# Passing a List as an Argument

def my\_fun(food):

    for x in food:

        print(x)

fruits = ["apple", "banana", "cherry"]

my\_fun(fruits)

#  Arbitrary Arguments, \*args

# If you do not know how many arguments that will be passed into your function, add a \* before the parameter name in the function definition.

# This way the function will receive a tuple of arguments, and can access the items accordingly:

def my\_function(\*kids):

  print("The youngest child is " + kids[2])

my\_function("Ram", "Raj", "Ratan")

# Keyword Arguments

# You can also send arguments with the key = value syntax.

# This way the order of the arguments does not matter.

def my\_function(child3, child2, child1):

  print("The youngest child is " + child3)

my\_function(child1 = "Ram", child2 = "Raj", child3 = "Ratan")

# Arbitrary Keyword Arguments, \*\*kwargs

# If you do not know how many keyword arguments that will be passed into your function, add two asterisk: \*\* before the parameter name in the function definition.

# This way the function will receive a dictionary of arguments, and can access the items accordingly:

def my\_function(\*\*kid):

  print("His last name is " + kid["lname"])

my\_function(fname = "Ram", lname = "Kumar")

# Unpacking Argument Lists using '\*' operator

print(list(range(1, 6)))            # normal call with separate arguments

args = [1, 6]

print(list(range(\*args)))            # call with arguments unpacked from a list

# In the same fashion, dictionaries can deliver keyword arguments with the \*\*-operator:

def fun(a, b='hunt', c='sky'):

    print(a, 'can', b, 'in the', c)

d = {'a': 'falcons', 'b': 'kill', 'c': 'SKY'}

fun(d)

fun(\*d)

fun(\*\*d)

fun(d['a'])

# passing dict values as arguments

def fun(a, b, c):

    print(a, 'can', b, 'in the', c)

d = {'a': 'falcons', 'b': 'kill', 'c': 'SKY'}

fun(\*\*d)

# Any formal parameters which occur after the \*args parameter are ‘keyword-only’ arguments, meaning that they can only be used as keywords rather than positional arguments.

def concat(\*args, sep="/"):

    print(sep.join(args))

concat("earth", "mars", "venus")

concat("earth", "mars", "venus", sep=".")