



AL NAFI,
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 خَطَايَايَ كَمَا
بَاعَدْتَ بَيْنَ الْمَشْرِقِ وَالْمَغْرِبِ،
اللَّهُمَّ نَقِّنِي مِنْ خَطَايَايَ كَمَا يُنْقَى الثَّوْبُ
الْأَبْيَضُ مِنَ الدَّنَسِ، اللَّهُمَّ اغْسِلْنِي مِنْ
خَطَايَايَ بِالثَّلْجِ وَالْمَاءِ وَالْبَرَدِ

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 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