* Sequence
* List
* Set
* Tuple

**Exercise 1 : Favorite Numbers**

**Instructions**

1. Create a *set* called my\_fav\_numbers with all your favorites numbers.
2. Add two new numbers to the set.
3. Remove the last number.
4. Create a *set* called friend\_fav\_numbers with your friend’s favorites numbers.
5. Concatenate my\_fav\_numbers and friend\_fav\_numbers to a new variable called our\_fav\_numbers.

**Exercise 2: Tuple**

**Instructions**

1. Given a tuple which value is integers, is it possible to add more integers to the tuple?

**Exercise 3: Print A Range Of Numbers**

**Instructions**

1. Use a for loop to print all numbers from 1 to 20, inclusive.

**Exercise 4: Floats**

**Instructions**

1. Recap – What is a float? What is the difference between an integer and a float?
2. Can you think of another way to generate a sequence of floats?
3. Create a list containing the following sequence 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5 (don’t hard-code the sequence).

**Exercise 5: Shopping List**

**Instructions**

Using this list basket = ["Banana", "Apples", "Oranges", "Blueberries"];

1. Remove “Banana” from the list.
2. Remove “Blueberries” from the list.
3. Add “Kiwi” to the end of the list.
4. Add “Apples” to the beginning of the list.
5. Count how many apples are in the basket.
6. Empty the basket.
7. Print(basket)

**Exercise 6 : Loop**

**Instructions**

1. Write a while loop that will continuously ask the user for their name, unless the input is equal to your name.

**Exercise 7**

**Instructions**

1. Given a list, use a loop to print out every element which has an even index.

**Exercise 8**

**Instructions**

1. Create a loop that goes from 1500 to 2500 and prints all multiples of 5 and 7.

**Exercise 9: Favorite Fruits**

**Instructions**

1. Ask the user to input their favorite fruit(s) (one or several fruits).  
   **Hint** : Use the built in input method. Ask the user to separate the fruits with a single space, eg. "apple mango cherry".
2. Store the favorite fruit(s) in a list (convert the string of words into a list of words).
3. Now that we have a list of fruits, ask the user to input a name of any fruit.
   * If the user’s input is in the favorite fruits list, print “You chose one of your favorite fruits! Enjoy!”.
   * If the user’s input is NOT in the list, print, “You chose a new fruit. I hope you enjoy”.

**Exercise 10: Who Ordered A Pizza ?**

**Instructions**

1. Write a loop that asks a user to enter a series of pizza toppings, when the user inputs ‘quit’ stop asking for toppings.
2. As they enter each topping, print a message saying you’ll add that topping to their pizza.
3. Upon exiting the loop print all the toppings on the pizza pie and what the total price is (10 + 2.5 for each topping).

**Exercise 11: Cinemax**

**Instructions**

1. A movie theater charges different ticket prices depending on a person’s age.
   * if a person is under the age of 3, the ticket is free.
   * if they are between 3 and 12, the ticket is $10.
   * if they are over the age of 12, the ticket is $15.
2. Ask a family the age of each person who wants a ticket.
3. Store the total cost of all the family’s tickets and print it out.
4. A group of teenagers are coming to your movie theater and want to watch a movie that is restricted for people between the ages of 16 and 21.  
   Write a program that asks every user for their age, and prints a list of the teens who are permitted to watch the movie.

**Exercise 12 : Who Is Under 16?**

**Instructions**

1. Given a list of names, write a program that asks every user for their age, if they are less than 16 years old remove them from the list.
2. At the end, print the final list.

**Exercise 13**

**Instructions**

1. Make a list called sandwich\_orders and fill it with the names of various sandwiches .
2. Then make an empty list called finished\_sandwiches.
3. As each sandwich is made, move it to the list of finished sandwiches.
4. After all the sandwiches have been made, print a message listing each sandwich that was made , such as I made your tuna sandwich.

**Exercise 14**

**Instructions**

1. Using the list sandwich\_orders from Exercise 13, make sure the sandwich ‘pastrami’ appears in the list at least three times.
2. Add code near the beginning of your program to print a message saying the deli has run out of pastrami, and then use a while loop to remove all occurrences of ‘pastrami’ from sandwich\_orders.
3. Make sure no pastrami sandwiches end up in finished\_sandwiches.