East from the West. O Allah, cleanse me of my transgressions as the white garment is cleansed of stains . O Allah , wash away my sins with ice and water and frost.

Al-Bukhari 1/181, Muslim 1/419

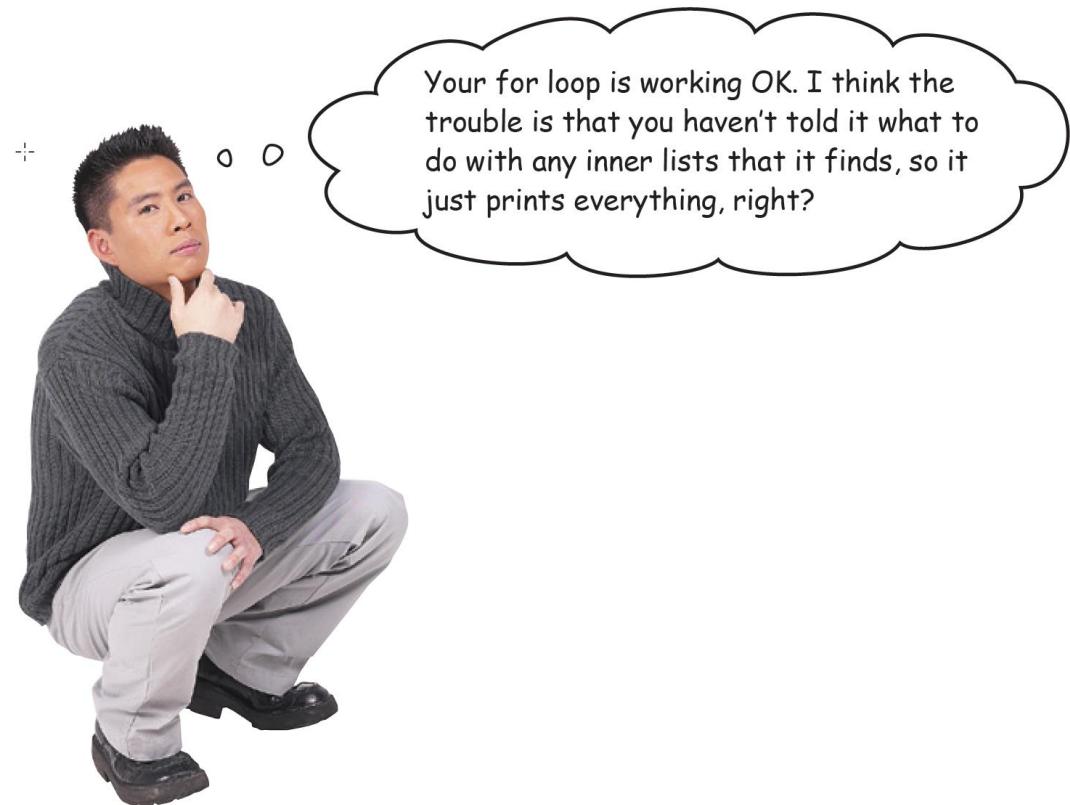
# Study, Rinse and Repeat

- Please subscribe to our [YouTube Channel](#) to be on top of your studies.
- Please logon to our website <https://alnafi.com/login/>
- Use your username and password to logon
- Please keep an eye on [zone@alnafi.com](mailto:zone@alnafi.com) emails
- Please review the videos of 30 minutes daily
- Please review the notes daily.
- Please take time for clearing up your mind and reflect on how things are proceeding in your studies.

# Loops

Every programming language has a looping construct. In Python we have two types.

- The **for** loop and
- The **while** loop



# Famous loop of all 😊

In real world we will see that **for** is by far the most popular.

Loops are used when we want to do something many times. Usually you will find that you need to do some operation or a set of operations on a piece of data over and over.

This is where loops come in. They make it really easy to apply this sort of logic to your data.

In machine learning, deep learning and AI loops are widely used. Along with Offensive hacking tools and scripts for both network, application and wireless hacking. More on that in due time 😊

# The for loop

As mentioned above, you use a loop when you want to iterate over something  $n$  number of times.

Let's use Python's builtin range function. The range function will create a list that is  $n$  in length.

For examples

`range(10)`

Or

`range(5,10)`

Or

`list(range(1,10,2))`

# For examples continued.....

```
for number in range(5):  
    print(number)
```

Or

```
for number in [0,1,2,3,4]:  
    print(number)
```

# For examples continued.....

```
a_dict = {"one":1, "two":2, "three":3}
```

```
for key in a_dict:  
    print(key)
```

# For sorting

```
a_dict = {1:"one", 2:"two", 3:"three"}
```

```
keys = a_dict.keys()
```

```
sorted(keys)
```

```
for key in keys:
```

```
    print(key)
```

# A more complex for example

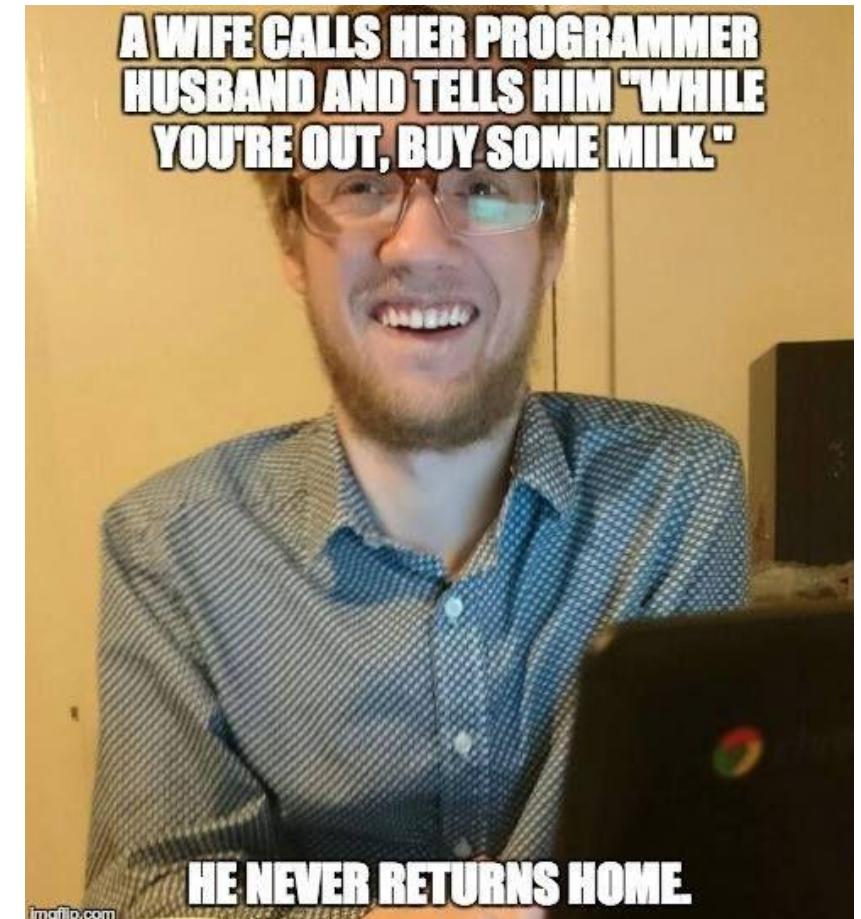
for number in range(10):

    if number % 2 == 0:

        print(number)

# The **while** loop

The while loop is also used to repeat sections of code, but instead of looping **n** number of times, it will only loop until a specific condition is met.



# The **while** loop example

```
i = 0
```

```
while i < 10:  
    print(i)
```

جذب الله

To ask questions, please logon to the portal <https://alnafi.com/login/> and use your username and password. From within the portal you can ask questions. We will only answer questions if they are coming through the portal and not on email.

For any other queries please reach out on [info@alnafi.com](mailto:info@alnafi.com)



AL NAIFI,  
A company with a focus on education,  
wellbeing and renewable energy.

# Python Alpha 400b

## Loops **for** and **while**

# Dua of the day to recite after Takbeer in Salah

## Supplication at the start of the prayer after takbeer

اللَّهُمَّ رَبَّ جِبْرائِيلَ ، وَمِيكَائِيلَ ، وَإِسْرَافِيلَ ، فَاطِرِ السَّمَاوَاتِ وَالْأَرْضِ ، عَالَمِ الْغَيْبِ وَالشَّهادَةِ  
أَنْتَ تَحْكُمُ بَيْنَ عِبَادِكَ فِيمَا كَانُوا فِيهِ يَخْتَلِفُونَ . اهْدِنِي لِمَا أَخْتَلِفَ فِيهِ مِنَ الْحَقِّ بِإِذْنِكَ ، إِنَّكَ  
تَهْدِي مَنْ تَشَاءُ إِلَى صِرَاطٍ مُسْتَقِيمٍ

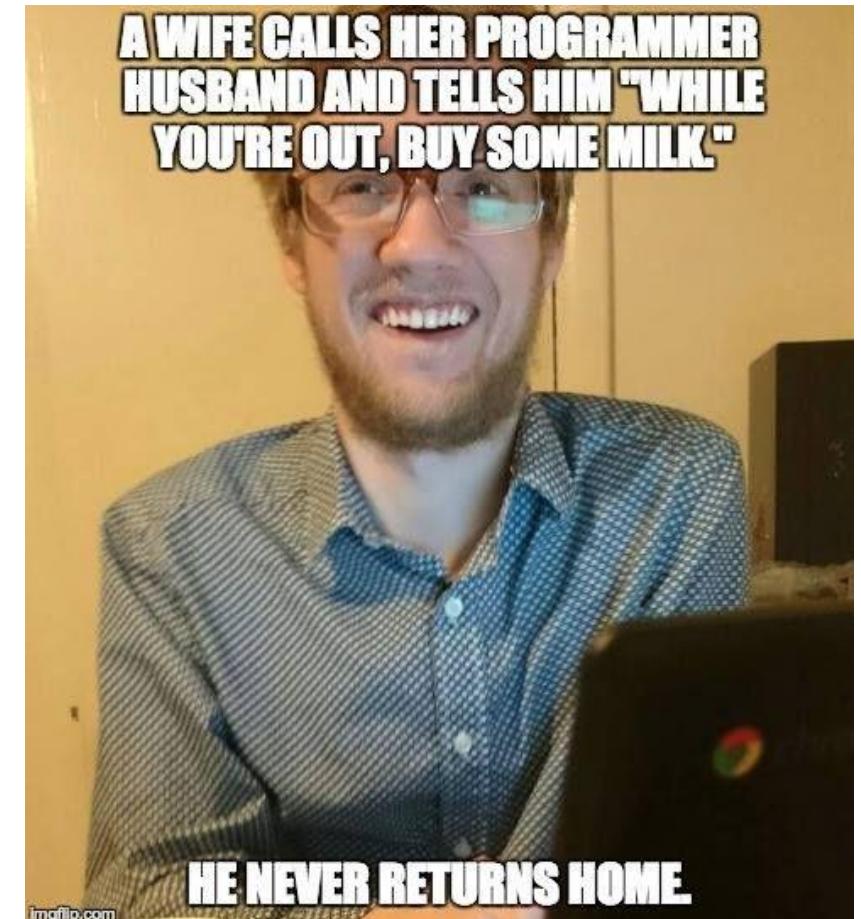
O Allah, Lord of Jibra-eel, Meeka-eel and Israfeel (great angles), Creator of the heavens and the Earth, Knower of the seen and the unseen. You are the arbitrator between Your servants in that which they have disputed. Guide me to the truth by Your leave, in that which they have differed, for verily You guide whom You will to a straight path. Reference: Muslim 1/534

# Study, Rinse and Repeat

- Please subscribe to our [YouTube Channel](#) to be on top of your studies.
- Please logon to our website <https://alnafi.com/login/>
- Use your username and password to logon
- Please keep an eye on [zone@alnafi.com](mailto:zone@alnafi.com) emails
- Please review the videos of 30 minutes daily
- Please review the notes daily.
- Please take time for clearing up your mind and reflect on how things are proceeding in your studies.

# The **while** loop

The while loop is also used to repeat sections of code, but instead of looping **n** number of times, it will only loop until a specific condition is met.



# The **while** loop example

i = 0

while i < 10:

    print(i)

    i = i+1

# while example with break

```
i = 0
```

```
while i < 10:
```

```
    print(i)
```

```
    if i == 5:
```

```
        break
```

```
    i += 1
```

# while loop example with flow control

```
i=0
```

```
while i<10:
```

```
    if i==3:
```

```
        i+=1
```

```
        continue
```

```
    print(i)
```

```
    if i==5:
```

```
        break
```

```
    i+=1
```

# What else is for in loops

The else statement in loops only executes if the loop completes successfully. The primary use of the else statement is for searching for items. For example:

# What else is for in loops complex example

```
my_list = [1, 2, 3, 4, 5]
for i in my_list:
    if i==3:
        print("Items found!")
        break
    else:
        print(i)
print("Item not found!")
```

جذب الله

To ask questions, please logon to the portal <https://alnafi.com/login/> and use your username and password. From within the portal you can ask questions. We will only answer questions if they are coming through the portal and not on email.

For any other queries please reach out on [info@alnafi.com](mailto:info@alnafi.com)



AL NAIFI,  
A company with a focus on education,  
wellbeing and renewable energy.

# Python Alpha 500

## Python Comprehensions

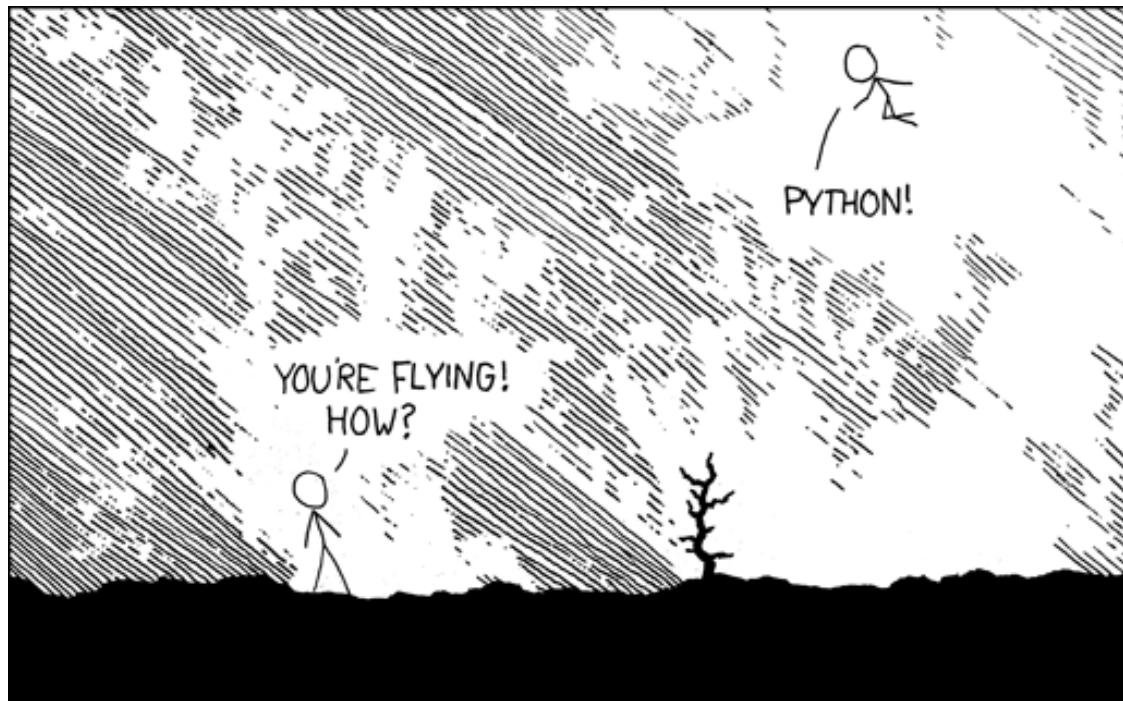
## Supplication at the start of the prayer after takbeer

اللَّهُمَّ رَبَّ جِبْرِيلَ ، وَمِيكَائِيلَ ، وَإِسْرَافِيلَ ، فَاطِرِ السَّمَاوَاتِ وَالْأَرْضِ ، عَالَمِ الْغَيْبِ وَالشَّهادَةِ  
أَنْتَ تَحْكُمُ بَيْنَ عِبَادِكَ فِيمَا كَانُوا فِيهِ يَخْتَلِفُونَ . اهْدِنِي لِمَا أَخْتَلِفَ فِيهِ مِنَ الْحَقِّ بِإِذْنِكَ ، إِنَّكَ  
تَهْدِي مَنْ تَشَاءُ إِلَى صِرَاطٍ مُسْتَقِيمٍ

O Allah, Lord of Jibra-eel, Meeka-eel and Israfeel (great angles), Creator of the heavens and the Earth, Knower of the seen and the unseen. You are the arbitrator between Your servants in that which they have disputed. Guide me to the truth by Your leave, in that which they have differed, for verily You guide whom You will to a straight path. Reference: Muslim 1/534

# Study, Rinse and Repeat

- Please subscribe to our [YouTube Channel](#) to be on top of your studies.
- Please logon to our website <https://alnafi.com/login/>
- Use your username and password to logon
- Please keep an eye on [zone@alnafi.com](mailto:zone@alnafi.com) emails
- Please review the videos of 30 minutes daily
- Please review the notes daily.
- Please take time for clearing up your mind and reflect on how things are proceeding in your studies.



I LEARNED IT LAST NIGHT! EVERYTHING IS SO SIMPLE!  
/ HELLO WORLD IS JUST  
print "Hello, world!"

I DUNNO...  
DYNAMIC TYPING?  
WHITESPACE?  
  
COME JOIN US!  
PROGRAMMING IS FUN AGAIN!  
IT'S A WHOLE NEW WORLD UP HERE!  
/ BUT HOW ARE YOU FLYING?

I JUST TYPED  
import antigravity  
THAT'S IT?  
  
/ ... I ALSO SAMPLED  
EVERYTHING IN THE  
MEDICINE CABINET  
FOR COMPARISON.  
/ BUT I THINK THIS  
IS THE PYTHON.

# Comprehensions

The Python language has a couple of methods for creating lists and dictionaries that are known as comprehensions. There is also a third type of comprehension for creating a Python known as comprehension.

You will find that the comprehension constructs build on the knowledge you have acquired from the previous classes as they contain loops and conditionals themselves.

Python's list comprehension is an example of the language's support for *functional programming concepts*. There's plenty of debate about the **best way** to develop program code: either procedurally, using functional programming techniques, or using object orientation.

# List comprehensions

List comprehensions in Python are very handy. They can also be a little hard to understand when and why you would use them.

List comprehensions tend to be harder to read than just using a simple for loop as well.

You may want to review the looping class before you continue. If you are ready, then we'll spend some time looking at how to construct list comprehensions and learn how they can be used.

# List comprehensions in one word

A list comprehension is basically a one line for loop that produces a Python list data structure. Here's a simple example:

```
x = [i for i in range(5)]  
x
```

Breaking the code down:

1. The code returns integers starting at 0 and ending at 4.
2. This can be useful if you need to create a list very quickly.

# Another list comprehensions examples

if [i for i in line if "SOME TERM" in i]:

Breaking the code down:

1. This code is parsing a file and looking for something in particular.

# Casting strings into integers 😊

```
x = ['1', '2', '3', '4', '5']
```

```
y = [int(i) for i in x]
```

```
y
```

Breaking the code down:

1. Mixing functions into the mix in this example.
2. In this code we are applying a function to every element in a list.
3. Such as when you need to cast a bunch of strings into integers.

# Strip the list

This sort of thing comes up more often than you'd think. I have also had to loop over a list of strings and call a string method, such as strip on them because they had all kinds of leading or ending white space:

```
myStrings = [s.strip() for s in myList]
```

# Nested list comprehensions

There are occasions where you need to create a nested list comprehension. Its like flatten multiple lists into one.

```
list = [[1,2,3], [4,5,6], [7,8,9]]  
[num for elem in list for num in elem]
```

# Dictionary Comprehensions

Dictionary comprehension is a method for transforming one dictionary into another dictionary. During this transformation, items within the original dictionary can be conditionally included in the new dictionary and each item can be transformed as needed.

# Dictionary Comprehensions examples

```
print( {i: str(i) for i in range(5)} )
```

Breaking the code down:

1. Basically it is creating an integer key and string value for each item in the range.

# Dictionary Comprehensions example continued...

```
my_dict = {1:"dog", 2:"cat", 3:"horse"}  
print( {value:key for key, value in my_dict.items()} )
```

Parking lot item as when we learn about class we will know more inshAllah.

More examples, assignments, code snippets, and projects will be shared in Python Beta so that what you learned in Python Primer and Python Alpha you are able to retain it inshAllah.

So don't confuse yourself with list comprehensions.

# Set Comprehensions

Set comprehensions are created in much the same way as dictionary comprehensions. Now a Python set is much like a mathematical set in that it doesn't have any repeated elements. You can create a normal set like this:

```
my_list = [1, 2, 2, 3, 4, 5, 5, 7, 8]  
my_set = set(my_list)  
my_set
```

Breaking the code down:

1. As you can see in this code a call has been set to remove the duplicates using set.

# Another set example

```
my_list = [1, 2, 2, 3, 4, 5, 5, 7, 8]
```

```
my_set = {x for x in my_list}
```

```
my_set
```

Breaking the code down:

1. Basically we changed the square brackets that a list comprehension uses to curly braces that the dictionary comprehensions utilizes. So there are multiple ways of using a set function.

# Wrapping up

Parking lot item as when we learn about class we will know more inshAllah.

More examples, assignments, code snippets, and projects will be shared in Python Beta so that what you learned in Python Primer and Python Alpha you are able to retain it inshAllah.

So don't confuse yourself with list comprehensions. Just try to use your imagination and fiddle around as more is on its way inshAllah.

جزاك الله

To ask questions, please logon to the portal <https://alnafi.com/login/> and use your username and password. From within the portal you can ask questions. We will only answer questions if they are coming through the portal and not on email.

For any other queries please reach out on [info@alnafi.com](mailto:info@alnafi.com)



AL NAIFI,  
A company with a focus on education,  
wellbeing and renewable energy.

# Python Alpha 500 easy version

## Python Comprehensions

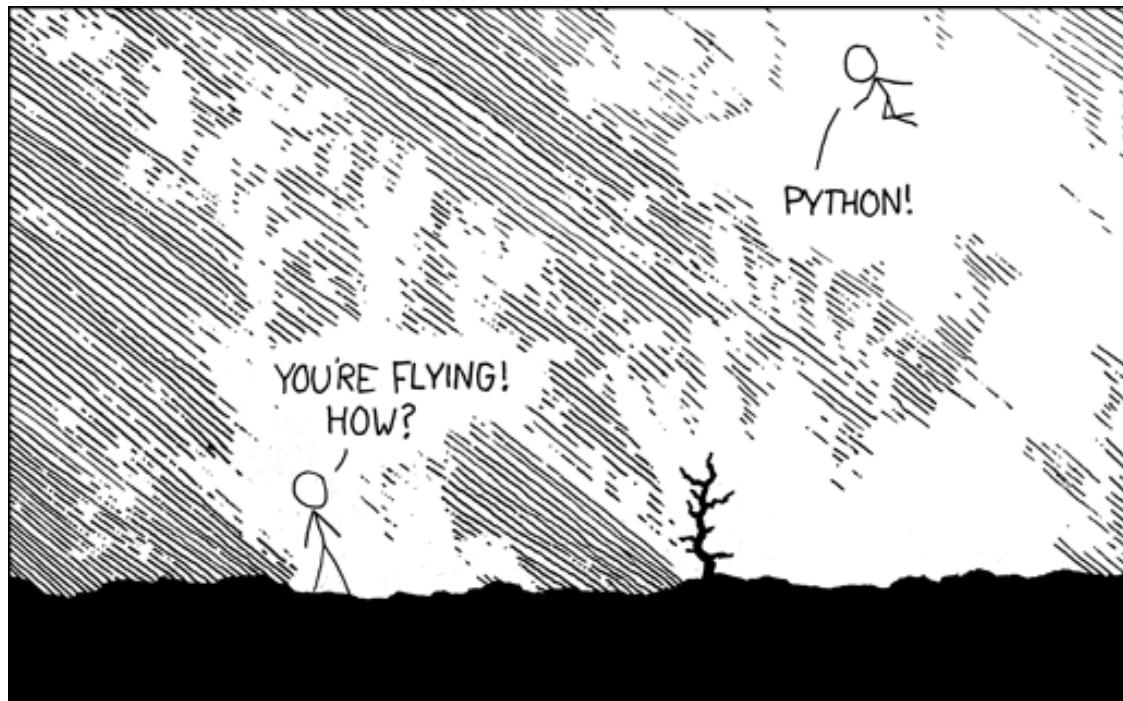
## Supplication at the start of the prayer after takbeer

اللَّهُمَّ رَبَّ جِبْرائِيلَ ، وَمِيكَائِيلَ ، وَإِسْرَافِيلَ ، فاطِرِ السَّمَاوَاتِ وَالْأَرْضِ ، عَالَمِ الْغَيْبِ وَالشَّهادَةِ  
أَنْتَ تَحْكُمُ بَيْنَ عِبَادِكَ فِيمَا كَانُوا فِيهِ يَخْتَلِفُونَ . اهْدِنِي لِمَا أَخْتَلِفَ فِيهِ مِنَ الْحَقِّ بِإِذْنِكَ ، إِنَّكَ  
تَهْدِي مَنْ تَشَاءُ إِلَى صِرَاطٍ مُسْتَقِيمٍ

O Allah, Lord of Jibra-eel, Meeka-eel and Israfeel (great angles), Creator of the heavens and the Earth, Knower of the seen and the unseen. You are the arbitrator between Your servants in that which they have disputed. Guide me to the truth by Your leave, in that which they have differed, for verily You guide whom You will to a straight path. Reference: Muslim 1/534

# Study, Rinse and Repeat

- Please subscribe to our [YouTube Channel](#) to be on top of your studies.
- Please logon to our website <https://alnafi.com/login/>
- Use your username and password to logon
- Please keep an eye on [zone@alnafi.com](mailto:zone@alnafi.com) emails
- Please review the videos of 30 minutes daily
- Please review the notes daily.
- Please take time for clearing up your mind and reflect on how things are proceeding in your studies.



I LEARNED IT LAST NIGHT! EVERYTHING IS SO SIMPLE!  
/ HELLO WORLD IS JUST  
print "Hello, world!"

I DUNNO...  
DYNAMIC TYPING?  
WHITESPACE?  
COME JOIN US!  
PROGRAMMING IS FUN AGAIN!  
IT'S A WHOLE NEW WORLD UP HERE!  
BUT HOW ARE YOU FLYING?

I JUST TYPED  
import antigravity  
THAT'S IT?  
/ ... I ALSO SAMPLED  
EVERYTHING IN THE  
MEDICINE CABINET  
FOR COMPARISON.  
BUT I THINK THIS  
IS THE PYTHON.

# Comprehensions

The Python language has a couple of methods for creating lists and dictionaries that are known as comprehensions. There is also a third type of comprehension for creating a Python known as comprehension.

You will find that the comprehension constructs build on the knowledge you have acquired from the previous classes as they contain loops and conditionals themselves.

Python's list comprehension is an example of the language's support for *functional programming concepts*. There's plenty of debate about the **best way** to develop program code: either procedurally, using functional programming techniques, or using object orientation.

# List comprehensions

List comprehensions in Python are very handy. They can also be a little hard to understand when and why you would use them.

List comprehensions tend to be harder to read than just using a simple for loop as well.

You may want to review the looping class before you continue. If you are ready, then we'll spend some time looking at how to construct list comprehensions and learn how they can be used.

# Another way to look at Comprehensions

List comprehensions are a unique way of quickly creating a list with Python.

If you find yourself using a loop along with `.append()` to create a list, list comprehensions are a good alternative!

Lets do some examples.

# List comprehensions examples!

## Good old python list

```
string = 'pakistan'  
mylist = []  
for letter in mystring:  
    mylist.append(letter)  
mylist
```

## List Comprehension

```
string = 'pakistan'  
mylist = [letter for letter in mystring]  
mylist
```

# Another list comprehension example!

```
mylist=[x for x in 'word']
```

```
mylist
```

```
mylist=[sharfoo for sharfoo in 'word']
```

```
mylist
```

# More examples!

```
mylist = [x for x in range(0,11)]
```

```
mylist
```

Or

```
mylist = [num for num in range(0,11)]
```

```
mylist
```

جزاك الله

To ask questions, please logon to the portal <https://alnafi.com/login/> and use your username and password. From within the portal you can ask questions. We will only answer questions if they are coming through the portal and not on email.

For any other queries please reach out on [info@alnafi.com](mailto:info@alnafi.com)



AL NAIFI,  
A company with a focus on education,  
wellbeing and renewable energy.

# Python Alpha 600a

Exceptions Handling in Python Theory  
Perspective.

اللَّهُمَّ إِنِّي أَصْبَحْتُ أَشْهِدُكَ وَأَشْهِدُ حَمَلَةَ عَرْشِكَ، وَمَلَائِكَتَكَ وَجَمِيعَ خَلْقَكَ، أَنْتَ أَنْتَ اللَّهُ لَا إِلَهَ إِلَّا  
أَنْتَ وَحْدَكَ لَا شَرِيكَ لَكَ، وَأَنَّ مُحَمَّداً عَبْدُكَ وَرَسُولُكَ

O Allaah, verily I have reached the morning and call on You, the bearers of Your throne, Your angels, and all of Your creation to witness that You are Allaah, none has the right to be worshipped except You, alone, without any partner and that Muhammad is Your Servant and Messenger.'

### **Hadith/Benefit**

Anas ibn Malik (R.A) the Prophet (S.A.W) said, "if anyone says (this Du'a) in the morning or in the evening Allah (SWT) will emancipate his fourth from hell, if anyone says twice Allah will emancipate his half, if anyone says it thrice, Allah will emancipate three-fourth and if he says four times, Allah will emancipate him from hell".  
[Sunan Abu Dawud, Book 41, Hadith 5051]

# Study, Rinse and Repeat

- Please subscribe to our [YouTube Channel](#) to be on top of your studies.
- Please logon to our website <https://alnafi.com/login/>
- Use your username and password to logon
- Please keep an eye on [zone@alnafi.com](mailto:zone@alnafi.com) emails
- Please review the videos of 30 minutes daily
- Please review the notes daily.
- Please take time for clearing up your mind and reflect on how things are proceeding in your studies.



STARECAT.COM

# Errors vs. Exceptions

An error and an exceptions are just different words that describe the same thing when we talk in the context of exceptions handling!

# Exceptions Handling of Python

What do you do when something bad happens in your program? Let's say you try to open a file, but you typed in the wrong path or you ask the user for information and they type in some garbage.

You don't want your program to crash, so you implement exception handling.

# Types of Exceptions Handling

In Python, the construct is usually wrapped in what is known as a try/except.

We will review the following exceptions handling:

- Common exception types
- Handling exceptions with try/except
- how try/except/ works
- How the else statement works in conjunction with the try/except

# Common Exceptions in Python

As per python documentation common exceptions are:

- **Exception** (this is what almost all the others are built off of)
- **AttributeError** - Raised when an attribute reference or assignment fails.
- **IOError** - Raised when an I/O operation (such as a print statement, the built-in open() function or a method of a file object) fails for an I/O-related reason, e.g., "file not found" or "disk full".

# Common Exceptions in Python continued.....

- **ImportError** - Raised when an import statement fails to find the module definition or when a from ... import fails to find a name that is to be imported.
- **IndexError** - Raised when a sequence subscript is out of range.
- **KeyError** - Raised when a mapping (dictionary) key is not found in the set of existing keys.
- **KeyboardInterrupt** - Raised when the user hits the interrupt key (normally Control-C or Delete).
- **NameError** - Raised when a local or global name is not found.

# Common Exceptions in Python continued...

- **OSError** - Raised when a function returns a system-related error.
- **SyntaxError** - Raised when the parser encounters a syntax error.
- **TypeError** - Raised when an operation or function is applied to an object of inappropriate type. The associated value is a string giving details about the type mismatch.
- **ValueError** - Raised when a built-in operation or function receives an argument that has the right type but an inappropriate value, and the situation is not described by a more precise exception such as IndexError.
- **ZeroDivisionError** - Raised when the second argument of a division or modulo operation is zero.

# Tons of other exceptions ☺

There are a lot of other exceptions as well, but you probably won't see them all that often. However, if you are interested, you can go and read all about them in the Python documentation.

<https://docs.python.org/3/>

جزاك الله

To ask questions, please logon to the portal <https://alnafi.com/login/> and use your username and password. We will circle back to you in 2-3 business days inshAllah.



AL NAFI,  
A company with a focus on education,  
wellbeing and renewable energy.

# Python Alpha 600b

Exceptions Handling in Python Theory  
Perspective.

اللَّهُمَّ إِنِّي أَصْبَحْتُ أَشْهِدُكَ وَأَشْهِدُ حَمَلَةَ عَرْشِكَ، وَمَلَائِكَتَكَ وَجَمِيعَ خَلْقَكَ، أَنْتَ أَنْتَ اللَّهُ لَا إِلَهَ إِلَّا  
أَنْتَ وَحْدَكَ لَا شَرِيكَ لَكَ، وَأَنَّ مُحَمَّداً عَبْدُكَ وَرَسُولُكَ

O Allaah, verily I have reached the morning and call on You, the bearers of Your throne, Your angels, and all of Your creation to witness that You are Allaah, none has the right to be worshipped except You, alone, without any partner and that Muhammad is Your Servant and Messenger.'

### **Hadith/Benefit**

Anas ibn Malik (R.A) the Prophet (S.A.W) said, "if anyone says (this Du'a) in the morning or in the evening Allah (SWT) will emancipate his fourth from hell, if anyone says twice Allah will emancipate his half, if anyone says it thrice, Allah will emancipate three-fourth and if he says four times, Allah will emancipate him from hell".  
[Sunan Abu Dawud, Book 41, Hadith 5051]

# Study, Rinse and Repeat

- Please subscribe to our [YouTube Channel](#) to be on top of your studies.
- Please logon to our website <https://alnafi.com/login/>
- Use your username and password to logon
- Please keep an eye on [zone@alnafi.com](mailto:zone@alnafi.com) emails
- Please review the videos of 30 minutes daily
- Please review the notes daily.
- Please take time for clearing up your mind and reflect on how things are proceeding in your studies.

MAKE SURE YOU'RE  
USING PYTHON 3,  
NOT PYTHON 2!



# Handling exceptions

Handling exceptions in Python is quite easy.

```
In [1]: 1 1/0
-----
-----
ZeroDivisionError                                     Traceback (
most recent call last)
<ipython-input-1-9e1622b385b6> in <module>()
----> 1 1/0

ZeroDivisionError: division by zero
```

# Catching errors with try and except

```
try:
```

```
    1/0
```

```
except ZeroDivisionError:
```

```
    print("You cannot divide by zero!")
```

In Math's we cannot divide anything with Zero. In Python, this operation will cause an error, as you can see in the first half of the example. To catch the error, we wrap the operation with a try/except statement.

# Bare excepts

```
try:
```

```
    1/0
```

```
except:
```

```
    print("You cannot divide by zero!")
```

It is not recommended! In Python, this is known as a bare except, which means it will catch any and all exceptions. The reason this is not recommended is that you don't know which exception you are catching. When you have something like `except ZeroDivisionError`, you are obviously trying to catch a division by zero error. In the code above, you cannot tell what you are trying to catch.

# Dictionary example

```
my_dict = {"a":1, "b":2, "c":3}  
try:  
    value = my_dict["d"]  
except KeyError:  
    print("That key does not exist!")
```

In this example, we create a 3-element dictionary. Then we try to access a key that is not in the dictionary. Because the key is not in the dictionary, it raises a `KeyError`, which we catch.

# List error message

```
my_list = [1, 2, 3, 4, 5]
try:
    my_list[6]
except IndexError:
    print("That index is not in the list!")
```

The example shows a list that has 5 items. We try to grab the 7th item from the index. Remember, Python lists are zero-based, so when you say [6], you're asking for the 7th item. Anyway, because there are only 5 items, it raises an IndexError, which we also catch.

# Catching multiple exceptions

```
my_dict = {"a":1, "b":2, "c":3}  
try:  
    value = my_dict["d"]  
except IndexError:  
    print("This index does not exist!")  
except KeyError:  
    print("This key is not in the dictionary!")  
except:  
    print("Some other error occurred!")
```

First we try to access a key that doesn't exist in the dictionary.

The try/except checks to see if you are catching a `KeyError`, which you are in the second except statement.

Then we are using a bare `except` which is not recommended but we will see it from time to time.

# Can we use () or Parenthesis

try:

```
    value = my_dict["d"]
except (IndexError, KeyError):
    print("An IndexError or KeyError occurred!")
```

Notice that in this example, we are putting the errors that we want to catch inside of parentheses. The problem with this method is that it's hard to tell which error has actually occurred, so the previous example is recommended.

جزاك الله

To ask questions, please logon to the portal <https://alnafi.com/login/> and use your username and password. We will circle back to you in 2-3 business days inshAllah.



AL NAFI,  
A company with a focus on education,  
wellbeing and renewable energy.

# Python Alpha 600c

Exceptions Handling in Python Theory  
Perspective.

اللَّهُمَّ إِنِّي أَصْبَحْتُ أَشْهِدُكَ وَأَشْهِدُ حَمَلَةَ عَرْشِكَ، وَمَلَائِكَتَكَ وَجَمِيعَ خَلْقَكَ، أَنْتَ أَنْتَ اللَّهُ لَا إِلَهَ إِلَّا  
أَنْتَ وَحْدَكَ لَا شَرِيكَ لَكَ، وَأَنَّ مُحَمَّداً عَبْدُكَ وَرَسُولُكَ

O Allaah, verily I have reached the morning and call on You, the bearers of Your throne, Your angels, and all of Your creation to witness that You are Allaah, none has the right to be worshipped except You, alone, without any partner and that Muhammad is Your Servant and Messenger.'

### **Hadith/Benefit**

Anas ibn Malik (R.A) the Prophet (S.A.W) said, "if anyone says (this Du'a) in the morning or in the evening Allah (SWT) will emancipate his fourth from hell, if anyone says twice Allah will emancipate his half, if anyone says it thrice, Allah will emancipate three-fourth and if he says four times, Allah will emancipate him from hell".  
[Sunan Abu Dawud, Book 41, Hadith 5051]

# Study, Rinse and Repeat

- Please subscribe to our [YouTube Channel](#) to be on top of your studies.
- Please logon to our website <https://alnafi.com/login/>
- Use your username and password to logon
- Please keep an eye on [zone@alnafi.com](mailto:zone@alnafi.com) emails
- Please review the videos of 30 minutes daily
- Please review the notes daily.
- Please take time for clearing up your mind and reflect on how things are proceeding in your studies.

I KNOW  
PYTHON  
WHAT'S YOUR  
SUPER  
POWER

# A statement “finally” to handle exceptions ☺

The finally statement is really easy to use.

# An example for finally

```
my_dict = {"a":1, "b":2, "c":3}  
try:  
    value = my_dict["d"]  
except KeyError:  
    print("A KeyError occurred!")  
finally:  
    print("The finally statement has executed!")
```

# try, except, or else!

The try/except statement also has an else clause. The else will only run if there are no errors raised.

# Example for else for handling exceptions

```
dict = {"a":1, "b":2, "c":3}  
try:  
    value = dict["a"]  
except KeyError:  
    print("A KeyError occurred!")  
else:  
    print("No error occurred!")
```

# Bring it all together try, except, else and finally

```
dict = {"a":1, "b":2, "c":3}
try:
    value = dict["a"]
except KeyError:
    print("A KeyError occurred!")
else:
    print("No error occurred!")
finally:
    print("The finally statement ran!")
```

جزاك الله

To ask questions, please logon to the portal <https://alnafi.com/login/> and use your username and password. We will circle back to you in 2-3 business days inshAllah.



AL NAIFI,  
A company with a focus on education,  
wellbeing and renewable energy.

# Python Alpha 700a

## Python and files

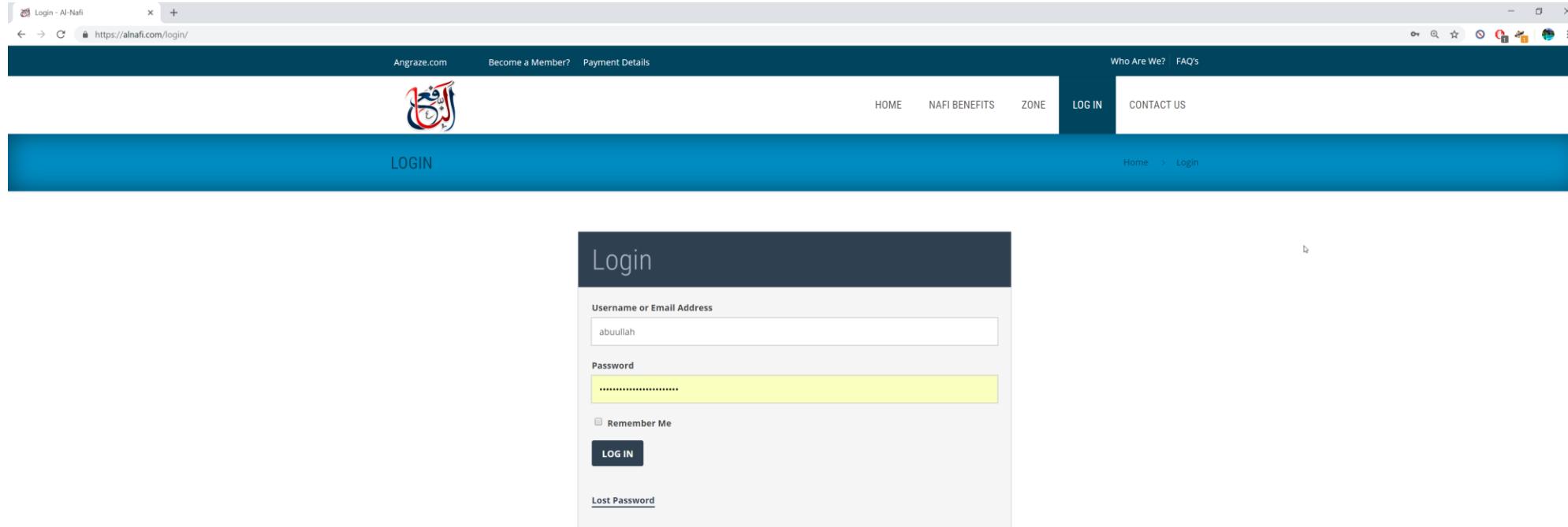
اللَّهُمَّ عَافِنِي فِي بَدَنِي، اللَّهُمَّ عَافِنِي فِي سَمْعِي، اللَّهُمَّ عَافِنِي فِي بَصَرِي، لَا إِلَهَ إِلَّا أَنْتَ. اللَّهُمَّ إِنِّي أَعُوذُ بِكَ مِنَ الْكُفْرِ، وَالْفَقْرِ، وَأَعُوذُ بِكَ مِنْ عَذَابِ الْقَبْرِ، لَا إِلَهَ إِلَّا أَنْتَ

O Allah, grant my body health, O Allaah, grant my hearing health, O Allaah, grant my sight health. None has the right to be worshipped except You. O Allaah, I take refuge with You from disbelief and poverty, and I take refuge with You from the punishment of the grave. None has the right to be worshipped except You.'

# Study, Rinse and Repeat

- Please subscribe to our [YouTube Channel](#) to be on top of your studies.
- Please logon to our website <https://alnafi.com/login/>
- Use your username and password to logon
- Please keep an eye on [zone@alnafi.com](mailto:zone@alnafi.com) emails
- Please review the videos of 30 minutes daily
- Please review the notes daily.
- Please take time for clearing up your mind and reflect on how things are proceeding in your studies.

# Members Please connect to our portal

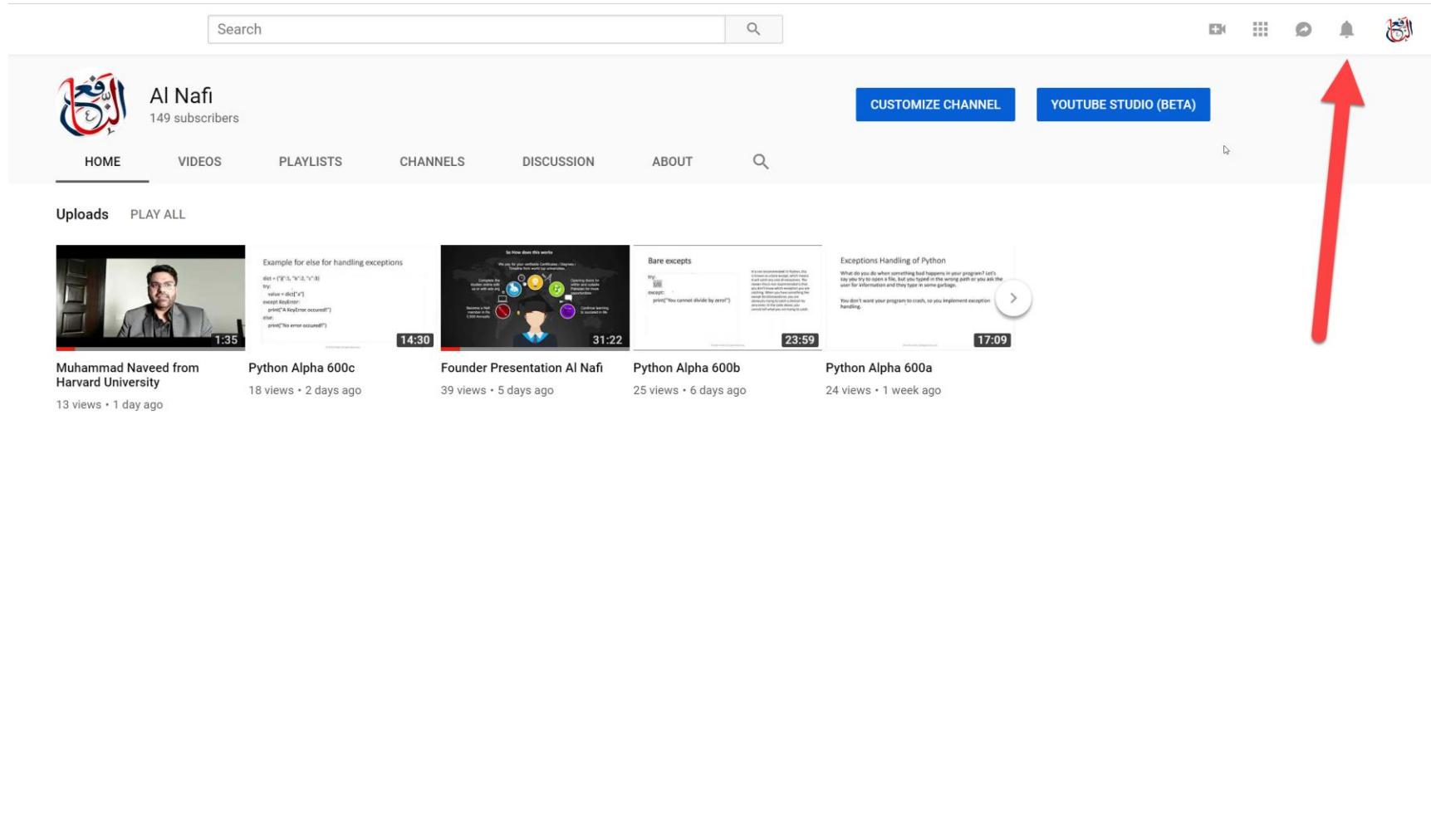


The screenshot shows a web browser window for 'Login - Al-Nafi' at the URL <https://alnafi.com/login/>. The page has a dark blue header with the Al-Nafi logo, navigation links for 'Angraze.com', 'Become a Member?', 'Payment Details', 'Who Are We?', 'FAQ's', and a 'LOG IN' button. Below the header is a teal-colored main content area with the word 'LOGIN' in white. The central part of the page is a 'Login' form with fields for 'Username or Email Address' (containing 'abuullah') and 'Password' (containing several dots). There is also a 'Remember Me' checkbox and a 'LOG IN' button. At the bottom of the form is a link to 'Lost Password'.



The footer of the website features a dark blue background. On the left, there is a logo and a brief description: 'Al-Nafi company with a focus on education, wellbeing and renewable energy in Pakistan.' In the center, there are four columns: 'POPULAR COURSES' (Cyber Security, IOT, Data Science, Cloud Computing), 'EXPLORE' (Home, Nafi Benefits, Zone, Log In, Contact us), 'GET IN TOUCH WITH US' (info@nafi.com, +1 647-680-1508, +92345-2494147, social media icons for Facebook, YouTube, and Twitter), and a copyright notice: '© 2018 Al-Nafi. All Rights Reserved.'

# Our YouTube channel



# Working with files in Python

The building function **open**

That we can use to open a file for reading or using it in the context of  
ML, DL, AI and offensive cyber security

# Using open in python to read a file

```
x = open(r"D:\PythonAlpha\test.txt","r")
```

# So what's up with r

Raw string versus regular string what's the difference ?

```
print("D:\\PythonAlpha\\test.txt")
```

Vs

```
print(r"D:\\PythonAlpha\\test.txt")
```

# Reading a file and see the data

```
x = open(r"D:\PythonAlpha\test.txt","r")
data = x.read()
print(data)
x.close()
```

# Reading a line in the file

```
x = open(r"D:\PythonAlpha\test.txt","r")
data = x.readline()
print(data)
x.close()
```

# Reading all the lines

```
x = open(r"D:\PythonAlpha\test.txt","r")
data = x.readlines()
print(data)
x.close()
```

# Reading files piece by piece

```
x = open(r"D:\PythonAlpha\test.txt","r")
for line in x:
    print(line)
x.close()
```

# Writing files in Python

For writing files we use “w” and “wb”

جزاك الله

To ask questions, please logon to the portal <https://alnafi.com/login/> and use your username and password. We will circle back to you in 2-3 business days inshAllah.



AL NAIFI,  
A company with a focus on education,  
wellbeing and renewable energy.

# Python Alpha 700b

## Python and files

# Hadith of the day

The messenger of Allah peace be upon him said:

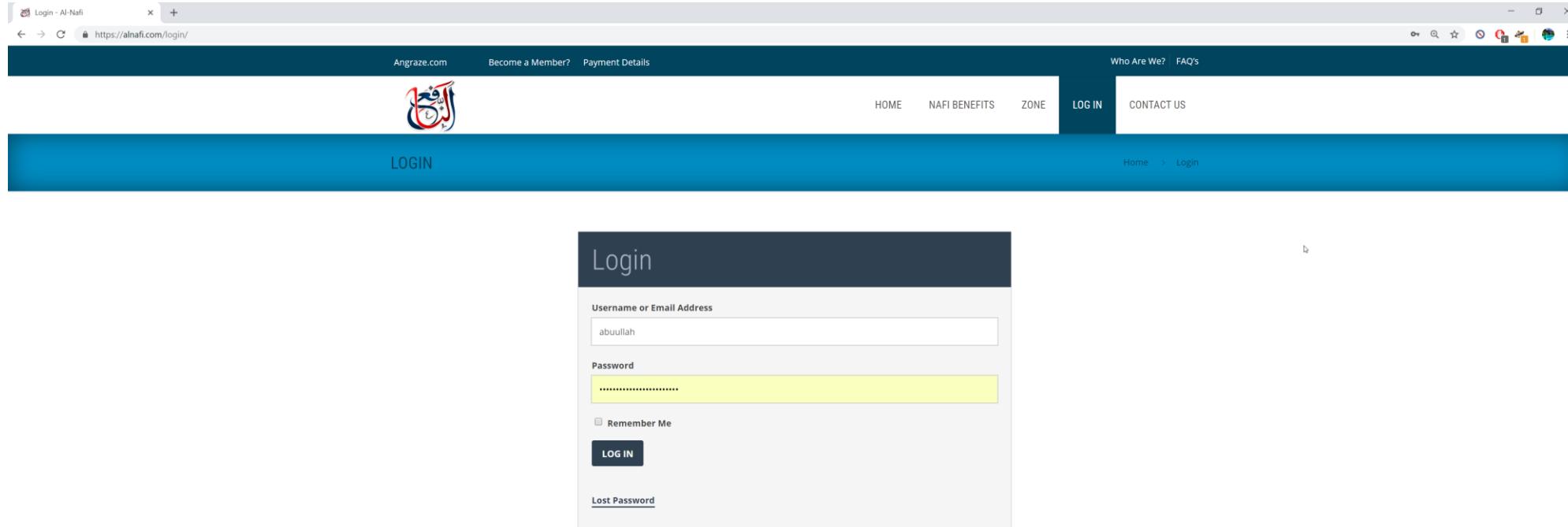
**“Seeking ‘Ilm is incumbent on every Muslim.”**

(Ibn Hajar al-Asqalani, Al-Matalib al-Aliyah)

# Study, Rinse and Repeat

- Please subscribe to our [YouTube Channel](#) to be on top of your studies.
- Please logon to our website <https://alnafi.com/login/>
- Use your username and password to logon
- Please keep an eye on [zone@alnafi.com](mailto:zone@alnafi.com) emails
- Please review the videos of 30 minutes daily
- Please review the notes daily.
- Please take time for clearing up your mind and reflect on how things are proceeding in your studies.

# Members Please connect to our portal

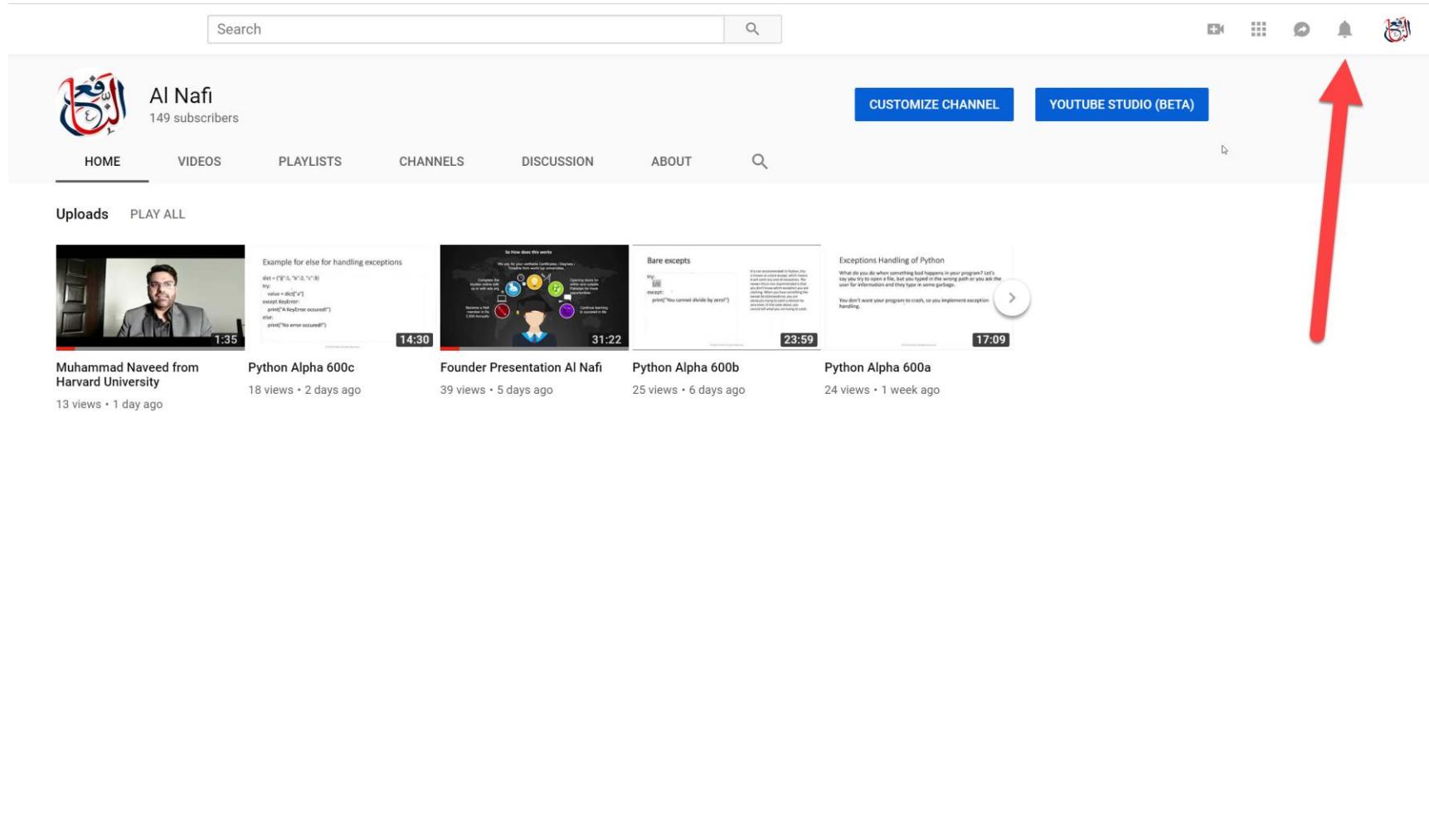


The screenshot shows a web browser window for 'Login - Al-Nafi' at the URL <https://alnafi.com/login/>. The page has a dark blue header with the Al-Nafi logo, navigation links for 'Angraze.com', 'Become a Member?', 'Payment Details', 'Who Are We?', 'FAQ's', and a 'LOG IN' button. Below the header is a teal-colored main content area with the word 'LOGIN' in white. The central part of the page is a 'Login' form with fields for 'Username or Email Address' (containing 'abuullah') and 'Password' (containing several dots). There is also a 'Remember Me' checkbox and a 'LOG IN' button. At the bottom of the form is a link to 'Lost Password'.



The footer of the website features a dark blue background. On the left, there is a logo and a brief description: 'Al-Nafi company with a focus on education, wellbeing and renewable energy in Pakistan.' In the center, there are four columns: 'POPULAR COURSES' (Cyber Security, IOT, Data Science, Cloud Computing), 'EXPLORE' (Home, Nafi Benefits, Zone, Log In, Contact us), 'GET IN TOUCH WITH US' (info@nafi.com, +1 647-680-1508, +92345-2494147, social media icons for Facebook, YouTube, and Twitter), and a copyright notice: '© 2018 Al-Nafi. All Rights Reserved.'

# Our YouTube channel



# Writing files in Python with w

When using "w" mode, if the file already exists, it will be overwritten with no warning!

# Writing Files in Python

```
x = open(r"D:\PythonAlpha\test.txt","w")
x.write("This is another line")
x.close()
```

# Assignment to be submitted on the portal

Please submit the last assignment as to what you have learned so far from Primer to up until Python Alpha 700 lectures.

جزاك الله

To ask questions, please logon to the portal <https://alnafi.com/login/> and use your username and password. We will circle back to you in 2-3 business days inshAllah.



AL NAIFI,  
A company with a focus on education,  
wellbeing and renewable energy.

# Python Alpha 800

## Importing code Python

# Hadith of the day

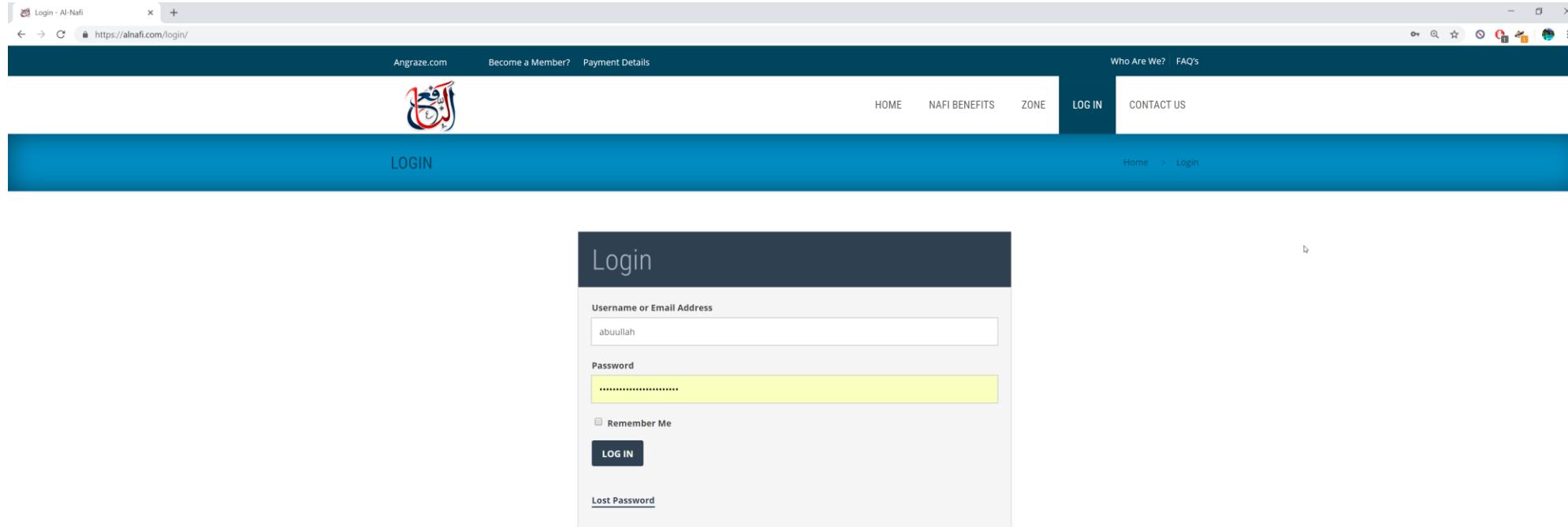
Jabir Ibn “Abd Allah narrated, the messenger of Allah (peace be upon him) said:

**“People are of different substances, the best of them in the Jahiliyah are the best in Islam, if they gain ‘ilm(knowledge).”**

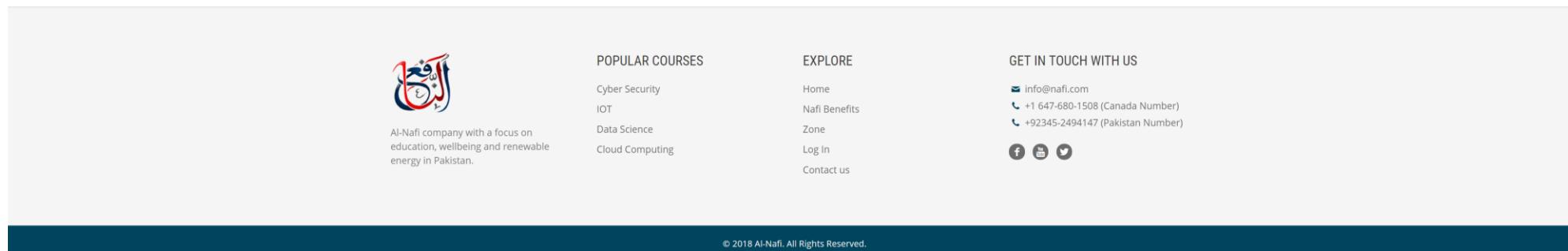
# Study, Rinse and Repeat

- Please subscribe to our [YouTube Channel](#) to be on top of your studies.
- Please logon to our website <https://alnafi.com/login/>
- Use your username and password to logon
- Please keep an eye on [zone@alnafi.com](mailto:zone@alnafi.com) emails
- Please review the videos of 30 minutes daily
- Please review the notes daily.
- Please take time for clearing up your mind and reflect on how things are proceeding in your studies.

# Members Please connect to our portal

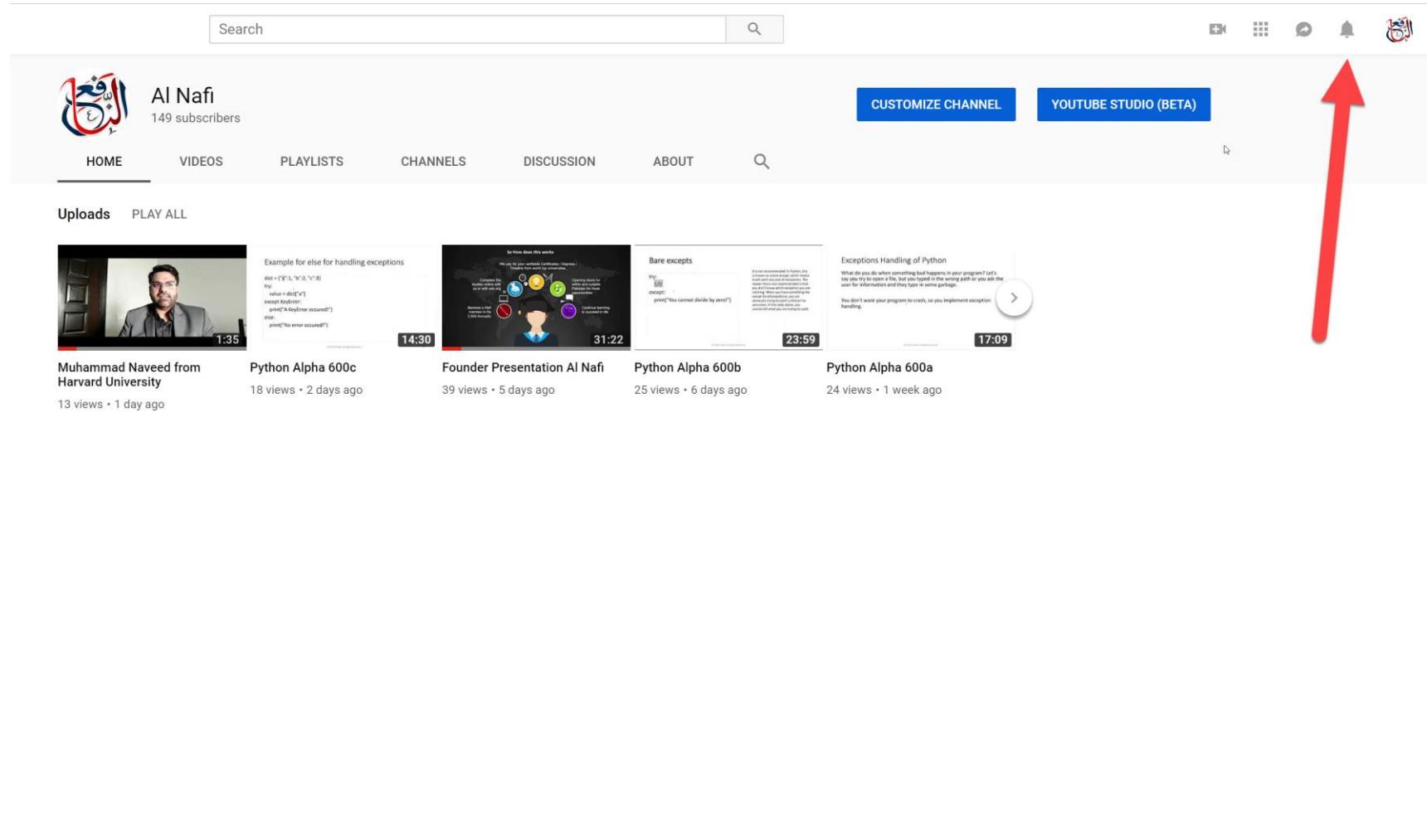


The screenshot shows a web browser window for 'Login - Al-Nafi' at the URL <https://alnafi.com/login/>. The page has a dark blue header with the Al-Nafi logo, navigation links for 'Angraze.com', 'Become a Member?', 'Payment Details', 'Who Are We?', 'FAQ's', and a 'LOG IN' button. Below the header is a teal banner with the word 'LOGIN'. The main content area is titled 'Login' and contains fields for 'Username or Email Address' (with 'abuullah' typed in) and 'Password' (with several dots). There is a 'Remember Me' checkbox and a 'LOG IN' button. At the bottom of the login form is a link for 'Lost Password'.



The footer of the website features the Al-Nafi logo and a brief description: 'Al-Nafi company with a focus on education, wellbeing and renewable energy in Pakistan.' To the right, there are four columns: 'POPULAR COURSES' (Cyber Security, IOT, Data Science, Cloud Computing), 'EXPLORE' (Home, Nafi Benefits, Zone, Log In, Contact us), 'GET IN TOUCH WITH US' (info@nafi.com, +1 647-680-1508, +92345-2494147, social media icons for Facebook, YouTube, and Twitter), and a copyright notice: '© 2018 Al-Nafi. All Rights Reserved.'

# Our YouTube channel



# Importing code in Python!

By default python comes with a pre-packaged code. These pieces of code are known as modules and packages.

A module is a single importable code whereas a package is made up of two or more modules.

A package can be imported the same way as module is. When ever you save a python script that you created, then you have a module at your hands.

# Import this

Python provides the import keyword for importing modules.

We use

`import this`

# Zen of Python

This is known as “Easter egg” in python known as the “Zen of Python”.

This is also known as unofficial best practices for python. The this module doesn’t actually do anything.

# math module

```
import math
```

```
math.sqrt(4)
```

to use the math module the syntax is

```
module_name.method_name(argument)
```

جزاك الله

To ask questions, please logon to the portal <https://alnafi.com/login/> and use your username and password. We will circle back to you in 2-3 business days inshAllah.



AL NAIFI,  
A company with a focus on education,  
wellbeing and renewable energy.

# Python Alpha 800

## Importing code Python

# Forty Hadith on the importance of Knowledge, learning and Teaching.

## Hadith # 3

Hudhaifah Ibn al-Yaman (May Allah be pleased with him) narrated, the Messenger of Allah (Peace be upon him) said:

“The excess of ilm is better than the excess of Ibadah, and the best of your religion is the wara.”

(at-Tabarani, Al-Aswar; Al-Bazzar)

Wara = means piety, self restrain.

# Study, Rinse and Repeat

- Please subscribe to our [YouTube Channel](#) to be on top of your studies.
- Please logon to our website <https://alnafi.com/login/>
- Use your username and password to logon
- Please keep an eye on [zone@alnafi.com](mailto:zone@alnafi.com) emails
- Please review the videos of 30 minutes daily
- Please review the notes daily.
- Please take time for clearing up your mind and reflect on how things are proceeding in your studies.

# Members Please connect to our portal

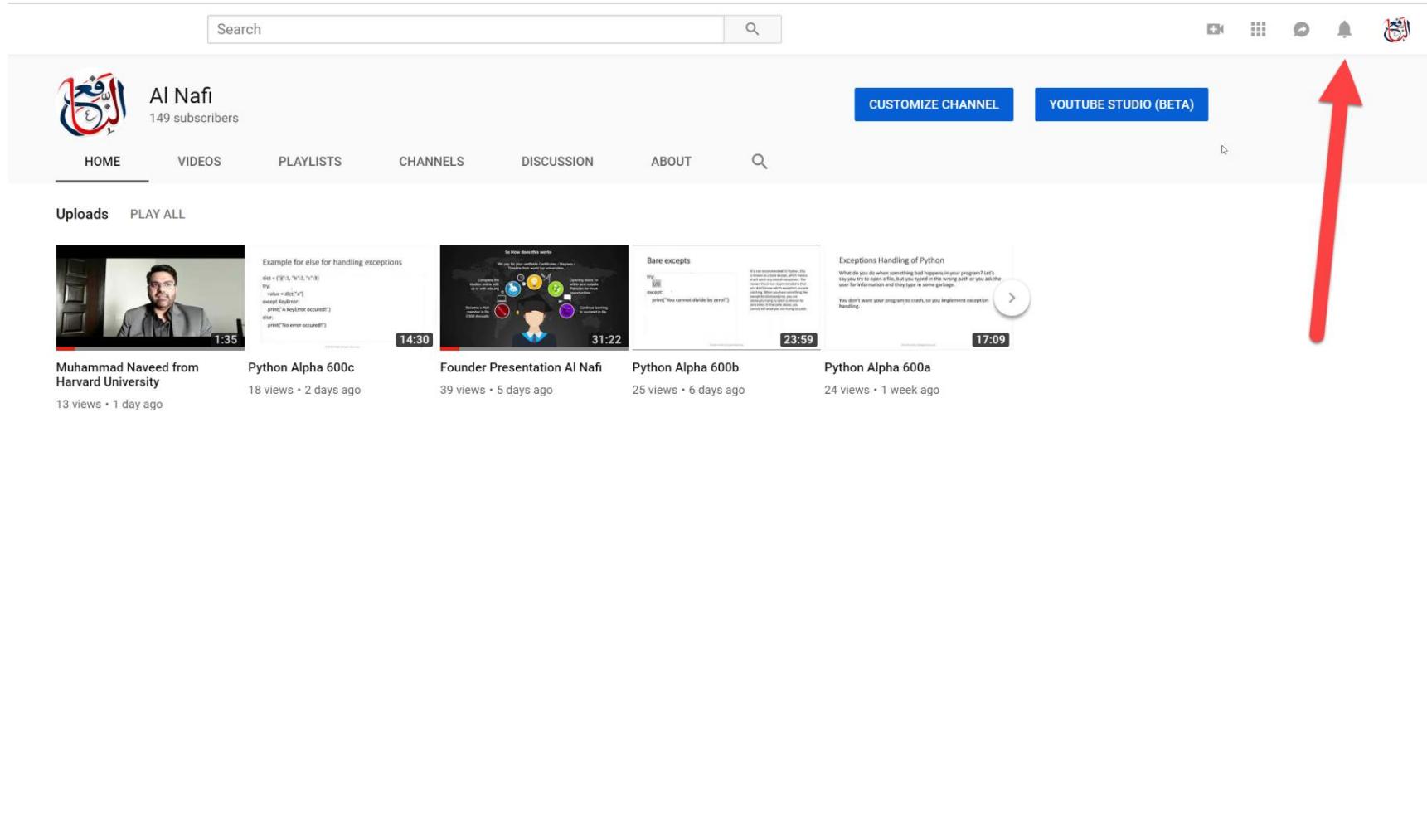


A screenshot of a web browser showing the login page for 'Angraze.com'. The URL in the address bar is <https://alnafi.com/login/>. The page has a dark blue header with navigation links for 'HOME', 'NAFI BENEFITS', 'ZONE', 'LOG IN', and 'CONTACT US'. Below the header is a large teal banner with the word 'LOGIN' in white. The main content area is titled 'Login' and contains fields for 'Username or Email Address' (with 'abuullah' typed in) and 'Password' (with several dots). There is also a 'Remember Me' checkbox and a 'LOG IN' button. At the bottom of the login form is a link to 'Lost Password'.



The footer of the website features a dark blue background. On the left, there is a logo with the text 'Al-Nafi' in a stylized font. Below the logo is a brief description: 'Al-Nafi company with a focus on education, wellbeing and renewable energy in Pakistan.' To the right of the logo are four columns of links: 'POPULAR COURSES' (Cyber Security, IOT, Data Science, Cloud Computing), 'EXPLORE' (Home, Nafi Benefits, Zone, Log In, Contact us), and 'GET IN TOUCH WITH US' (info@nafi.com, +1 647-680-1508, +92345-2494147, social media icons for Facebook, YouTube, and Twitter). At the very bottom of the footer is a copyright notice: '© 2018 Al-Nafi. All Rights Reserved.'

# Our YouTube channel



# Using from to import

Another way to import a function is to use from

```
from math import sqrt  
sqrt(16)
```

This works pretty much the same as the previous command we discussed.

# Using import vs. from import

`module_name.method_name(argument)`

vs

from the math module, import the sqrt function

Lets discuss this shall we

# Comparing two import functions

`module_name.method_name(argument)`

`math.sqrt(4)`

vs

`from the math module import the sqrt function`

`from math import sqrt`

# Importing multiple functions

```
from math import pi, sqrt
```

In the above example we are importing both pi and sqrt. If we tried to access pi it will only give us a value and not a function.

Point to note is that when we use an import, we may also end up importing a value a function or even another module.

# Import everything!

There is another way in python to import all the functions and values from a module as well.

This is a not recommended as it can contaminate our **namespace**.

A **namespace** is where of our variables live during the life of the program.

# Examples

```
from math import sqrt  
sqrt = 5
```

\* mean wild card  
which means  
everything ☺

---

```
-----  
from math import*  
sqrt = 5  
sqrt(16)
```

جزاك الله

To ask questions, please logon to the portal <https://alnafi.com/login/> and use your username and password. We will circle back to you in 2-3 business days inshAllah.



AL NAIFI,  
A company with a focus on education,  
wellbeing and renewable energy.

# Python Alpha 900

## Functions in Python

# Forty Hadith on the importance of Knowledge, learning and Teaching.

## Hadith # 4

Ibn' Abbas reported, the Messenger of Allah peace be upon him said to a group of his companions:

“When you pass by the meadows of Paradise indulge freely in it!  
They said: O Messenger of Allah! What are the meadows of  
Paradise?

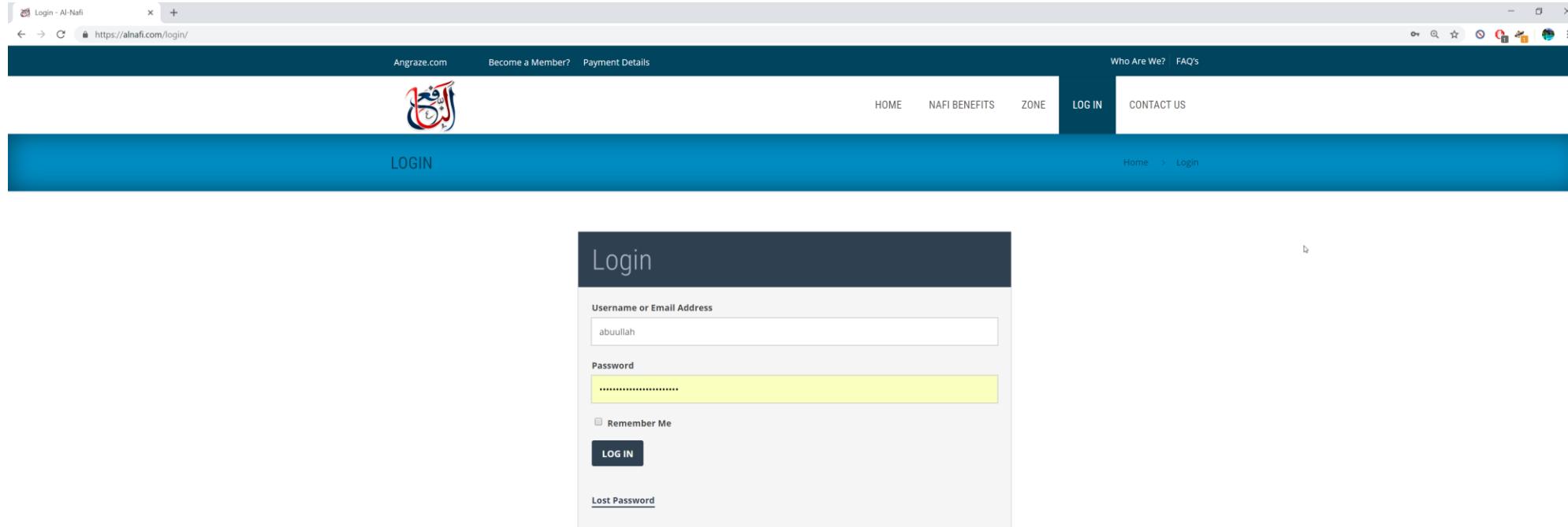
He Said: The circles of “ilm.”

At- Tabarani, Al-Mu’jum Al-Kabir

# Study, Rinse and Repeat

- Please subscribe to our [YouTube Channel](#) to be on top of your studies.
- Please logon to our website <https://alnafi.com/login/>
- Use your username and password to logon
- Please keep an eye on [zone@alnafi.com](mailto:zone@alnafi.com) emails
- Please review the videos of 30 minutes daily
- Please review the notes daily.
- Please take time for clearing up your mind and reflect on how things are proceeding in your studies.

# Members Please connect to our portal

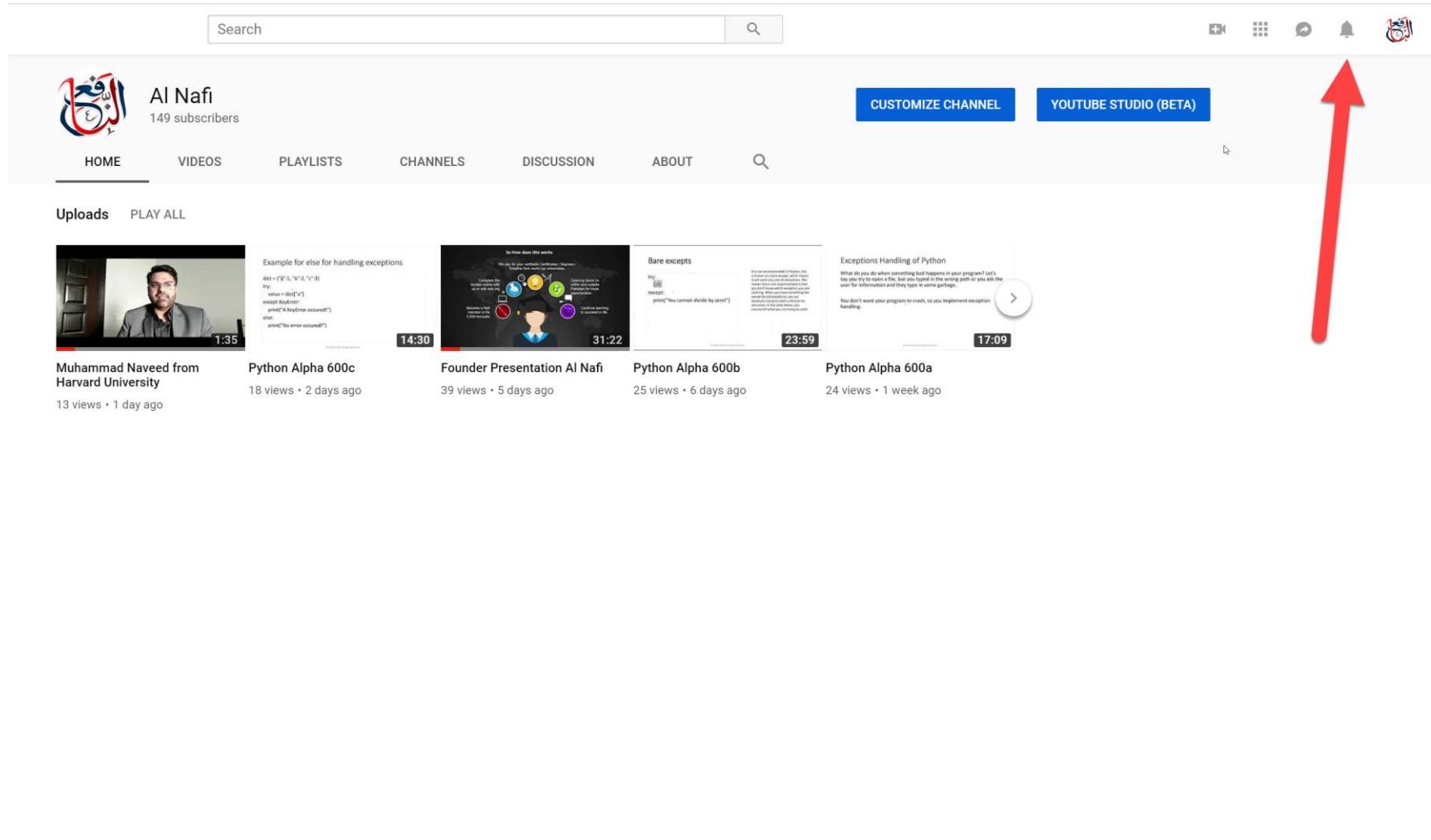


The screenshot shows a web browser window for 'Login - Al-Nafi' at the URL <https://alnafi.com/login/>. The page has a dark blue header with the Al-Nafi logo, navigation links for 'Angraze.com', 'Become a Member?', 'Payment Details', 'Who Are We?', 'FAQ's', and a 'LOG IN' button. Below the header is a teal banner with the word 'LOGIN'. The main content area is titled 'Login' and contains fields for 'Username or Email Address' (with 'abuullah' typed in) and 'Password' (with several dots). There is a 'Remember Me' checkbox and a 'LOG IN' button. At the bottom of the login form is a link to 'Lost Password'.



The footer of the website features the Al-Nafi logo and a brief description: 'Al-Nafi company with a focus on education, wellbeing and renewable energy in Pakistan.' To the right, there are four columns: 'POPULAR COURSES' (Cyber Security, IOT, Data Science, Cloud Computing), 'EXPLORE' (Home, Nafi Benefits, Zone, Log In, Contact us), 'GET IN TOUCH WITH US' (info@nafi.com, +1 647-680-1508, +92345-2494147, social media icons for Facebook, YouTube, and Twitter), and a copyright notice: '© 2018 Al-Nafi. All Rights Reserved.'

# Our YouTube channel



# Functions in Python

So what is a function in python?

A function is a block of code that starts with the **def** keyword, followed by a name for the functions and a colon.

```
def a_function():  
    print("we just created a function!")
```

# An Empty function AKA the stub

```
def empty_function():
    pass
```

The pass statement is known as a null operation, which means that when pass is executed, nothing happens.

# Now adding arguments to a function

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

Or

```
add(1,2)
```

Or

```
add(1)
```

# Examples continued

add(a=2, b=3)

total = add(b=4, a=5)

print(total)

So what happens when we add c ?

add(c=5, d=2)

جزاك الله

To ask questions, please logon to the portal <https://alnafi.com/login/> and use your username and password. We will circle back to you in 2-3 business days inshAllah.



AL NAIFI,  
A company with a focus on education,  
wellbeing and renewable energy.

# Python Alpha 900a

## Functions in Python

# Forty Hadith on the importance of Knowledge, learning and Teaching.

# Hadith # 5

Abu Bakrah reported, I heard the Prophet Peace be upon him saying:

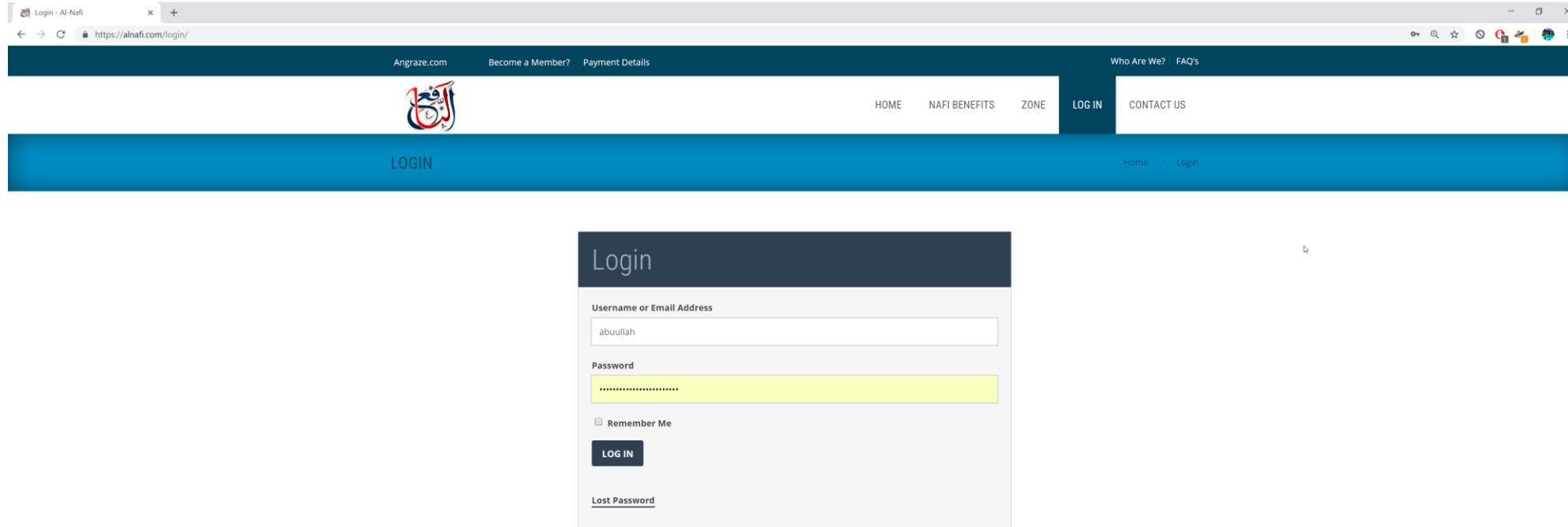
“ You should be a scholar, or a student, or a listener, or a lover of ‘Ilm and scholars, and you should not be the fifth which makes you perish. ‘Ata, said, Mis’ar said to me: you added a fifth point we do not have. The fifth point is: to hate ‘Ilm and its people.”

(AT Tabarani; Al Bazzar)

# Study, Rinse and Repeat

- Please subscribe to our [YouTube Channel](#) to be on top of your studies.
- Please logon to our website <https://alnafi.com/login/>
- Use your username and password to logon
- Please keep an eye on [zone@alnafi.com](mailto:zone@alnafi.com) emails
- Please review the videos of 30 minutes daily
- Please review the notes daily.
- Please take time for clearing up your mind and reflect on how things are proceeding in your studies.

# Members Please connect to our portal

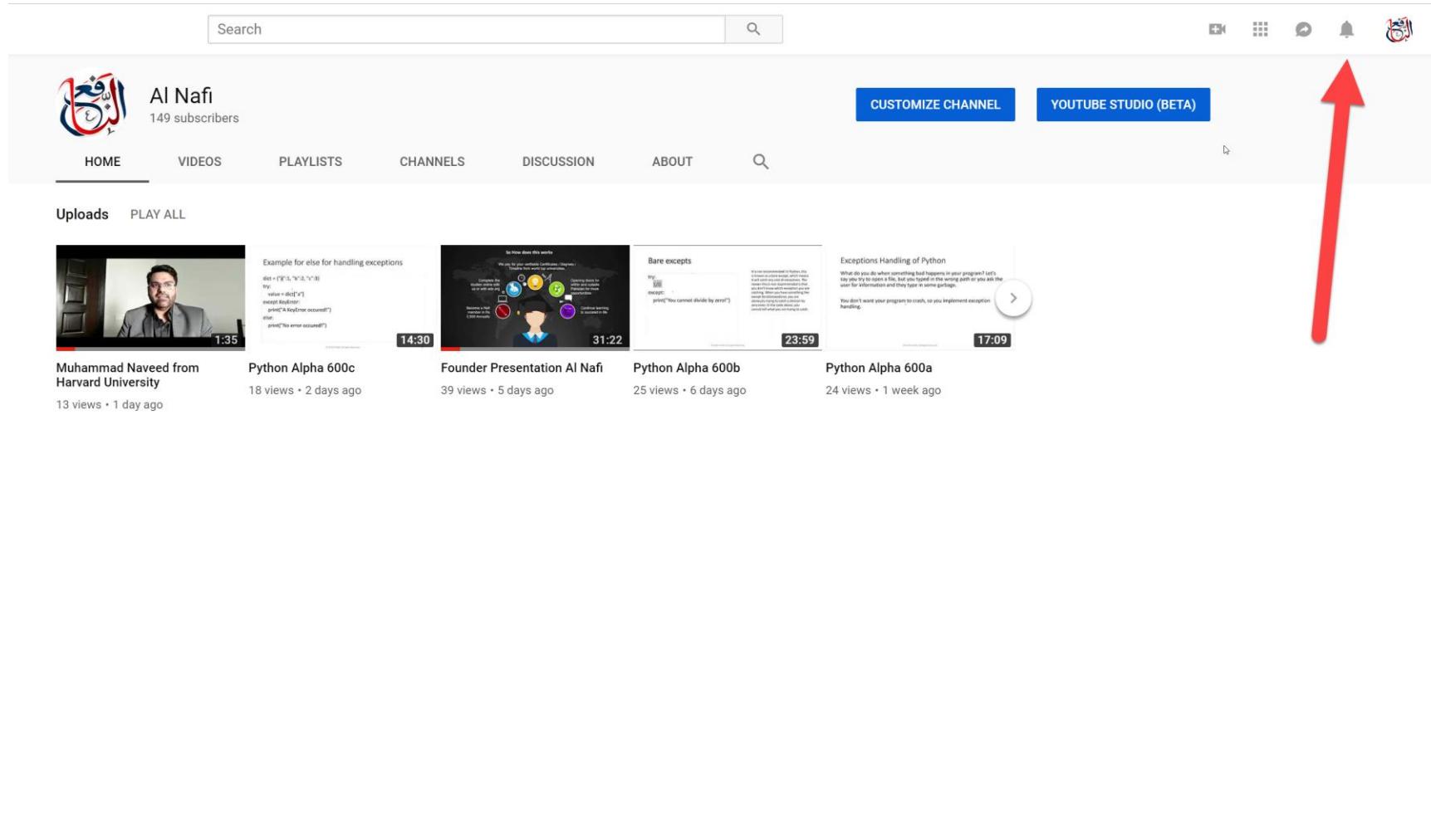


The screenshot shows a web browser window for 'Login - Al-Nafi' at the URL <https://alnafi.com/login/>. The page has a dark blue header with the Al-Nafi logo, navigation links for 'Angraze.com', 'Become a Member?', 'Payment Details', 'Who Are We?', 'FAQ's', and a 'LOG IN' button. Below the header is a teal banner with the word 'LOGIN'. The main content area is titled 'Login' and contains fields for 'Username or Email Address' (with 'abuullah' typed in) and 'Password' (with several dots). There is a 'Remember Me' checkbox and a 'LOG IN' button. At the bottom of the login form is a link for 'Lost Password'.



The footer of the website features the Al-Nafi logo and a brief description: 'Al-Nafi company with a focus on education, wellbeing and renewable energy in Pakistan.' To the right, there are four columns: 'POPULAR COURSES' (Cyber Security, IOT, Data Science, Cloud Computing), 'EXPLORE' (Home, Nafi Benefits, Zone, Log In, Contact us), 'GET IN TOUCH WITH US' (info@nafi.com, +1 647-680-1508, +92345-2494147, social media icons for Facebook, YouTube, and Twitter), and a copyright notice: '© 2018 Al-Nafi. All Rights Reserved.'

# Our YouTube channel



# Keyword Arguments

Functions also need keyword arguments!

They can accept both regular and keyword arguments.

What this means is that we can specify which keyword are which and pass them in.

For example

```
def keyword_function(a=1 , b=2):  
    return a+b
```

```
keyword_function(b=4, a=5)
```

# Default argument keyword!

Now this is also known as default argument

```
def keyword_function(a=1 , b=2):  
    return a+b
```

If we type

```
keyword_function()
```

# Mixed functions 😊

Lets create a function which has both regular and keyword arguments

```
def mixed_function(a, b=2, c=3):  
    return a+b+c
```

# Example continued!

`mixed_function(b=4, c=5)`

But if we try this

`mixed_function(1, b=4, c=5)`

And if we try `mixed_function(1)`

جزاك الله

To ask questions, please logon to the portal <https://alnafi.com/login/> and use your username and password. We will circle back to you in 2-3 business days inshAllah.



AL NAIFI,  
A company with a focus on education,  
wellbeing and renewable energy.

# Python Alpha 900b

## Functions in Python

# Forty Hadith on the importance of Knowledge, learning and Teaching.

# Hadith # 6

Abu Hurairah narrated, the Messenger of Allah peace be upon him said,

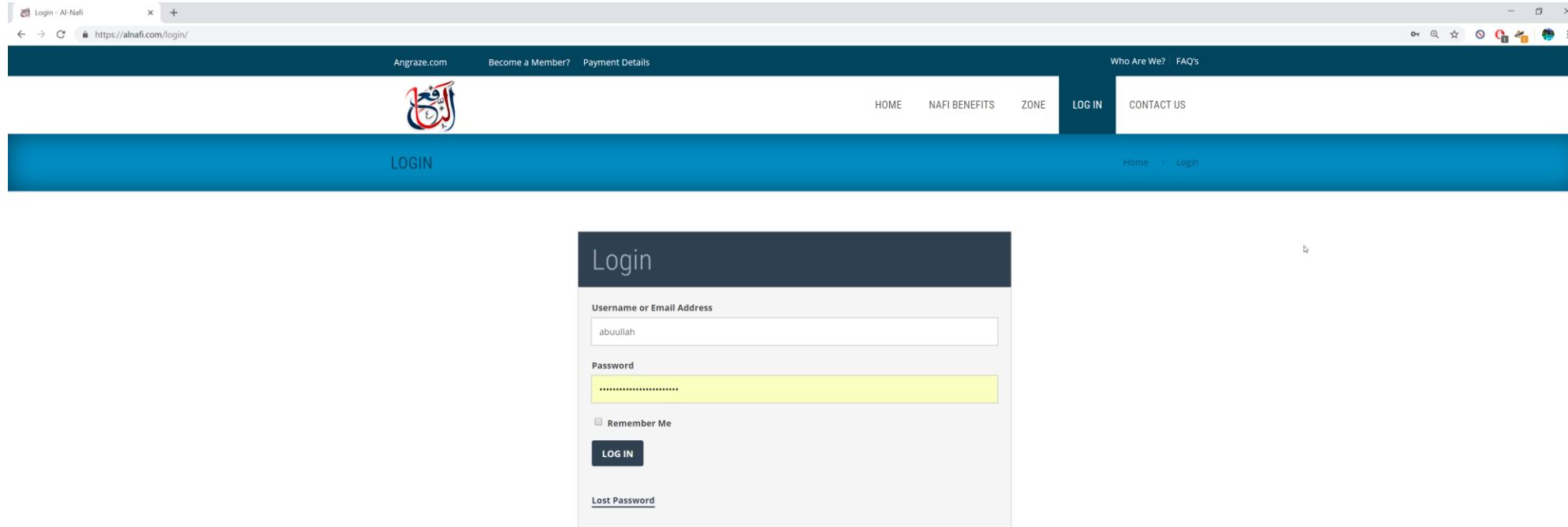
“The most honest men of future generations will carry this ‘Ilm, (i.e. Hadiath) they will purify it from the falsification of the extremists, and the assumptions of the liars, and the misinterpretation of the fools.”

(Al-Khatib al-Baghdadi, Sharaf As/had al-Hadiath; at-Tabrizi,  
Mishkat al-Masabih)

# Study, Rinse and Repeat

- Please subscribe to our [YouTube Channel](#) to be on top of your studies.
- Please logon to our website <https://alnafi.com/login/>
- Use your username and password to logon
- Please keep an eye on [zone@alnafi.com](#) emails
- Please review the videos of 30 minutes daily
- Please review the notes daily.
- Please take time for clearing up your mind and reflect on how things are proceeding in your studies.

# Members Please connect to our portal

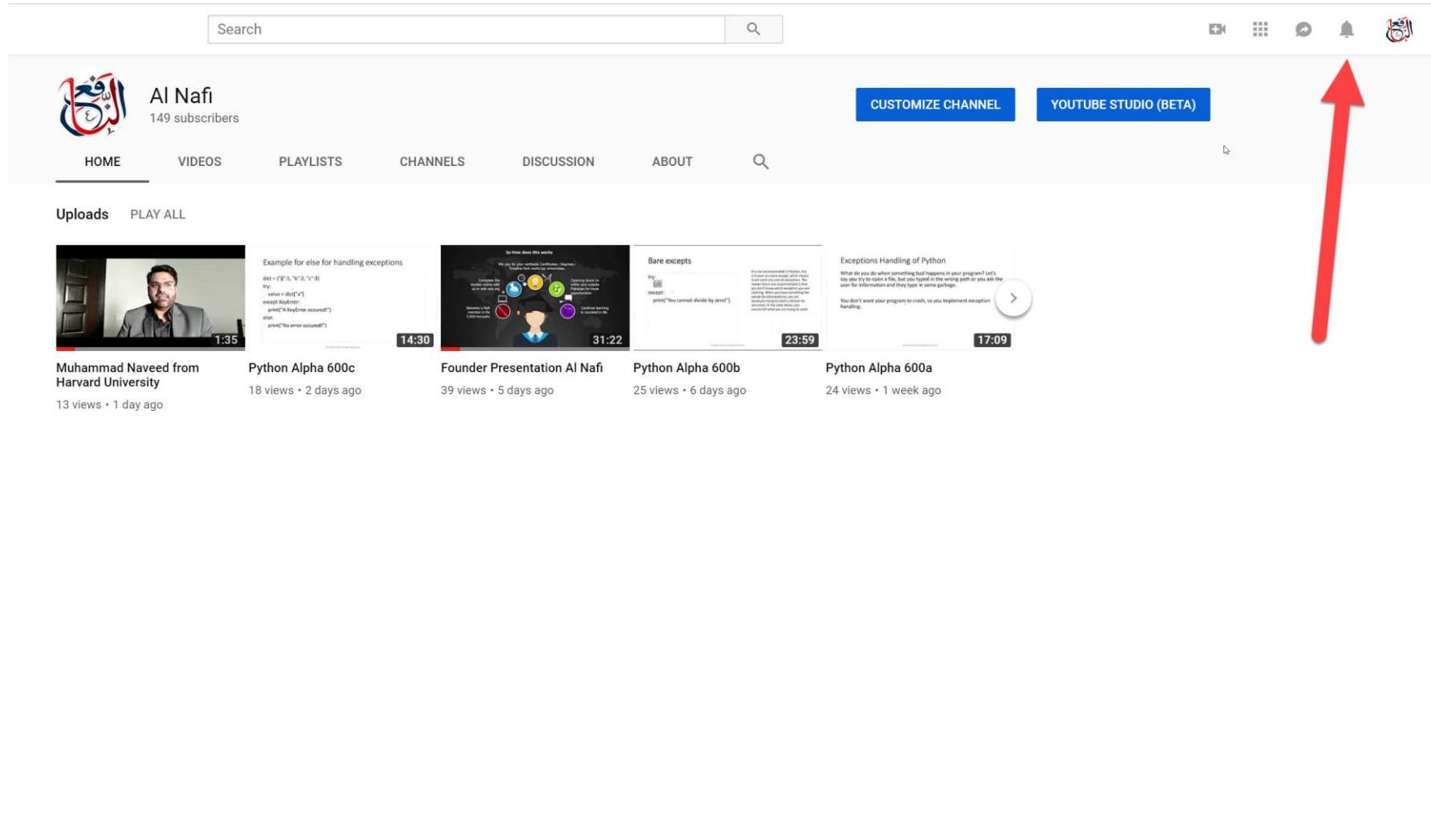


The screenshot shows a web browser window for 'Login - Al-Nafi' at the URL <https://alnafi.com/login/>. The page has a dark blue header with the Al-Nafi logo, navigation links for 'Angraze.com', 'Become a Member?', 'Payment Details', 'Who Are We?', 'FAQ's', and a 'LOG IN' button. Below the header is a teal-colored main menu bar with 'LOGIN' and 'CONTACT US'. The central content area is titled 'Login' and contains fields for 'Username or Email Address' (with 'abuullah' typed) and 'Password' (with several dots). There is also a 'Remember Me' checkbox and a 'LOG IN' button. At the bottom of the login form is a link to 'Lost Password'.



The footer of the website features the Al-Nafi logo and a brief description: 'Al-Nafi company with a focus on education, wellbeing and renewable energy in Pakistan.' To the right, there are four columns: 'POPULAR COURSES' (Cyber Security, IOT, Data Science, Cloud Computing), 'EXPLORE' (Home, Nafi Benefits, Zone, Log In, Contact us), 'GET IN TOUCH WITH US' (info@nafi.com, +1 647-680-1508, +92345-2494147, social media icons for Facebook, YouTube, and Twitter), and a copyright notice: '© 2018 Al-Nafi. All Rights Reserved.'

# Our YouTube channel



# Infinite arguments and keyword arguments using \* and \*\*

We can also create functions to accept any number of arguments or keyword arguments by using a special syntax using \* and \*\*

To get infinite arguments, we use \*args and for infinite we use \*\*kwargs

Please note that the “args” and “kwargs” are not important as That's just convention. We can use any word in their place. The key is using the number of asterisks

Within the programs we will also see \*a and \*\*kw from time to time.

# Examples

```
def many(*args, **kwargs):  
    print(args)  
    print(kwargs)
```

And then run this

```
many(1,2,3, name="Ali" , job="Cyber Warrior")
```

# Scope and Global concepts

In Python we have a concept of **scope** just like other programming languages.

Scope tells us when a variable is available to use and where.

We have to define the variable inside a function, as those variables can only be used inside that function only. Once that function ends, that variable cannot be used again and is known as out of scope.

# Examples of scope and global

```
def function_a():
```

```
    a = 1
```

```
    b = 2
```

```
    return a+b
```

```
def function_b():
```

```
    c = 3
```

```
    return a+c
```

```
print(function_a())
```

```
print(function_b())
```

# Examples continued

```
def function_a():
    global a
    a = 1
    b = 2
    return a+b
```

```
def function_b():
    c=3
    return a+c
```

```
print(function_a())
print(function_b())
```

جزاك الله

To ask questions, please logon to the portal <https://alnafi.com/login/> and use your username and password. We will circle back to you in 2-3 business days inshAllah.



AL NAIFI,  
A company with a focus on education,  
wellbeing and renewable energy.

# Python Alpha 1000a

## Classes in Python

# Forty Hadith on the importance of Knowledge, learning and Teaching.

# Hadith # 7 Education

Ibn'Abbas narrated, the messenger of Allah peace be upon him said:

“Teach! Make things easy! And do not make things complicated!  
He (peace be upon him) repeated this three time, then he added:

And if you get angry, keep quiet!

(Ibn Hajar, al-'Asqalani, al-Matalib al-Aliyah; at-tayalisi; Ibn Hanbal; Ibn Abi Shaibah)

# Hadith # 7 continued

Al-Bukhari narrated it on the authority of Anas, as follows:

“Make things easy! Do not make them complicated! Be Cheerful! And do not be repulsive!”

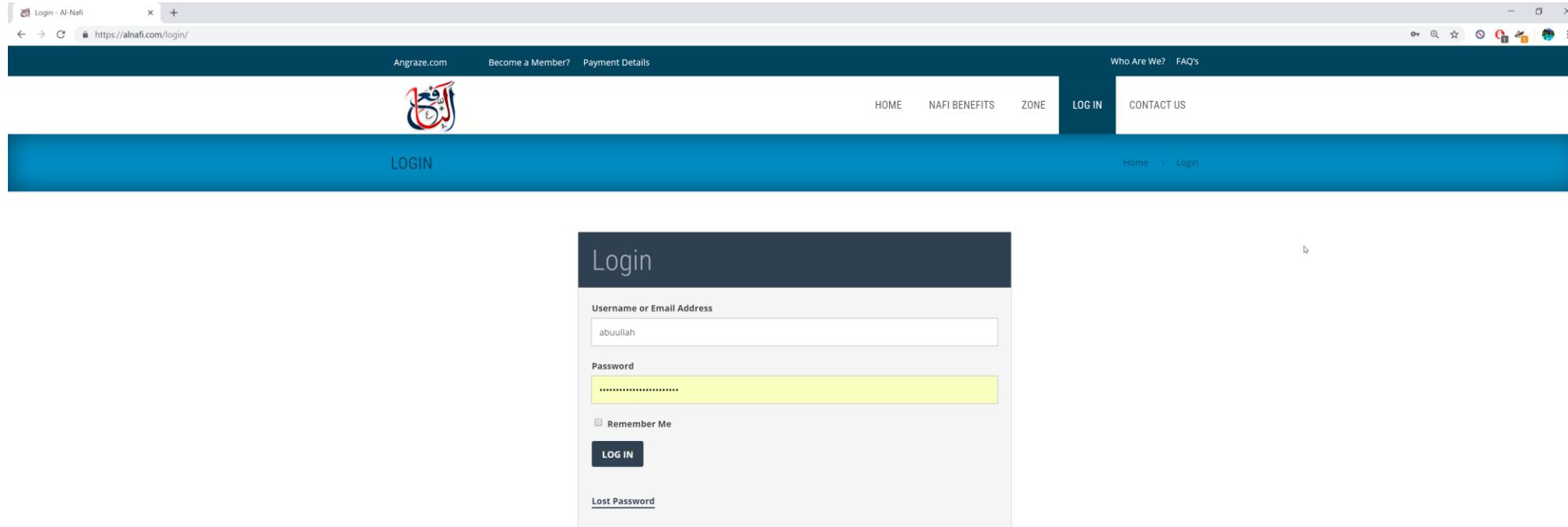
Ahmad Ibn Hanbal narrated it as follows:

“Teach and give good tidings! Make things easy, and do not make them hard!”

# How Nafi Members Study!

1. Please subscribe to our YouTube channel  
<https://www.youtube.com/channel/UC2yAW4Oq27r1yuRE8ePKRvA>
2. Follow us on Facebook <https://www.facebook.com/info.alnafi/>
3. Twitter <https://twitter.com/nafiPakistan>
4. All Nafi members MUST study on the portal <https://alnafi.com/login/> and connect using your Nafi member username and password
5. Once on the portal they can follow their classes by:
  - watching videos
  - asking questions
  - attempting quizzes and flash cards
  - studying official Nafi notes
  - taking their own notes online just like evernote
  - keep track of their studies long with many more features

# Members Please connect to our portal

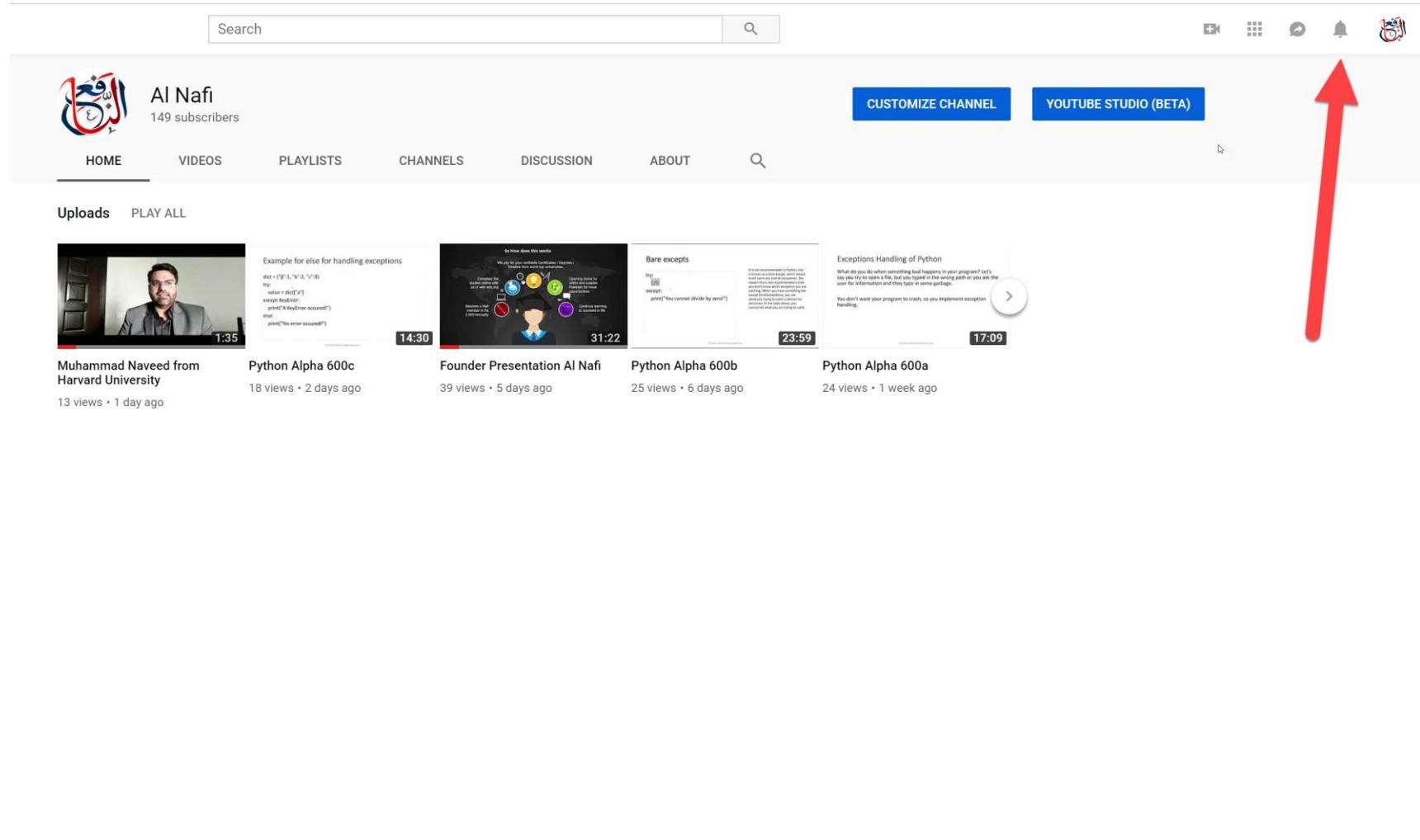


The screenshot shows a web browser window for 'Login - Al-Nafi' at the URL <https://alnafi.com/login/>. The page has a dark blue header with the Al-Nafi logo, navigation links for 'Angraze.com', 'Become a Member?', 'Payment Details', 'Who Are We?', 'FAQ's', and a 'LOG IN' button. Below the header is a teal-colored main menu bar with 'LOGIN' and 'CONTACT US'. The central content area is titled 'Login' and contains fields for 'Username or Email Address' (with 'abuullah' typed in) and 'Password' (with several dots). There is also a 'Remember Me' checkbox and a 'LOG IN' button. At the bottom of the login form is a link for 'Lost Password'.



The footer of the website features a dark blue background. On the left, there is a logo and a brief description: 'Al-Nafi company with a focus on education, wellbeing and renewable energy in Pakistan.' In the center, there are four columns: 'POPULAR COURSES' (Cyber Security, IOT, Data Science, Cloud Computing), 'EXPLORE' (Home, Nafi Benefits, Zone, Log In, Contact us), 'GET IN TOUCH WITH US' (info@nafi.com, +1 647-680-1508, +92345-2494147, social media icons for Facebook, YouTube, and Twitter), and a copyright notice: '© 2018 Al-Nafi. All Rights Reserved.'

# Our YouTube channel



# Classes

In Python everything has method and values. This is known as a class. A class is the blueprint of an object.

Lets try this:

```
x = "alnafi"
```

```
dir(x)
```

# So what does `dir(x)` shows us!

When we have a string assigned to the variable `x`. The `dir` shows us how many methods are there.

There are 71 methods here!

Normally we do not call methods that start with underscores directly so that reduces the total count to 38.

This means that string is based on a class and `x` is an instance of that class!

# Python 2 vs. Python 3

All of you are using Anaconda which is running python 3. but Python 3 is also backward compatible with Python 2 😊 so if you run any code of python 2 which uses a slightly different syntax things will still run 😊

Lets jump into some code testing shall we!

# Creating a class in python 2

Creating a class in Python 2:

```
class Vehicle(object):
    """docstring"""
    def __init__(self):
        """Constructor"""
        pass
```

Code explanation:

**class** is a keyword we use in python 2 followed by the name of the class which in this case **Vehicle**. The first letter is always capital. **object** requires a () parentheses. Classes have a special method **\_\_init\_\_** ( this is required for initialization). The **\_\_init\_\_** is a constructor

# Creating a class in python 3

```
class Vehicle:  
    """docstring"""  
    def __init__(self):  
        """Constructor"""  
        pass
```

The only difference is that in Python 3 we no longer need the parentheses.

# Side by Side code comparasion of Class in python 2 and python 3

Creating a class in Python 2:

```
class Vehicle(object):  
    """docstring"""  
    def __init__(self):  
        """Constructor"""  
        pass
```

Creating a class in Python 3:

```
class Vehicle:  
    """docstring"""  
    def __init__(self):  
        """Constructor"""  
        pass
```

# Now lets run some code!

```
1 class Vehicle(object):
2     """docstring"""
3     def __init__(self, color, doors, tires):
4         """Constructor"""
5         self.color = color
6         self.doors = doors
7         self.tires = tires
8
9     def brake(self):
10        """
11            stop the car
12        """
13        return "Braking"
14    def drive(self):
15        """
16            Drive the car
17        """
18        return "I'm driving!"
```

# What is self in Python?

**We will cover this in our next lecture inshAllah!**

جزاك الله

To ask questions, please logon to the portal <https://alnafi.com/login/> and use your username and password. We will circle back to you in 2-3 business days inshAllah.