**Commenting object-oriented code**

Did you notice anything special about the answer key in the previous exercise? The Pants class and the SalesPerson class contained docstrings! A docstring is a type of comment that describes how a Python module, function, class, or method works. Docstrings are not unique to object-oriented programming.

For this section of the course, you just need to remember to use docstrings and to comment your code. It will help you understand and maintain your code and even make you a better job candidate.

From this point on, please always comment your code. Use both inline comments and document-level comments as appropriate.

To learn more about docstrings, see [Example Google Style Python Docstrings](https://sphinxcontrib-napoleon.readthedocs.io/en/latest/example_google.html).

**Docstrings and object-oriented code**

The following example shows a class with docstrings. Here are a few things to keep in mind:

* Make sure to indent your docstrings correctly or the code will not run. A docstring should be indented one indentation underneath the class or method being described.
* You don't have to define self in your method docstrings. It's understood that any method will have self as the first method input.

class Pants:

"""The Pants class represents an article of clothing sold in a store

"""

def \_\_init\_\_(self, color, waist\_size, length, price):

"""Method for initializing a Pants object

Args:

color (str)

waist\_size (int)

length (int)

price (float)

Attributes:

color (str): color of a pants object

waist\_size (str): waist size of a pants object

length (str): length of a pants object

price (float): price of a pants object

"""

self.color = color

self.waist\_size = waist\_size

self.length = length

self.price = price

def change\_price(self, new\_price):

"""The change\_price method changes the price attribute of a pants object

Args:

new\_price (float): the new price of the pants object

Returns: None

"""

self.price = new\_price

def discount(self, percentage):

"""The discount method outputs a discounted price of a pants object

Args:

percentage (float): a decimal representing the amount to discount

Returns:

float: the discounted price

"""

return self.price \* (1 - percentage)