

## Docstring

The docstring can be placed anywhere in the definition of the object, but it is typically placed at the beginning of the definition.

The docstring can contain any text, but it is common to include the following information:

The purpose of the object  
The arguments that the object takes  
The return value of the object  
Any other relevant information

```
In [ ]:  ▶ 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.  
        """
```

**A context manager is an object that controls the flow of execution within a block of code. It is typically used to manage resources, such as files or sockets.**

```
In [ ]:  ▶ from contextlib import contextmanager  
  
    @contextmanager  
    def open_file(filename):  
        """Opens a file and closes it when the context is exited."""  
        file = open(filename, 'r')  
        try:  
            yield file  
        finally:  
            file.close()
```

## Decorators

A decorator is a function that takes another function as its argument and returns a new function. The new function is called the decorated function.

Decorators are often used to add functionality to functions, such as logging, timing, or error handling.

```
In [ ]: ▶ def decorator(function):  
        def wrapper(*args, **kwargs):  
            # Do something before the function is called.  
            result = function(*args, **kwargs)  
            # Do something after the function is called.  
            return result  
  
        return wrapper
```