

# Functions That Work with Iterators

- Convenience functions
- Numerical and logical functions
- Itertools
- Functools
- Example

# Using Convenience Functions

- Several built-in functions for working with iterators
- `zip()`, `map()`, `enumerate()`, and `filter()`
- How these functions compare to list comprehensions and generator expressions

# Using Numerical and Logical Functions

- Several built-in functions for working with iterators
- Logical functions – `any()`, `all()`
- Numerical functions – `min()`, `max()`, `sum()`
- How these functions compare to generator expressions

# The Itertools Module

- The itertools module which provides –
  - A wide variety of functions to create (or operate on) iterators
  - To select and group elements from iterators
  - To implement combinatorial logic



# The Functools Module

- The concept of higher-order functions
- The functools module

When a function takes  
functions as arguments

Or returns functions a return  
values

It is called a higher-order  
function

This is used often in  
functional programming