

Exercises

1. Declare a variable `num` and assign it an integer value. Use a loop to print the even numbers from 0 to `num` (inclusive).
2. Declare a variable `my_dict` and assign it a dictionary with at least 3 key-value pairs. Add a new key-value pair to the dictionary, and then remove one of the existing key-value pairs.
3. Declare a variable `my_string` and assign it a string of words. Use the `split()` method to split the string into a list of words, and then print the list.
4. Declare a variable `my_list` and assign it a list of numbers. Use a loop to calculate the product of the numbers in the list.
5. Declare a variable `my_tuple` and assign it a tuple of integers. Print the maximum and minimum values in the tuple.
6. Declare a variable `my_list` and assign it a list of strings. Use a loop to create a new list that contains the length of each string in the original list.
7. Declare a variable `x` and assign it an integer value. Declare another variable `y` and assign it the value of `x` divided by 2. Print the value of `y`, and then update the value of `x` to be 10. Print the value of `y` again. Declare a variable `my_set1` and assign it a set of numbers.
8. Declare another variable `my_set2` and assign it a set of numbers that includes some (but not all) of the same numbers as `my_set1`. Use set operations to create a new set that contains all the unique numbers from both sets.
9. Declare a variable `my_dict` and assign it a dictionary with at least 3 key-value pairs. Use a loop to print the keys of the dictionary in alphabetical order.
10. 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 in reverse order.

Exercises and Solution

1. Declare a variable `num` and assign it an integer value. Use a loop to print the even numbers from 0 to `num` (inclusive).

```
num = 10
```

```
for i in range(0, num+1, 2):
```

```
    print(i)
```

2. Declare a variable `my_dict` and assign it a dictionary with at least 3 key-value pairs. Add a new key-value pair to the dictionary, and then remove one of the existing key-value pairs.

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

```
my_dict["occupation"] = "programmer"
```

```
del my_dict["age"]
```

```
print(my_dict)
```

3. Declare a variable `my_string` and assign it a string of words. Use the `split()` method to split the string into a list of words, and then print the list.

```
my_string = "The quick brown fox"
```

```
my_list = my_string.split()
```

```
print(my_list)
```

4. Declare a variable `my_list` and assign it a list of numbers. Use a loop to calculate the product of the numbers in the list.

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

```
product = 1
```

```
for num in my_list:
```

```
    product *= num
```

```
print(product)
```

5. Declare a variable `my_tuple` and assign it a tuple of integers. Print the maximum and minimum values in the tuple.

```
my_tuple = (1, 2, 3, 4, 5)
```

```
print(max(my_tuple))
```

```
print(min(my_tuple))
```

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

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

```
new_list = []
```

```
for word in my_list:
```

```
    new_list.append(len(word))
```

```
print(new_list)
```

7. Declare a variable `x` and assign it an integer value. Declare another variable `y` and assign it the value of `x` divided by 2. Print the value of `y`, and then update the value of `x` to be 10. Print the value of `y` again. Declare a variable `my_set1` and assign it a set of numbers.

```
x = 20
```

```
y = x / 2
```

```
print(y)
```

```
x = 10
```

```
print(y)
```

8. Declare another variable `my_set2` and assign it a set of numbers that includes some (but not all) of the same numbers as `my_set1`. Use set operations to create a new set that contains all the unique numbers from both sets.

```
my_set1 = {1, 2, 3, 4, 5}
my_set2 = {3, 4, 5, 6, 7}
new_set = my_set1.union(my_set2)
print(new_set)
```

9. Declare a variable `my_dict` and assign it a dictionary with at least 3 key-value pairs. Use a loop to print the keys of the dictionary in alphabetical order.

```
my_dict = {"name": "John", "age": 30, "city": "New York"}
for key in sorted(my_dict.keys()):
    print(key)
```

10. 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 in reverse order.

```
my_list = ["apple", "banana", "orange"]
new_list = []
for i in range(len(my_list)-1, -1, -1):
    new_list.append(my_list[i])
print(new_list)
```