எங்கள் வாழ்வும் எங்கள் வளமும்
மங்காத தமிழ் என்று சங்கே முழங்கு ... *புரட்சிக்கவி*

NOTICE

- We support open-source products to spread Technology to the mass.
- This is completely a FREE training course to provide introduction to Python language
- All materials/contents/images/examples and logo used in this document are owned by the respective companies/websites. We use those contents for FREE teaching purposes only.
- We take utmost care to provide credits when ever we use materials from external source/s. If we missed to acknowledge any content that we had used here, please feel free to inform us at info@DataScienceInTamil.com.
- All the programing examples in this document are for teaching purposes only.

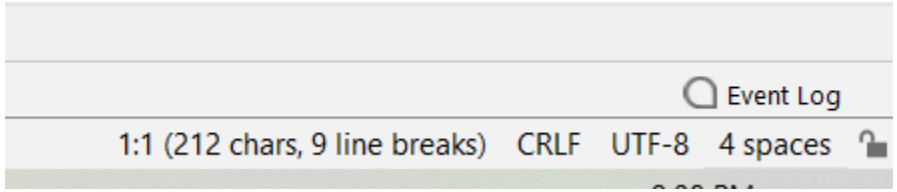
Thanks to all the open source community and to the below websites from where we take references / content /code example. definitions, please use these websites for further reading:

https://www.geeksforgeeks.org/indentation-in-python/
https://docs.python.org/2.0/ref/indentation.html
https://www.askpython.com/python/python-indentation
https://docs.python.org/3/reference/lexical_analysis.html

What to cover today?

1. THE RIGHT INDENTATION
2. THE WRONG INDENTATION
3. PYTHON INDENTATION RULES
4. EXAMPLES
5. INDENTATION ERROR
6. LEXICAL ANALYZER
7. INDENT

8. DEDENT



Indentation is a very important concept of Python because without proper indenting the Python code, you will end up seeing `IndentationError` and the code will not get compiled.

Python indentation is **a way of telling a Python interpreter that the group of statements belongs to a particular block of code.** A block is a combination of all these statements. Block can be regarded as the grouping of statements for a specific purpose

THE RIGHT INDENTATION

```
def addFunction():
    print("Hi")

    if True:
        print("true")
    else:
        print("false")


print("Task is Done")
---------------
```

THE WRONG INDENTATION

```
if True:
    print("true")
     print("Hi")
else:
    print("false")
-------------
```

PYTHON INDENTATION RULES

1. We can’t split indentation into multiple lines using backslash.

2. The first line of Python code can't have indentation, it will throw IndentationError.
3. You should avoid mixing tabs and whitespaces to create indentation. It's because text editors in Non-Unix systems behave differently and mixing them can cause wrong indentation.
4. It is preferred to use whitespaces for indentation than the tab character.
5. The best practice is to use 4 whitespaces for first indentation and then keep adding additional 4 whitespaces to increase the indentation.

EXAMPLES

```
bankName = "SBI"
if bankName == "SBI":
    print("Yes, you have selected the correct bank")
 else: # before else there is space
    print("Sorry, you to select the correct bank to proceed")
```

output

```
  File
"C:\Users\Melcose\PycharmProjects\PythonDSIT\DataScienceInTamil.py", line
4
```

```
    else:
        ^
```

IndentationError: unindent does not match any outer indentation level

---

```
def testIndentation():
    a = 10
    if a == 10:
        print("Yes it is ")
    else:
        print("No it is not")

testIndentation()
```

output

```
def testIndentation():
    a = 10
    if a == 10:
        print("Yes it is ")
     else: # if and else is not in the same  indent
         print("No it is not")

testIndentation()
```

output

```
    else: # if and else is not in the same  indent
                                          ^
IndentationError: unindent does not match any outer indentation level
--------------
```

For Python, Guido van Rossum based the grouping of statements on indentation. The reasons for this are explained in the first section of the "Design and History Python FAQ". (https://docs.python.org/3/faq/design.html) Colons, :, are used to declare an indented code block(https://docs.python.org/3/faq/design.html#why-are-colons-required-for-the-if-while-def-class-statements), such as the following example:

The recommended (https://www.python.org/dev/peps/pep-0008/#tabs-or-spaces)indentation is 4 spaces but tabs or spaces can be used so long as they are consistent. Do not mix tabs and spaces in Python as this will cause an error in Python 3 and can causes errors in Python 2.

Whitespace is handled by the lexical analyzer before being parsed.

The lexical analyzer uses a stack to store indentation levels. At the beginning, the stack contains just the value 0,which is the leftmost position. Whenever a nested block begins, the new indentation level is pushed on the stack,and an "INDENT" token is inserted into the token stream which is passed to the parser. There can never be more than one "INDENT" token in a row (IndentationError).

When a line is encountered with a smaller indentation level, values are popped from the stack until a value is on top which is equal to the new indentation level (if none is found, a syntax error occurs). For each value popped, a "DEDENT" token is generated. Obviously, there can be multiple "DEDENT" tokens in a row.

The lexical analyzer skips empty lines (those containing only whitespace and possibly comments), and will never generate either "INDENT" or "DEDENT" tokens for them.

At the end of the source code, "DEDENT" tokens are generated for each indentation level left on the stack, until just the 0 is left.

The parser handles the "INDENT" and "DEDENT" tokens as block delimiters.

========