# **Random sampling from a list in Python (random.choice, sample, choices)**

In Python, you can randomly sample elements from a list with **the choice(), sample(), and choices()** **functions from the random module.** These functions can also be used with strings and tuples.

**choice()** returns a single random element, while **sample()** and **choices()** return a list of multiple random elements. **sample()** is for random sampling without replacement, whereas **choices()** is for random sampling with replacement.

**Contents**

* [Pick a random element: random.choice()](https://note.nkmk.me/en/python-random-choice-sample-choices/#pick-a-random-element-randomchoice)
* [Random sample without replacement: random.sample()](https://note.nkmk.me/en/python-random-choice-sample-choices/#random-sample-without-replacement-randomsample)
* [Random sample with replacement: random.choices()](https://note.nkmk.me/en/python-random-choice-sample-choices/#random-sample-with-replacement-randomchoices)
* [Fix the random seed with random.seed()](https://note.nkmk.me/en/python-random-choice-sample-choices/#fix-the-random-seed-with-randomseed)

For information on selecting elements from a list based on specific conditions, refer to the following article.

* [Extract, replace, convert elements of a list in Python](https://note.nkmk.me/en/python-list-select-replace/)

If you want to shuffle an entire list or create a list of random numbers, see the following articles.

* [Shuffle a list, string, tuple in Python (random.shuffle, sample)](https://note.nkmk.me/en/python-random-shuffle/)
* [Generate random int/float in Python (random, randrange, randint, etc.)](https://note.nkmk.me/en/python-random-randrange-randint/)

# **Pick a random element: random.choice()**

random.choice() returns a random element from a list.

* [random.choice() — Generate pseudo-random numbers — Python 3.11.3 documentation](https://docs.python.org/3/library/random.html#random.choice)

import random

l **=** [0, 1, 2, 3, 4]

print(random**.**choice(l))

# 1

Tuples and strings are also handled similarly. When a string is provided, one character is returned.

print(random**.**choice(('xxx', 'yyy', 'zzz')))

# yyy

print(random**.**choice('abcde'))

# b

An error is raised if the list, tuple, or string is empty.

# print(random.choice([]))

# IndexError: Cannot choose from an empty sequence

# **Random sample without replacement: random.sample()**

random.sample() randomly samples multiple elements from a list without replacement. Call it with a list and the desired number of elements to return a list.

* [random.sample — Generate pseudo-random numbers — Python 3.11.3 documentation](https://docs.python.org/3/library/random.html#random.sample)

import random

l **=** [0, 1, 2, 3, 4]

print(random**.**sample(l, 3))

# [3, 1, 0]

print(type(random**.**sample(l, 3)))

# <class 'list'>

If the second argument is set to 1, a list with one element is returned. If set to 0, an empty list is returned. If set to a value that exceeds the number of elements of the list, an error is raised.

print(random**.**sample(l, 1))

# [1]

print(random**.**sample(l, 0))

# []

# print(random.sample(l, 10))

# ValueError: Sample larger than population or is negative

Even when you pass a tuple or a string as the first argument, a list is returned.

print(random**.**sample(('xxx', 'yyy', 'zzz'), 2))

# ['zzz', 'xxx']

print(random**.**sample('abcde', 2))

# ['c', 'd']

Use tuple() or join() to convert a list into a tuple or a string, respectively.

* [Convert list and tuple to each other in Python](https://note.nkmk.me/en/python-list-tuple-convert/)
* [Concatenate strings in Python (+ operator, join, etc.)](https://note.nkmk.me/en/python-string-concat/)

print(tuple(random**.**sample(('xxx', 'yyy', 'zzz'), 2)))

# ('zzz', 'yyy')

print(''**.**join(random**.**sample('abcde', 2)))

# be

Note that if the original list or tuple contains duplicate elements, the same values may be selected.

l\_dup **=** [0, 0, 0, 1, 1, 1, 2, 2, 2, 3, 3, 3]

print(random**.**sample(l\_dup, 3))

# [2, 0, 0]

If you want to avoid duplicate values, use set() to convert lists and tuples to sets, extract unique elements, and then use sample().

* [Remove/extract duplicate elements from list in Python](https://note.nkmk.me/en/python-list-unique-duplicate/)

print(set(l\_dup))

# {0, 1, 2, 3}

print(random**.**sample(list(set(l\_dup)), 3))

# [0, 2, 1]

Starting from Python 3.11, directly specifying set as the first argument to sample() will result in an error. It is necessary to explicitly convert it to a list or a similar data structure, as demonstrated above.

# **Random sample with replacement: random.choices()**

random.choices() randomly samples multiple elements from a list with replacement.

* [random.choices() — Generate pseudo-random numbers — Python 3.11.3 documentation](https://docs.python.org/3/library/random.html#random.choices)

Specify the number of elements you want with the k argument. Since elements are chosen with replacement, k can be larger than the number of elements in the original list.

Since k is a keyword-only argument, it is necessary to specify it like k=3.

import random

l **=** [0, 1, 2, 3, 4]

print(random**.**choices(l, k**=**3))

# [2, 1, 0]

print(random**.**choices(l, k**=**10))

# [3, 4, 1, 4, 4, 2, 0, 4, 2, 0]

k is set to 1 by default. If omitted, a list with one element is returned.

print(random**.**choices(l))

# [1]

You can specify the weight (probability) for each element with the weights argument. The type of the list element specified in weights can be either int or float. If set to 0, the element is not selected.

print(random**.**choices(l, k**=**3, weights**=**[1, 1, 1, 10, 1]))

# [0, 2, 3]

print(random**.**choices(l, k**=**3, weights**=**[1, 1, 0, 0, 0]))

# [0, 1, 1]

Cumulative weights can be specified with the cum\_weights argument. cum\_weights in the following sample code is equivalent to the first weights in the above code.

print(random**.**choices(l, k**=**3, cum\_weights**=**[1, 2, 3, 13, 14]))

# [3, 2, 3]

By default, both weights and cum\_weights are set to None, so each element is selected with the same probability.

An error is raised if the length (number of elements) of weights or cum\_weights doesn't match that of the original list.

# print(random.choices(l, k=3, weights=[1, 1, 1, 10, 1, 1, 1]))

# ValueError: The number of weights does not match the population\_

Also, an error is raised if you specify weights and cum\_weights simultaneously.

# print(random.choices(l, k=3, weights=[1, 1, 1, 10, 1], cum\_weights=[1, 2, 3, 13, 14]))

# TypeError: Cannot specify both weights and cumulative weights

Like the previous functions, random.choices() also works with tuples and strings, not just lists.

# **Fix the random seed with random.seed()**

You can fix the random seed and initialize the random number generator with random.seed().

* [random.seed() — Generate pseudo-random numbers — Python 3.11.3 documentation](https://docs.python.org/3/library/random.html#random.seed)

After initializing with the same seed, elements are selected in the same way.

random**.**seed(0)

print(random**.**choice(l))

# 3

random**.**seed(0)

print(random**.**choice(l))

# 3

# **OTHER LINKS :-**

**Related Categories**

* [Python](https://note.nkmk.me/en/python/)
* [List](https://note.nkmk.me/en/list/)

**Related Articles**

* [How to flatten a list of lists in Python](https://note.nkmk.me/en/python-list-flatten/)
* [Pretty-print with pprint in Python](https://note.nkmk.me/en/python-pprint-pretty-print/)
* [Extract common/non-common/unique elements from multiple lists in Python](https://note.nkmk.me/en/python-list-common/)
* [Initialize a list with given size and values in Python](https://note.nkmk.me/en/python-list-initialize/)
* [Convert a list of strings and a list of numbers to each other in Python](https://note.nkmk.me/en/python-list-str-num-conversion/)
* [Unpack a tuple and list in Python](https://note.nkmk.me/en/python-tuple-list-unpack/)
* [Sort a list of dictionaries by the value of the specific key in Python](https://note.nkmk.me/en/python-dict-list-sort/)
* [Sort a list of numeric strings in Python](https://note.nkmk.me/en/python-sort-num-str/)
* [Convert numpy.ndarray and list to each other](https://note.nkmk.me/en/python-numpy-list/)
* [Deque with collections.deque in Python](https://note.nkmk.me/en/python-collections-deque/)
* [Count elements in a list with collections.Counter in Python](https://note.nkmk.me/en/python-collections-counter/)
* [How to start enumerate() at 1 in Python](https://note.nkmk.me/en/python-enumerate-start/)
* [Unpack and pass list, tuple, dict to function arguments in Python](https://note.nkmk.me/en/python-argument-expand/)
* [Get the n-largest/smallest elements from a list in Python](https://note.nkmk.me/en/python-max-min-heapq-nlargest-nsmallest/)
* [Remove an item from a list in Python (clear, pop, remove, del)](https://note.nkmk.me/en/python-list-clear-pop-remove-del/)

**Categories**

* [**Python**](https://note.nkmk.me/en/python/)
  + [NumPy](https://note.nkmk.me/en/numpy/)
  + [OpenCV](https://note.nkmk.me/en/opencv/)
  + [pandas](https://note.nkmk.me/en/pandas/)
  + [Pillow](https://note.nkmk.me/en/pillow/)
  + [pip](https://note.nkmk.me/en/pip/)
  + [scikit-image](https://note.nkmk.me/en/scikit-image/)
* [Jupyter Notebook](https://note.nkmk.me/en/jupyter-notebook/)
* [Image Processing](https://note.nkmk.me/en/image-processing/)
* [File](https://note.nkmk.me/en/file/)
* [JSON](https://note.nkmk.me/en/json/)
* [Date and time](https://note.nkmk.me/en/date-and-time/)
* [String](https://note.nkmk.me/en/string/)
* [Regex](https://note.nkmk.me/en/regex/)
* [Numeric](https://note.nkmk.me/en/numeric/)
* [Dictionary](https://note.nkmk.me/en/dictionary/)
* [**List**](https://note.nkmk.me/en/list/)
* [Error handling](https://note.nkmk.me/en/error-handling/)
* [Mathematics](https://note.nkmk.me/en/mathematics/)
* [Summary](https://note.nkmk.me/en/summary/)

**About**

* GitHub: [nkmk](https://github.com/nkmk)

**Sponsored Link**

**Related Articles**

* [How to flatten a list of lists in Python](https://note.nkmk.me/en/python-list-flatten/)
* [Pretty-print with pprint in Python](https://note.nkmk.me/en/python-pprint-pretty-print/)
* [Extract common/non-common/unique elements from multiple lists in Python](https://note.nkmk.me/en/python-list-common/)
* [Initialize a list with given size and values in Python](https://note.nkmk.me/en/python-list-initialize/)
* [Convert a list of strings and a list of numbers to each other in Python](https://note.nkmk.me/en/python-list-str-num-conversion/)

# **GIT NOTE LINK :-**

* [python-snippets/notebook at 3abe98e7a38a984db90d0e608c4d5569f2984408 · nkmk/python-snippets · GitHub](https://github.com/nkmk/python-snippets/tree/3abe98e7a38a984db90d0e608c4d5569f2984408/notebook)
* [python-snippets/notebook/random\_choice.py at 3abe98e7a38a984db90d0e608c4d5569f2984408 · nkmk/python-snippets · GitHub](https://github.com/nkmk/python-snippets/blob/3abe98e7a38a984db90d0e608c4d5569f2984408/notebook/random_choice.py#L1-L6)