```
Python 3.14.0 (tags/v3.14.0:ebf955d, Oct  7 2025, 10:15:03) [MSC v.1944 64 bit
(AMD64)] on win32
Enter "help" below or click "Help" above for more information.
#======LIST COMPREHENSION=========
# BASIC PART - SINGLE VARIABLE LIST COMPREHENSION
# Version 1: Split the iterable and convert it into a list
# Let us split iterable range(1, 11) into its constituent elements
# and make all elements a part of list
L = [x for x in range(1, 11)]
print(L)
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
# General Syntax of Version 1:
# L = [v for v in iterable] # v -> variable name of your choice
# iterable -> any iterable object of your choice
# Iterable -> object whose class contains implementation of __iter__()
# Iterable object is by definition compatible with for loop

# Version 2
L = [x for x in range(1, 11) if x % 2 == 0]
print(L)
[2, 4, 6, 8, 10]
# GENERAL SYNTAX OF VERSION 2
# L = [v for v in iterable if cond(v)]
# v -> varaible name of your choice
# iterable -> any iterable object of your choice
# cond(v) is any boolean expression involving variable v
# Evaluation:
# Step 1: Split the iterable into its constituent elements
# Step 2: Apply cond(v) on each element derived in Step 1
# Step 3: Finalize the list with all elements in step 2 for whom answer was True.

# Version 3:
L = [x**2 for x in range(1, 11)]
print(L)
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
# Genralized version:
# L = [f(v) for v in iterable]
# where v-> variable name of your choice, iterable -> iterable object of your
choice
# f -> function accepting one variable and returning some value or it can simply
# be a right hand side expression involving 'v'
# Evaluation:
>>> # Step 1: Split the iterable into its constituent elements
>>> # Step 2: Apply function f on all values derived in Step 1
>>> # Step 3: Make a final list of the results of applying function f on all values
in step 2
>>>
>>> # Version 4:
>>> L = [x**2 for x in range(1, 11) if x % 2 == 0]
>>> print(L)
```

```
[4, 16, 36, 64, 100]
>>> # Generalised version:
>>> # L = [f(v) for v in iterable if cond(v)]
>>> # Evaluation:
>>> # Step 1: Split the iterable into its constituent elements
>>> # Step 2: Apply cond(v) on all values derived in Step 1
>>> # Step 3: Create temporary list of all values which satisfied the condition in
Step 2
>>> # Step 4: Apply function 'f' on all values in the temporary list formed in Step
3 and make
>>> # final list out of it.
>>> # You can achive this result combining map and filter
>>> C =
SyntaxError: invalid syntax
>>> C = '''
... L = [f(v) for v in iterable if cond(v)]
... ==
... list(map(f, filter(cond, iterable)))
... '''
>>> L = [x**2 for x in range(1, 11) if x % 2 == 0]
>>> print(L)
[4, 16, 36, 64, 100]
>>> L_map_filter = list(map(lambda n : n ** 2, filter(lambda m : m % 2 == 0,
range(1, 11))))
>>> L_map_filter
[4, 16, 36, 64, 100]
```