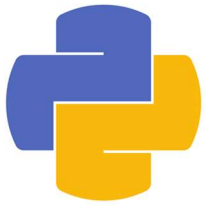


Python Programming

Session 2

Seyyed Ali Shohadaalhosseini

TOPICS

Python Fundamentals and Programming

- **Python Syntax**

Python Syntax

Syntax refers to the structure of the language.

- Comments
- DocString
- End-of-Line
- Semicolon
- Indentation
- Whitespaces
- Parentheses

Codes > Average.py > avg

```python
# @AliShhde  - - - - - - - - -  Ali Shohadaee

def avg():
    """ This function will calculate
    the average of two number """

    # Getting the first number
    num1 = int(input("Enter a number: "))

    # Getting the second number
    num2 = int(input("Enter an other number: "))

    # Calculating its sum
    sum = num1 + num2

    # Calculating its divide
    sum /= 2

    # Showing the answere in output
    print(sum)

print("Hello we can calculate the avarage of the two number",
    " So in the Following you'll be notify to enter numbers.")

avg()
```

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.
- Comments starts with a #
- Python ignores all the text **in front of a comment**

Multi-Line Comments

Python does not really have a syntax for multi line comments.

To add a multi line comment you could insert a # for each line

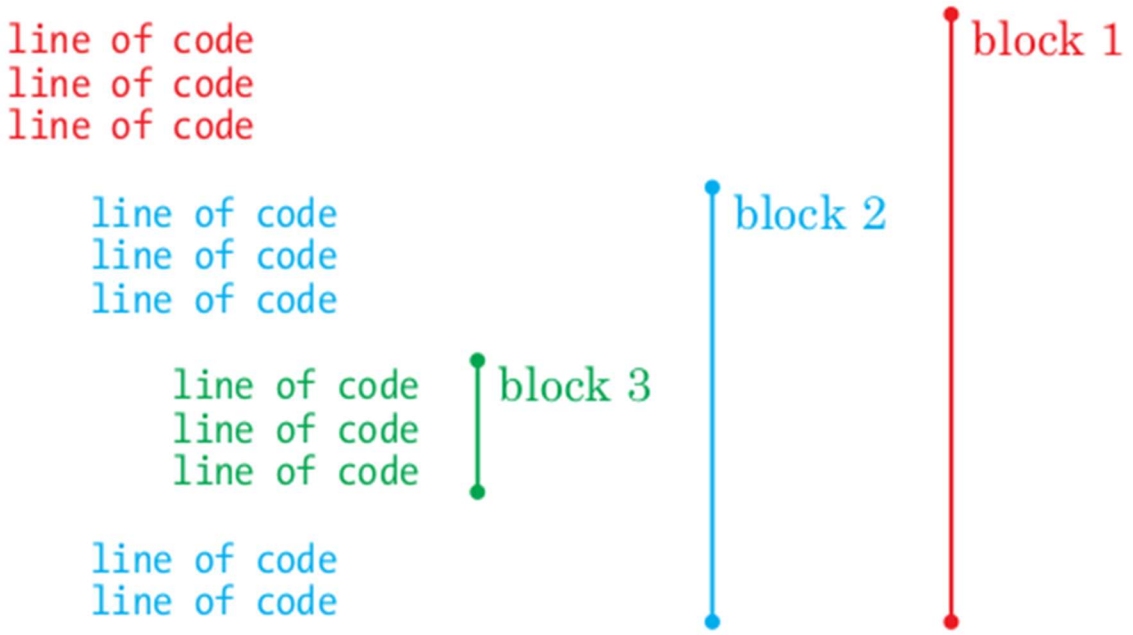
BUT

Since python will ignore String literals that are not assigned to a variable, you can add a multiline string in your code as a comment.

A multiline String “”” Your comments here “””

Indentation in python

- We separate code with indentation in python



Errors or Friends

Remember: **Errors are our Good Friend**

We have three kind of Errors

- Syntax errors
- Runtime errors
- Logic errors

Syntax Errors

```
  File "c:\Users\Alishhde\Desktop\Python Powerpoint\Session 2\Codes\Average.py", line 22
    prin t("Hello we can calculate the avarage of the two number",
         ^
SyntaxError: invalid syntax
```

Easiest to find and correct

The interpreter will tell you where it got into trouble.

Usually the error is on the exact line indicated by the interpret or the Line just before it.

Runtime Errors

```
Traceback (most recent call last):
  File "c:\Users\Alishhde\Desktop\Python Powerpoint\Session 2\Codes\Average.py", line 6, in <module>
    print(num2/num1)
ZeroDivisionError: division by zero
```

Runtime errors are intermediate in difficulty.

Python will tell you where is the problem, but you must check it how it has happened.

In this kind of error we don't see any syntax error or syntax problem, the program runs and then you see there is somewhere that an operation can not be executed.

Like: **zero Division error**

Logical Errors

A logic error is when your program runs and compile successfully and shows you the output BUT it shows an output that you did not expect.

In this kind of program, the programs successfully runs but does a job that you did not program it to do, I mean it does wrong job.

Input/Output in python

If you want to show something in output, actually in terminal, you can use *print()* method to do that.

And also if you want to get something, some value from outside of the program you can use *input()* method to do that.