Comments and Documentation

All Versions

Examples

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:

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

Programmatically Accessing Docstrings

Part of the advantage a docstring provides over regular comments is that they are stored as an attribute of the function, meaning that you can access them programmatically.

```
def func():
    """This is a function that does nothing at all"""
    return

print(func.__doc__)
# This is a function that does nothing at all

help(func)
# Help on function func in module __main__:
#
# func()
#     This is a function that does nothing at all
```

`function.__doc__` is just the actual docstring as a string, while the `help` function provides general information about a function, including the docstring. Here's a more helpful example:

```
def greet(name, greeting="Hello"):
    """Print a greeting to the user `name`

    Optional parameter `greeting` can change what they're greeted with."""

    print("{} {}".format(greeting, name))

help(greet)
# Help on function greet in module __main__:
#
# greet(name, greeting='Hello')
#     Print a greeting to the user `name`
#
#     Optional parameter `greeting` can change what they're greeted with.
```

Just putting no docstring or a regular comment in a function makes it a lot less helpful.

```
def greet(name, greeting="Hello"):
    # Print a greeting to the user `name`
    # Optional parameter `greeting` can change what they're greeted with.

    print("{} {}".format(greeting, name))

print(greet.__doc__)
# None
```

```
help(greet)
# Help on function greet in module __main__:
#
# greet(name, greeting='Hello')
#
```

Syntax

```
#this is a single line comment
```

```
print("") #this is an inline comment
```

```
"""
this is
a multi-line comment
"""
```

Parameters

Remarks

Developers should follow the PEP257 - Docstring Conventions guidelines. In some cases, style guides (such as Google Style Guide ones) or documentation rendering third-parties (such as Sphinx) may detail additional conventions for docstrings.