# PYTHON SYNTAX

## Execute Python Syntax

```
>>> print("Hello, World!")
Hello, World!
```

Or by creating a python file on the server, using the .py file extension, and running it in the Command Line:

```
C:\Users\Your Name>python myfile.py
```

## Python Indentation

Indentation refers to the spaces at the beginning of a code line.

Where in other programming languages the indentation in code is for readability only, the indentation in Python is very important.

Python uses indentation to indicate a block of code.

ExampleGet your own Python Server

```
if 5 > 2:
  print("Five is greater than two!")
```

Python will give you an error if you skip the indentation:

ExampleGet your own Python Server

Syntax Error:

```
if 5 > 2:
print("Five is greater than two!")
```

The number of spaces is up to you as a programmer, the most common use is four, but it has to be at least one.

ExampleGet your own Python Server

```
if 5 > 2:
 print("Five is greater than two!")
if 5 > 2:
        print("Five is greater than two!")
```

You have to use the same number of spaces in the same block of code, otherwise Python will give you an error:

ExampleGet your own Python Server

Syntax Error:

```
if 5 > 2:
 print("Five is greater than two!")
        print("Five is greater than two!")
```

## Python Variables

In Python, variables are created when you assign a value to it:

ExampleGet your own Python Server

Variables in Python:

```
x = 5
y = "Hello, World!"
```

Python has no command for declaring a variable.

You will learn more about variables in the Python Variables chapter.

## Comments

Python has commenting capability for the purpose of in-code documentation.

Comments start with a #, and Python will render the rest of the line as a comment:

ExampleGet your own Python Server

Comments in Python:

```
#This is a comment.
print("Hello, World!")
```

## Python Comments

Comments can be used to explain Python code.

Comments can be used to make the code more readable.

Comments can be used to prevent execution when testing code.

## Creating a Comment

Comments starts with a #, and Python will ignore them:

ExampleGet your own Python Server

```
#This is a comment
print("Hello, World!")
```

Comments can be placed at the end of a line, and Python will ignore the rest of the line:

ExampleGet your own Python Server

```
print("Hello, World!") #This is a comment
```

A comment does not have to be text that explains the code, it can also be used to prevent Python from executing code:

ExampleGet your own Python Server

```
#print("Hello, World!")
print("Cheers, Mate!")
```

## Multiline Comments

Python does not really have a syntax for multiline comments.

To add a multiline comment you could insert a # for each line:

ExampleGet your own Python Server

```
#This is a comment
#written in
```

```
#more than just one line
print("Hello, World!")
```

Or, not quite as intended, you can use a multiline string.

Since Python will ignore string literals that are not assigned to a variable, you can add a multiline string (triple quotes) in your code, and place your comment inside it:

ExampleGet your own Python Server

```
"""
This is a comment
written in
more than just one line
"""
print("Hello, World!")
```

As long as the string is not assigned to a variable, Python will read the code, but then ignore it, and you have made a multiline comment.

KEYWORDS:

```
>>> help("keywords")

Here is a list of the Python keywords.  Enter any keyword to get more help.

False               class               from                or
None                continue            global              pass
True                def                 if                  raise
```

and

as

assert

async

await

break

del

elif

else

except

finally

for

import

in

is

lambda

nonlocal

not

return

try

while

with

yield