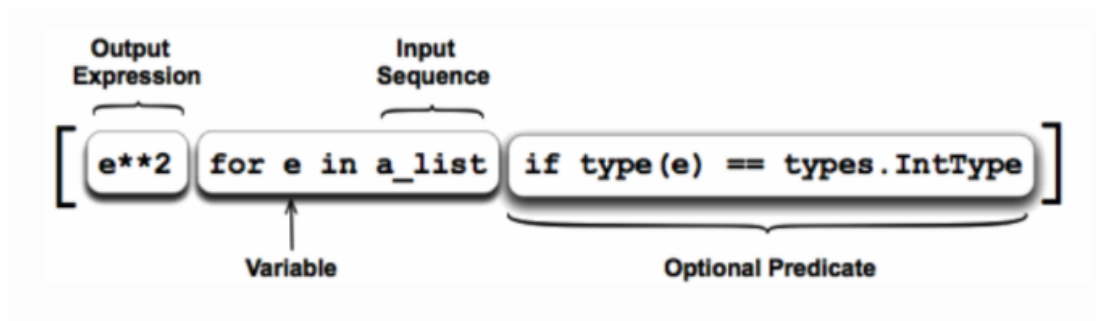


Comprehensions

Comprehensions are a shorter and more concise way to create new sequences in python, most of the time with one line of code, without using a classic for loop. There are four types of comprehensions: list, dictionary, set and generator comprehensions.

The components of a list comprehension are the following:



List comprehension

Basically we have three components:

1. The output expression where we can alter the variables
2. The sequence generation expression with for where we define the sequence
3. The optional conditional expression that allows us to put an inclusion condition on a variable

```

75 new_list = [x for x in range(10)] #list comprehension
76 print(new_list)
77 # output => [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
78
79 new_dict = {x:x+1 for x in range(10)} #dictionary comprehension
80 print(new_dict)
81 # output => {0: 1, 1: 2, 2: 3, 3: 4, 4: 5, 5: 6, 6: 7, 7: 8, 8: 9, 9: 10}
82
83 new_set = {x for x in range(10)} #set comprehension
84 print(new_set)
85 # output => {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
86
87 new_gen = (x for x in range(10)) #generator comprehension
88 print(new_gen)
89 # output => <generator object <genexpr> at 0x10556e6d0>

```

```

12 numbers = [1,1,2,8,4,5,3,8,2,1,6,3,8,1,6,8]
13
14 eights = [x for x in numbers if x == 8]
15 count_eights = len(eights)
16 print(f'The number of eights: {count_eights}')

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: Python

```

→ ~ /opt/homebrew/bin/python3 /Users/tepe/Desktop/GitHub/VSCODE/PY
The number of eights: 4

```