

Best Practices for Writing Functions: Takeaways



by Dataquest Labs, Inc. - All rights reserved © 2019

Syntax

DOCSTRINGS

- Get a function's raw docstring (includes leading spaces):

```
print(function_name.__doc__)
```

- Retrieve a better formatted version of the docstring (without leading spaces):

```
import inspect  
print(inspect.getdoc(function_name))
```

- Write a Google Style docstring:

```
def function_name(arguments):  
    """  
    Description of what the function does.  
    Description of the arguments, if any.  
    Description of the return value(s), if any.  
    Description of errors raised, if any.  
    Optional extra notes or examples of usage.  
    """
```

Concepts

- A **docstring** is a string written as the first line of a function that describes what the functions does. Docstrings contain some (although usually not all) of these five key pieces of information:
 - What the function does
 - What the arguments are
 - What the return value or values should be
 - Info about any errors raised
 - Anything else you'd like to say about the function

- To access a built-in function's docstring in Jupyter notebook, press "Shift" + "Tab" while the cursor is within the parentheses of the function.
- The "Don't repeat yourself" principle, also known as DRY, states that it's better to wrap repeated logic in a function. The "Do One Thing" principle states that each function should only have a single responsibility. Following these best practices will makes your code more flexible, simpler to test, simpler to debug, and easier to change.
- **Mutable** variables can be changed, whereas **immutable** variables cannot be changed. There are only a few immutable data types in Python because almost everything is represented as an object.
- Instead of using a mutable variable as a default value in a function, default to None and set the argument in the function, so that your function doesn't behave unexpectedly.

Resources

- [Google style docstring guide](#)
- [Numpydoc docstring guide](#)



Takeaways by Dataquest Labs, Inc. - All rights reserved © 2019