Map, Filter and Reduce

In Python, the map, filter, and reduce functions are built-in functions that allow you to apply a function to a sequence of elements and return a new sequence. These functions are known as higher-order functions, as they take other functions as arguments.

map

The map function applies a function to each element in a sequence and returns a new sequence containing the transformed elements. The map function has the following syntax:

```
map(function, iterable)
```

The function argument is a function that is applied to each element in the iterable argument. The iterable argument can be a list, tuple, or any other iterable object.

Here is an example of how to use the map function:

```
# List of numbers
numbers = [1, 2, 3, 4, 5]

# Double each number using the map function
doubled = map(lambda x: x * 2, numbers)

# Print the doubled numbers
print(list(doubled))
```

In the above example, the lambda function lambda x: x * 2 is used to double each element in the numbers list. The map function applies the lambda function to each element in the list and returns a new list containing the doubled numbers.

filter

The filter function filters a sequence of elements based on a given predicate (a function that returns a boolean value) and returns a new sequence containing only the elements that meet the predicate. The filter function has the following syntax:

```
filter(predicate, iterable)
```

The predicate argument is a function that returns a boolean value and is applied to each element in the iterable argument. The iterable argument can be a list, tuple, or any other iterable object.

Here is an example of how to use the filter function:

```
# List of numbers
numbers = [1, 2, 3, 4, 5]
```

```
# Get only the even numbers using the filter function
evens = filter(lambda x: x % 2 == 0, numbers)

# Print the even numbers
print(list(evens))
```
In the above example, the lambda function lambda x: x % 2 == 0 is used to filter the numbers list and return only the even numbers. The filter function applies the lambda function to each element in the list and returns a new list containing only the even numbers.

reduce

The reduce function is a higher-order function that applies a function to a sequence and returns a single value. It is a part of the functools module in Python and has the following syntax:

```
reduce(function, iterable)
```
The function argument is a function that takes in two arguments and returns a single value. The iterable argument is a sequence of elements, such as a list or tuple.

The reduce function applies the function to the first two elements in the iterable and then applies the function to the result and the next element, and so on. The reduce function returns the final result.

Here is an example of how to use the reduce function:

```
from functools import reduce

# List of numbers
numbers = [1, 2, 3, 4, 5]

# Calculate the sum of the numbers using the reduce function
sum = reduce(lambda x, y: x + y, numbers)

# Print the sum
print(sum)
```
In the above example, the reduce function applies the lambda function lambda x, y: x + y to the elements in the numbers list. The lambda function adds the two arguments x and y and returns the result. The reduce function applies the lambda function to the first two elements in the list (1 and 2), then applies the function to the result (3) and the next element (3), and so on. The final result is the sum of all the elements in the list, which is 15.

It is important to note that the reduce function requires the functools module to be imported in order to use it.