

Exercises

1. Declare a variable `my_list` and assign it a list of integers. Use a loop to calculate the sum of the numbers in the list.
2. Declare a variable `my_tuple` and assign it a tuple of strings. Use a loop to create a new list that contains the uppercase versions of the strings in the tuple.
3. Declare a variable `my_string` and assign it a string of words. Use string formatting to print the string in title case.
4. Declare a variable `my_list` and assign it a list of strings. Use a loop to create a new list that contains the strings from the original list with all the vowels removed.
5. Declare a variable `my_dict` and assign it a dictionary with at least 3 key-value pairs. Use the `items()` method to print the key-value pairs in the dictionary.
6. Declare a variable `my_string` and assign it a string of words. Use a loop to create a new string that contains the words from the original string in reverse order, separated by spaces.
7. Declare a variable `my_list` and assign it a list of integers. Use a loop to create a new list that contains the square of each number in the original list.
8. Declare a variable `my_set` and assign it a set of strings. Use the `sorted()` function to create a new list that contains the strings from the original set in alphabetical order.
9. Declare a variable `my_list` and assign it a list of strings. Use a loop to create a new list that contains the first letter of each string in the original list.
10. Declare a variable `my_dict` and assign it a dictionary with at least 3 key-value pairs. Use the `keys()` method to create a new list that contains the keys from the dictionary in reverse order.

Exercises and Solution

1. Declare a variable `my_list` and assign it a list of integers. Use a loop to calculate the sum of the numbers in the list.

```
my_list = [1, 2, 3, 4, 5]
```

```
sum = 0
```

```
for num in my_list:
```

```
    sum += num
```

```
print(sum)
```

2. Declare a variable `my_tuple` and assign it a tuple of strings. Use a loop to create a new list that contains the uppercase versions of the strings in the tuple.

```
my_tuple = ("apple", "banana", "orange")
```

```
new_list = []
```

```
for word in my_tuple:
```

```
new_list.append(word.upper())
```

```
print(new_list)
```

3. Declare a variable `my_string` and assign it a string of words. Use string formatting to print the string in title case.

```
my_string = "the quick brown fox"
```

```
print(my_string.title())
```

4. Declare a variable `my_list` and assign it a list of strings. Use a loop to create a new list that contains the strings from the original list with all the vowels removed.

```
my_list = ["apple", "banana", "orange"]
```

```
new_list = []
```

```
vowels = "aeiou"
```

```
for word in my_list:
```

```
    new_word = ""
```

```
    for letter in word:
```

```
        if letter.lower() not in vowels:
```

```
            new_word += letter
```

```
    new_list.append(new_word)
```

```
print(new_list)
```

5. Declare a variable `my_dict` and assign it a dictionary with at least 3 key-value pairs. Use the `items()` method to print the key-value pairs in the dictionary.

```
my_dict = {"name": "John", "age": 30, "city": "New York"}
```

```
for key, value in my_dict.items():
```

```
    print(key, value)
```

6. Declare a variable `my_string` and assign it a string of words. Use a loop to create a new string that contains the words from the original string in reverse order, separated by spaces.

```
my_string = "the quick brown fox"
```

```
words = my_string.split()
```

```
new_string = ""
```

```
for i in range(len(words)-1, -1, -1):
```

```
    new_string += words[i] + " "
```

```
print(new_string.strip())
```

7. Declare a variable `my_list` and assign it a list of integers. Use a loop to create a new list that contains the square of each number in the original list.

```
my_list = [1, 2, 3, 4, 5]
```

```
new_list = []
```

```
for num in my_list:
```

```
    new_list.append(num ** 2)
```

```
print(new_list)
```

8. Declare a variable `my_set` and assign it a set of strings. Use the `sorted()` function to create a new list that contains the strings from the original set in alphabetical order.

```
my_set = {"orange", "apple", "banana"}
```

```
new_list = sorted(list(my_set))
```

```
print(new_list)
```

9. Declare a variable `my_list` and assign it a list of strings. Use a loop to create a new list that contains the first letter of each string in the original list.

```
my_list = ["apple", "banana", "orange"]
```

```
new_list = []
```

```
for word in my_list:
```

```
    new_list.append(word[0])
```

```
print(new_list)
```

10. Declare a variable `my_dict` and assign it a dictionary with at least 3 key-value pairs. Use the `keys()` method to create a new list that contains the keys from the dictionary in reverse order.

```
my_dict = {"name": "John", "age": 30, "city": "New York"}
```

```
new_list = list(reversed(list(my_dict.keys())))
```

```
print(new_list)
```