Agenda of the day

Conditional statement

1. if <expression-topevaluate> o/p --->True or False

# Action 1

# Action 2

2. if <expression-topevaluate> o/p --->True or False

# Action 1

else:

# Action 1

3. if elif and else

if condition\_1:

statement\_block\_1

elif condition\_2:

statement\_block\_2

elif another\_condition:

another\_statement\_block

else:

else\_block

eg:

""""

If it rains tomorrow,

I'll clean up the basement.

After that I will paint the walls.

If there is any time left,

I will file my tax return.

Otherwise, if it will not rain,

I will go swimming.

In the evening, I will go to the cinema with my wife!

"""

"""

if rains:

# I'll clean up the basement

# I will paint the walls

if anytime left:

# I will file my tax return

else:

# I will go swimming

I will go to the cinema with my wife!

"""

Lines and Indentation

Python uses a different principle. Python programs get structured through indentation, i.e. code blocks are defined by their indentation.

Okay thats what we expect from any program code, is not it? Yes, but in the case of Python it is a language requirement, not a matter of style.

This principle makes it easier to read and understand other people s Python code.

So, how does it work? All statements with the same distance to the right belong to the same block of code, i.e. the statements within a block line up vertically.

The block ends at a line less indented or the end of the file. If a block has to be more deeply nested, it is simply indented further to the right.

Beginners are not supposed to understand the following example, because we have not introduced most of the used structures, like conditional statements and loops.

Precedence of Operator

Precedence Operator Description

1 \*\* exponent

2 \*, /, %, // multiply, divide, modulo, floor division

3 +, - plus, minus

4 >, <, >=, <= comparison

5 ==, != equality

6 = %= /= //= -= += = \*= assignment

7 and, or, not logical

Associativity

if the Precedence of both the Operator are same then we do execution from left to right.

Only in case of exponent the execution will happend from Right to left.

eg:

a = 3

b = 2

c = 3

print(a \* (b \* c))

# Ans : 6561 --> This is the correct ans

# Ans: 729, combination 1 : (a \* b) \* c ---> left to right

# Ans: 6561, combination 2 : a \* (b \* c) --> Right to left