



DOCSTRINGS IN PYTHON

Understanding the Importance of Documentation

MUKESH KUMAR

Agenda

- 1. What are Docstrings?
- 2. Syntax of a Docstring
- 3. Importance of Docstrings
- 4. Writing Good Docstrings
- 5. Docstring Conventions (PEP 257)
- 6. Accessing Docstrings
- 7. Example of a Well-Documented Function
- 8. Best Practices
- 9. Conclusion
- 10. Q&A

What are Docstrings?

- - A docstring is a string literal that appears right after the definition of a function, method, class, or module.
- - It is used to document the purpose and functionality of the code.
- - The docstring is accessible via the `__doc__` attribute.

Syntax of a Docstring

- - Docstrings are written in between triple quotes ('''...' or '"""..."""').
- - Example:

```
def add(a, b):  
    """  
    Adds two numbers.  
  
    Parameters:  
    a (int or float): The first number.  
    b (int or float): The second number.  
  
    Returns:  
    int or float: The sum of the two numbers.  
    """  
    return a + b
```

Importance of Docstrings

- - Improves code readability.
- - Provides helpful information about function parameters, return values, and usage.
- - Ensures easier maintenance and collaboration in teams.

Writing Good Docstrings

- - Be concise: Briefly explain the purpose of the function or class.
- - Include parameters and return types: Describe the expected types and behavior.
- - Follow PEP 257 conventions: Standardize docstring formats for consistency.

Docstring Conventions (PEP 257)

One-liner Docstrings: When the docstring fits on one line, it should be a brief summary.

```
def square(x):  
    """Return the square of x."""  
    return x ** 2
```

Docstring Conventions (PEP 257)

- Multi-line Docstrings: Should start with a short summary and include detailed descriptions on new lines.

```
def divide(a, b):  
    """  
    Divide a by b.  
  
    Parameters:  
    a (float): Dividend  
    b (float): Divisor  
  
    Returns:  
    float: The result of division.  
    """  
    return a / b
```


Accessing Docstrings

- - You can access the docstring using the `help()` function or the `__doc__` attribute:
 - `help(add)` # or
 - `print(add.__doc__)`

Example of a Well-Documented Function

```
def multiply(x, y):  
    """  
    Multiply two numbers.  
  
    Parameters:  
    x (int or float): The first number to multiply.  
    y (int or float): The second number to multiply.  
  
    Returns:  
    int or float: The product of x and y.  
    """  
    return x * y
```

Best Practices

- - Always document your code, especially when working in teams.
- - Keep docstrings up to date with code changes.
- - Be clear about the function's purpose, input, and output.

Conclusion

- - Docstrings are a simple but powerful way to make your code more readable and maintainable.
- - They are an essential tool in professional coding practices.