

# Python List Length and Element Checks and Defaults

Fan Wang

2020-10-24

## Contents

<b>1</b>	<b>List</b>	<b>1</b>
1.1	Loop Through a List get Value and Index	1
1.2	Parse Elements of a List	1
1.3	Check if Any Element of a List is another List	2
1.4	Convert a List to a String List	2
1.5	Concatenate a List of Strings with Join	3
1.6	Concatenate a List to a String with Separator	3
1.7	Add Nth Element to List when Nth Element Does not Exist	3
1.8	Check If List Has N Elements of None for Some Elements	4
1.9	Add a Default Value to Nth Element of List	4

## 1 List

Go to the [RMD](#), [PDF](#), or [HTML](#) version of this file. Go back to [Python Code Examples](#) Repository ([bookdown site](#)) or the [pyfan](#) Package ([API](#)).

```
import numpy as np
```

### 1.1 Loop Through a List get Value and Index

There is a list, loop through it, get out both the index and the value of each indexed element together.

```
ls_ob_combo_type = ["e", "20201025x_esr_list_tKap_mlt_ce1a2", ["esti_param.kappa_ce9901", "esti_param.kappa_ce0209"], "C1E31M3S3=1"]
for it_idx, ob_val in enumerate(ls_ob_combo_type):
    print(f'{it_idx=} and {ob_val=}')

## it_idx=0 and ob_val='e'
## it_idx=1 and ob_val='20201025x_esr_list_tKap_mlt_ce1a2'
## it_idx=2 and ob_val=['esti_param.kappa_ce9901', 'esti_param.kappa_ce0209']
## it_idx=3 and ob_val=1
## it_idx=4 and ob_val='C1E31M3S3=1'
```

### 1.2 Parse Elements of a List

A list has multiple elements, deal them out.

```
list_test = ['C1E126M4S3', 2]
[compesti_short_name, esti_top_which] = list_test
print(f'{compesti_short_name=} and {esti_top_which=}')

## compesti_short_name='C1E126M4S3' and esti_top_which=2
```

### 1.3 Check if Any Element of a List is another List

There is a list with two elements, there is another list with say four elements. Check if any of the two element is in the list with four elements.

```
ar_int_list_a1 = [1,2]
ar_int_list_a2 = [2,1111]
ar_int_list_a3 = [2111,1111]
ar_int_list_b = [1,2,3,11,999]
ar_int_list_c = [2]

check_any_a1_in_b = any(item in ar_int_list_a1 for item in ar_int_list_b)
check_any_a2_in_b = any(item in ar_int_list_a2 for item in ar_int_list_b)
check_any_a3_in_b = any(item in ar_int_list_a3 for item in ar_int_list_b)
check_any_a1_in_c = any(item in ar_int_list_a1 for item in ar_int_list_c)

print(f'{check_any_a1_in_b=}')

## check_any_a1_in_b=True
print(f'{check_any_a2_in_b=}')

## check_any_a2_in_b=True
print(f'{check_any_a3_in_b=}')

## check_any_a3_in_b=False
print(f'{check_any_a1_in_c=}')

## check_any_a1_in_c=True
```

### 1.4 Convert a List to a String List

Given a list of string and numeric values, convert to a list of string values. The MAP function is like [apply](#) in R.

- [How to concatenate items in a list to a single string?](#)

```
ls_spec_key = ['ng_s_d', 'esti_test_11_simu', 2, 3]
ls_st_spec_key = list(map(str, ls_spec_key))
print(ls_st_spec_key)
```

```
## ['ng_s_d', 'esti_test_11_simu', '2', '3']
```

Additionally, append some common element to each element of the string using MAP.

```
ls_spec_key = ['ng_s_d', 'esti_test_11_simu', 2, 3]
ls_st_spec_key = list(map(lambda x: 'add++' + str(x), ls_spec_key))
print(ls_st_spec_key)
```

```
## ['add++ng_s_d', 'add++esti_test_11_simu', 'add++2', 'add++3']
```

Equivalently, via list comprehension

```
ls_spec_key = ['ng_s_d', 'esti_test_11_simu', 2, 3]
ls_st_spec_key = ['list_comprehension' + str(spec_key) for spec_key in ls_spec_key]
print(ls_st_spec_key)
```

```
## ['list_comprehensionng_s_d', 'list_comprehensionesti_test_11_simu', 'list_comprehension2', 'list_comprehension3']
```

## 1.5 Concatenate a List of Strings with Join

Concatenate a list of strings together, with possible empty values. Note the *filter(None, ls)* structure below.

```
ls_model_assumption = ['', 'ITG', 'GE', 'ITG_GE']
ls_graph_panda_list_name = ['min_graphs', 'main_aA_graphs', 'all_graphs_tables']
for model_assumption in ls_model_assumption:
    for graph_panda_list_name in ls_graph_panda_list_name:
        # Join strings together
        st_curdatestr = "_".join(filter(None, [model_assumption, graph_panda_list_name]))
        print(f'{st_curdatestr=}')

## st_curdatestr='min_graphs'
## st_curdatestr='main_aA_graphs'
## st_curdatestr='all_graphs_tables'
## st_curdatestr='ITG_min_graphs'
## st_curdatestr='ITG_main_aA_graphs'
## st_curdatestr='ITG_all_graphs_tables'
## st_curdatestr='GE_min_graphs'
## st_curdatestr='GE_main_aA_graphs'
## st_curdatestr='GE_all_graphs_tables'
## st_curdatestr='ITG_GE_min_graphs'
## st_curdatestr='ITG_GE_main_aA_graphs'
## st_curdatestr='ITG_GE_all_graphs_tables'
```

## 1.6 Concatenate a List to a String with Separator

Given a list of strings and numeric data types, concatenate list to a string with some separator. Also in reverse, generate a list by breaking a string joined by some separator.

```
ls_spec_key = ['ng_s_d', 'esti_test_11_simu', 2, 3]
st_separator = '='
st_spec_key = st_separator.join(list(map(lambda x : str(x), ls_spec_key)))
print(st_spec_key)
```

```
## ng_s_d=esti_test_11_simu=2=3
```

Now break string apart:

```
st_spec_key = '='.join(list(map(lambda x : '$' + str(x) + '$', ['ng_s_d', 'esti_test_11_simu', 2, 3])))
print(st_spec_key.split('='))
```

```
## ['$ng_s_d$', '$esti_test_11_simu$', '$2$', '$3$']
```

## 1.7 Add Nth Element to List when Nth Element Does not Exist

There is a list with 2 elements, check if the list has 3 elements, if not, add another element.

```
ls_string_A = ['c', '20180918']
ls_string_B = ['c', '20180918', ['esti_param.alpha_k']]

for ls_string in [ls_string_A, ls_string_B]:
    if len(ls_string) == 2:
        ls_string.insert(2, None)

    print(ls_string)
```

```
## ['c', '20180918', None]
## ['c', '20180918', ['esti_param.alpha_k']]
```

## 1.8 Check If List Has N Elements of None for Some Elements

In the example below, for A, B and C, do something, for D and E do something else.

```
ls_string_A = ['c', '20180918']
ls_string_B = ['c', '20180918', None]
ls_string_C = ['c', '20180918', None, None]
ls_string_D = ['c', '20180918', ['esti_param.alpha_k'], None]
ls_string_E = ['c', '20180918', ['esti_param.alpha_k'], 5]

for ls_string in [ls_string_A, ls_string_B, ls_string_C, ls_string_D, ls_string_E]:
    if len(ls_string) >= 3 and ls_string[2] is not None:
        print(ls_string)
    else:
        print(ls_string[0:2])
```

```
## ['c', '20180918']
## ['c', '20180918']
## ['c', '20180918']
## ['c', '20180918', ['esti_param.alpha_k'], None]
## ['c', '20180918', ['esti_param.alpha_k'], 5]
```

## 1.9 Add a Default Value to Nth Element of List

There is a string list with potential potentially three elements. But sometimes the input only has two elements. Provide default third element value if third element is NONE or if the string list only has two elements.

```
ls_string_A = ['c', '20180918']
ls_string_B = ['c', '20180918', None]
ls_string_C = ['c', '20180918', ['esti_param.alpha_k']]

for ls_string in [ls_string_A, ls_string_B, ls_string_C]:

    if len(ls_string) <= 2:
        # Deals with situation A
        ls_string.append(['esti_param.alpha_k'])
    elif ls_string[2] is None:
        # Deals with situation B
        ls_string[2] = ['esti_param.alpha_k']
    else:
        # Situation C
        pass

    print(ls_string)
```

```
## ['c', '20180918', ['esti_param.alpha_k']]
## ['c', '20180918', ['esti_param.alpha_k']]
## ['c', '20180918', ['esti_param.alpha_k']]
```

Now do the same thing for a numeric list:

```

ls_string_A = [11, 22]
ls_string_B = [11, 22, None]
ls_string_C = [11, 22, 33]

for ls_string in [ls_string_A, ls_string_B, ls_string_C]:

    if len(ls_string) <= 2:
        # Deals with situation A
        ls_string.append(33)
    elif ls_string[2] is None:
        # Deals with situation B
        ls_string[2] = 33
    else:
        # Situation C
        pass

    print(ls_string)

## [11, 22, 33]
## [11, 22, 33]
## [11, 22, 33]

```