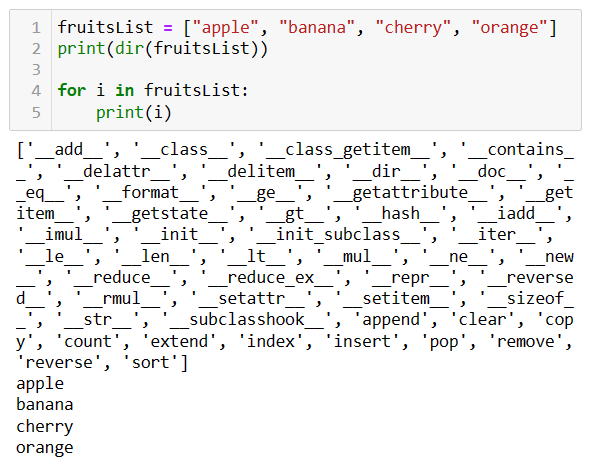
***Iterable and Iterator***

***Iterable:***

*An iterable is any object that can be looped over or iterated upon. Examples of iterables include* ***lists, tuples, strings, dictionaries, sets, and more.***

*Iterables must implement the* ***\_\_iter\_\_()*** *method, which returns an iterator.*

**

***Python Iterators***

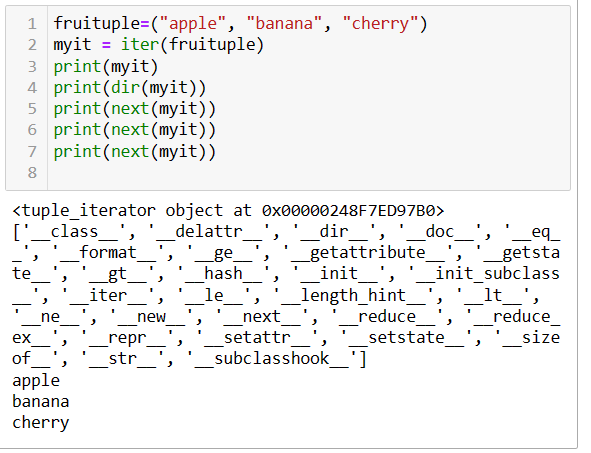
*In Python, an iterator is an object that implements the iterator protocol, which consists of the methods \_****\_iter\_\_()****and****\_\_next\_\_()*** *.*

*An example result of map and filter is iterators since both contain* ***\_\_iter\_\_()****and****\_\_next\_\_()*** *.*

**

***Create an iterator from iterable***

**

**

***Iterator vs Iterable:***

*Lists, tuples, dictionaries, and sets are all iterable objects. They are iterable containers which you can get an iterator from. All these objects have an iter() method which is used to get an iterator:*

|  |  |
| --- | --- |
| ***Iterable*** | ***iterator*** |
| *An iterable is any object that can be looped over or iterated upon.* | *An iterator is an object that represents a stream of data.* |
| *Iterables must implement the* ***\_\_iter\_\_()*** *method.* | *Iterators must implement the* ***\_\_iter\_\_()*** *and* ***\_\_next\_\_()*** *methods.* |
| *Iterable objects can be passed to functions or constructs like* ***for*** *loops directly.* | *The* ***\_\_iter\_\_()*** *method returns the iterator object itself.  The* ***\_\_next\_\_()*** *method returns the next item in the stream. If there are no more items, it raises a StopIteration exception.* |
| *Examples of iterables include lists, tuples, strings, dictionaries, sets, and more.* | *An example is “result of map and filter” is iterators since both contain* ***\_\_iter\_\_()*** *and* ***\_\_next\_\_()*** *.* |

***enumerates():***

*In Python,* ***the enumerate()*** *function is used to iterate over a list (or any other iterable), providing both the* ***index*** *and the* ***value*** *of each element during iteration. This is particularly useful when you need to keep track of the index while iterating over the elements of a list.*

**