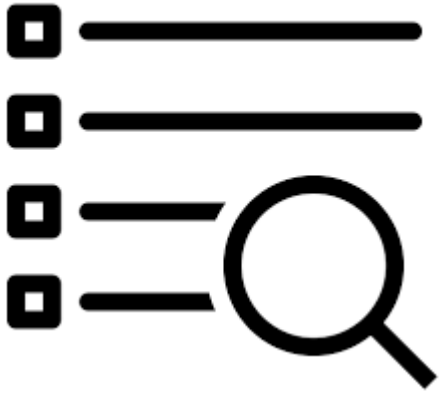
Chapter 7
Using Lists

Outline

- Processing Lists
- Lists and Functions
- Lists and Files
- List Comprehension and Tuples
- Two-Dimensional Lists
- Plotting Lists with **`matplotlib`**



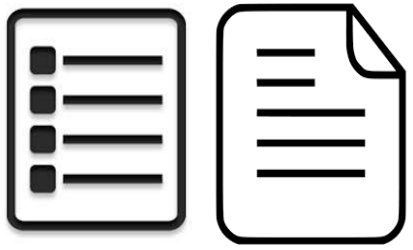
Processing Lists

- Numerical lists
  - Ex: List of daily sales
    - Loop through list
    - Accumulate daily sales into total sales
  - Ex: List of mortgage payments
    - Loop through list, accumulate total
    - Divide total by list length to get average
- String lists
  - Ex. List of customer cities
    - Loop through list
    - Count number of customers is a particular city
- **Lect7_Loans_Proc.py**

Lists and Functions

- Passing/returning lists to/from functions
  - A list goes into the function -> a single value comes out
    - Many in -> One out
      - List of payments sent to the function; the average payment returned
  - A single value goes in -> a list of values is returned
    - One in -> Many out
      - Payment threshold goes in; a list of customers over the threshold comes out
  - Nothing goes in -> a list of values is returned
    - Nothing in -> Many out
      - List gathering input function
  - A list goes in -> a list of values is returned
    - Many in -> Many out
      - List of customer cities goes in; a list of city counts comes out

Lists and Files

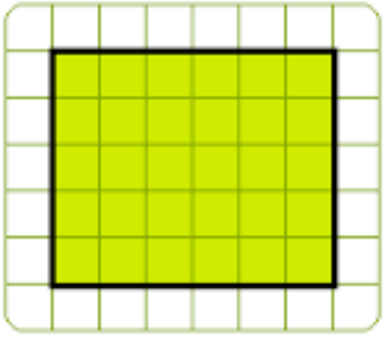


- Writing from a list into a file
  - Loop through list and write each element to a new line
    - Strings written as strings plus the newline character
      - Writing customer cities to **Cities.txt** file
    - Numbers (`int`, `float`) written as strings plus the newline character
      - Writing mortgage payments to **Payments.txt** file
- Reading from a file into a list
  - Use `readlines()` file method to read all the lines into a list
  - Loop through the list to process each element
    - Strings need to be stripped of the newline character
      - Reading customer cities from **Cities.txt** file
    - Numbers need to be converted from strings to `int` or `float`
      - Reading mortgage payments from **Payments.txt** file

List Comprehension and Tuples

- List comprehension
  - An efficient way of creating one list from another
    - Create list of city lengths from the list of cities
  - Can include an if clause that acts like a filter
    - Create a list of payments under $1,500
- Tuples
  - Are immutable lists whose content cannot be modified
  - Support all the same methods and functions except those that would change their contents
  - Faster to process than lists, safe because they can't be modified
  - Can be converted from a list with the **`tuple()`** function
    - Convert payments list into a tuple and try modifying an element

Two-Dimensional Lists



- A list where each element is another list
  - Assume each list element is of the same size
  - The resulting data structure can be thought of as a table with rows and columns
  - There are more efficient Python objects used to store and process multi-dimensional data such as **`DataFrame`** from **`pandas`** and **`array`** from **`NumPy`** packages
- Examples
  - Listing customer cities and counts of customers in those cities

    `[['Albuquerque', 5], ['Santa Fe', 6], ['Taos', 5]]`
  - Processing and summarizing loans

Plotting Lists with `matplotlib`

- Install matplotlib library package
  ```
  > pip install matplotlib
  ```
- Create alias to pyplot module with basic charts
  ```
  >>> import matplotlib.pyplot as plt
  ```
- Basic pyplot functions (***italics*** are optional parameters) are:
  - Line chart: `plt.plot(x_coords, y_coords, marker)`
  - Bar chart: `plt.bar(pos, heights, bar_width)`
  - Pie chart: `plt.pie(values, labels)`
- Use the `plt.show()` method to display the chart
- Some of the common attribute functions include:
  - `plt.title(),plt.xlabel(),plt.ylabel()`
  - `plt.grid(),plt.xticks(),plt.yticks()`
- Several attribute functions unique to a particular chart type
  - Ex. Line chart: `plt.xlim(xmin, xmax),plt.ylim(ymin, ymax)`
- **Lect7_Loan_Plots.py**

Summary

- Demonstrated examples of numerical and string lists with basic processing tasks
- Showed how to pass the data contained within lists back and forth between functions
- Reviewed reading and writing of lists from/to files
- Defined tuples as immutable lists
- Introduced a more complex multi-dimensional lists
- Finished by introducing the **`matplotlib`** package