

Forum:[Hack Forums](#)

Thread: [How to make random pairs of elements in a list?](#)

E V O - March 28, 2019, 4:06 pm

I'm trying to randomly pair two elements from a list. The list contains more lists as elements. The random pair serves as input for a function. What would be the best approach to make this pairing as random as possible? In the end I want to iterate the process.

datguy.dev - March 28, 2019, 5:49 pm

(Mar 28, 2019 04:06 PM)E V O Wrote: [\[CITE\]](#) I'm trying to randomly pair two elements from a list. The list contains more lists as elements. The random pair serves as input for a function. What would be the best approach to make this pairing as random as possible? In the end I want to iterate the process.

Can't really without the application. Are you expecting any element of a list, and nested lists, to be viable for all arguments of the function? You could extrapolate on:

```
Code seed = randrange(randrange(randrange(0, len(some_list_with_lists) ),
len(some_list_with_lists)*2), len(some_list_with_lists)*3)
```

AeroBlue - March 28, 2019, 7:46 pm

I would look into [itertools](#) and [random](#)

Idk if this is what you are looking for but I made this:

```
Code list1= ["1","2","3"],["a","b","c"],["x","y","z"],["4","5","6"]
def pair(list, r):
    return list(itertools.combinations(list_, r))
```

```
combinations = pair(list, 2)
random.choice(combinations)Code Output:(['a', 'b', 'c'], ['4', '5', '6'])
```

If you wanted to not iterate through the combination again you could do something like this:

```
Code combination = random.choice(combinations)
combinations.remove(combination)
```

Dark - March 28, 2019, 10:59 pm

Just use the random package....