



## 23.1. Introduction: Map, Filter, List Comprehensions, and Zip

Let's revisit the [accumulator pattern](#). We have frequently taken a list and produced another list from it that contains either a subset of the items or a transformed version of each item. When each item is transformed we say that the operation is a **mapping, or just a map** of the original list. When some items are omitted, we call it a **filter**.

Python provides built-in functions `map` and `filter`. Python also provides a new syntax, called **list comprehensions**, that lets you express a mapping and/or filtering operation. Just as with named functions and lambda expressions, some students seem to find it easier to think in terms of the map and filter functions, while other students find it easier to read and write list comprehensions. You'll learn both ways; one may even help you understand the other. Most python programmers use list comprehensions, so make sure you learn to read those. In this course, you can choose to learn to write list comprehensions or to use map and filter, whichever you prefer. You should learn to read both list comprehensions and map/filter.

Other common accumulator patterns on lists aggregate all the values into a single value.

Map, and filter are commands that you would use in high-performance computing on big datasets. See [MapReduce on Wikipedia](#).

You have attempted 1 of 1 activities on this page

✓ Completed. Well Done!

[23. More on Accumulation: Map, Filter, List Comprehension, and Zip](#)>

[23. More on Accumulation: Map, Filter, List Comprehension, and Zip](#)>

[23.2. Map](#)>



[23.2. Map](#)>Next Section - 23.2. Map