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.

What to cover today

1. COMMENTS AND DOCUMENTATION
2. INLINE COMMENT
3. WRITE DOCUMENTATION USING DOCSTRINGS
4. ONE-LINE DOCSTRINGS
5. MULTI-LINE DOCSTRING
6. PEP 257 DEFINITION FOR COMMENTS AND DOCUMENTATION

COMMENTS AND DOCUMENTATION

- Single line, inline and multiline comments
- Comments are used to explain code when the basic code itself isn't clear.

- Python ignores comments, and so will not execute code in there, or raise syntax errors for plain English sentences.
- Single-line comments begin with the hash character (#) and are terminated by the end of line.

Single line comment:

This is a single line comment in Python

========

Inline comment:

```
print("Hello World") # This line prints "Hello World"
```

Comments spanning multiple lines have """ or ''' on either end. This is the same as a multiline string, but

they can be used as comments:

```
print("Hello World") # This line prints "Hello World"

"""
This type of comment spans multiple lines.
These are mostly used for documentation of functions,
```

```
classes and modules.
"""
============
```

Write documentation using docstrings

A docstring is a multi-line comment used to document modules, classes, functions and methods. It has to be the first statement of the component it describes.

Below docstring is given to a method

```
def hello(name):
    """Greet someone.
 Print a greeting ("Hello") for the person with the given name.
"""
    print("Hello "+name)
```

Below docstring is given to a class

```
class Greeter:
    """An object used to greet people.
```

```
It contains multiple greeting functions for several languages
 and times of the day.
 """

---------
```

The value of the docstring can be accessed within the program and is - for example - used by the help command.

Notes:

1. Docstring must be given at the start of the function or class
2. Else it treats as multiline comment
3. First line indent must be 4 space, other lines does not care about indentation

See the code below for doctstring and for mulitline comment

```
def fn1():
    """
    this is part of  doctstring
    multiline
    format
    """
    print(10+20)
    '''''
```

```
    This not part
     of the docstring
    """
    print(100 + 200)

fn1()
print(fn1.__doc__)
```

output

30

300

```
    this is part of  doctstring
    multiline
    format
        ============
```

Syntax conventions PEP 257

PEP 257 defines a syntax standard for docstring comments.

It basically allows two types: One-line Docstrings: According to PEP 257, they should be used with short and simple functions.
Everything is placed in one line, e.g:

ONE-LINE DOCSTRINGS:

```
def hello():
    """Say hello to your friends."""
    print("hello my friends")

print(hello.__doc__)
======
```

The docstring shall end **with a period,** the verb should be in the imperative form.

MULTI-LINE DOCSTRING

should be used for longer, more complex functions, modules or classes.

```
def hello(name, language="en"):
    """Say hello to a person.
 Arguments:
 name: the name of the person
 language: the language in which the person should be greeted
```

```
    """
    print(name, language)
print(hello("Dhanu", "Tamil"))
```

They start with a short summary (equivalent to the content of a one-line docstring) which can be on the same line as the quotation marks or on the next line, give additional detail and list parameters and return values.

Note PEP 257 defines what information should be given (https://www.python.org/dev/peps/pep-0257/#multi-line-docstrings)within a docstring, it doesn't define in which format it should be given. This was the reason for other parties and documentation parsing tools to specify their own standards for documentation, some of which are listed below and in this question (https://stackoverflow.com/questions/5334531/using-javadoc-for-python-documentation)