1-for

June 9, 2022

1 Creating a for Loop

The **for** loop is the most popular looping construct in Python. A **for** loop is created using the following syntax:

```
for x in iterable:
    # do something
```

[1]:

```
# Print each item in list
my_list = [1, 2, 3]
for item in my_list:
    print(item)
```

```
1
2
3
```

[3]:

```
# Double every item in list
my_list = [1, 2, 3]
for item in my_list:
    print(f'{item * 2}')
```

```
2
4
6
```

[6]:

```
# Print only even numbers
my_tup = (1, 2, 3, 4, 5, 6, 7)
for item in my_tup:
    if item % 2 == 0:
        print(f'{item} is even')
```

```
2 is even
4 is even
6 is even
```

[18]:

```
# Iterate over a range
for i in range(1,10):
    print(i, end=" ")
```

```
1 2 3 4 5 6 7 8 9
```

[10]:

```
# Using the len fuction
my_list = ["a","b","c","d","e"]
for i in range(len(my_list)):
    print(i)
```

```
0
1
2
3
4
```

[11]:

```
# Iterate over a sting
my_str = 'abcdefg'
for letter in my_str:
    print(letter)
```

```
a
b
c
d
e
f
g
```

[27]:

```
# Iterate over a dictionary
dict = {'key1': 'value1', 'key2': 'value2', 'key3': 'value3'}
for key in dict.keys():
    print(key)
```

```
key1
key2
key3
```

[16]:

```
# Iterate over list of tuples
list_of_tuples = [(1, 'banana'), (2, 'apple'), (3, 'pear')]
for number, fruit in list_of_tuples:
    print(f'{number} - {fruit}')
```

```
1 - banana
2 - apple
3 - pear
```

[17]:

```
# Shorthand used in a lab
var = 1
var += 1  # shorthand for 'var = var + 1'
print(var)
```

```
2
```

[ ]:

```
```