

Python Dictionary Examples and Usages

Fan Wang

2020-05-23

Contents

1 Dictionary	1
1.1 Loop Through a Dictionary	1
1.2 Select One Key-value Pair	1
1.3 Copying Dictionary and Updating Copied Dictionary	2
1.4 Create a List of Dictionaries	3
1.5 Iteratively Add to A Dictionary	4
1.6 Select by Keys Dictionaries from list of Dictionaries	4
1.7 Drop Element of Dictionary	5

1 Dictionary

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 pprint
import copy as copy
```

1.1 Loop Through a Dictionary

Given a dictionary, loop through all of its elements

```
dc_speckey_dict = {0: 'mpoly_1',
                   1: 'ng_s_t',
                   2: 'ng_s_d',
                   3: 'ng_p_t',
                   4: 'ng_p_d'}
for speckey_key, speckey_val in dc_speckey_dict.items():
    print('speckey_key:' + str(speckey_key) + ', speckey_val:' + speckey_val)
```

```
## speckey_key:0, speckey_val:mpoly_1
## speckey_key:1, speckey_val:ng_s_t
## speckey_key:2, speckey_val:ng_s_d
## speckey_key:3, speckey_val:ng_p_t
## speckey_key:4, speckey_val:ng_p_d
```

1.2 Select One Key-value Pair

Given a dictionary, select a single key-value pair, based on either the key or the value.

```
# select by key
ls_it_keys = [0, 4]
```

```

dc_speckey_dict_select_by_key = {it_key: dc_speckey_dict[it_key] for it_key in ls_it_keys}
print(f'{dc_speckey_dict_select_by_key=}')
# select by value

## dc_speckey_dict_select_by_key={0: 'mpoly_1', 4: 'ng_p_d'}
ls_st_keys = ['ng_s_d', 'ng_p_d']
dc_speckey_dict_select_by_val = {it_key: st_val for it_key, st_val in dc_speckey_dict.items()
                                if st_val in ls_st_keys}
print(f'{dc_speckey_dict_select_by_val=}')

## dc_speckey_dict_select_by_val={2: 'ng_s_d', 4: 'ng_p_d'}
See Get key by value in dictionary.

```

1.3 Copying Dictionary and Updating Copied Dictionary

First, below, it looks as if the default dictionary has been copied, and that the updates to the dictionary will only impact the `dc_invoke_main_args`, but that is not the case:

```

# list update
dc_invoke_main_args_default = {'speckey': 'ng_s_t',
                              'ge': False,
                              'multiprocess': False,
                              'estimate': False,
                              'graph_panda_list_name': 'min_graphs',
                              'save_directory_main': 'simu',
                              'log_file': False,
                              'log_file_suffix': ''}

dc_invoke_main_args = dc_invoke_main_args_default
dc_invoke_main_args['speckey'] = 'b_ge_s_t_bis'
dc_invoke_main_args['ge'] = True
print(f'speckey in dc_invoke_main_args is {dc_invoke_main_args["speckey"]}.')

## speckey in dc_invoke_main_args is b_ge_s_t_bis.
print(f'speckey in dc_invoke_main_args_default is {dc_invoke_main_args_default["speckey"]}.')

## speckey in dc_invoke_main_args_default is b_ge_s_t_bis.

```

Now this has the intended result. After updating the deep-copied dictionary, the key-values in the original dictionary are preserved:

```

# list update
dc_invoke_main_args_default = {'speckey': 'ng_s_t',
                              'ge': False,
                              'multiprocess': False,
                              'estimate': False,
                              'graph_panda_list_name': 'min_graphs',
                              'save_directory_main': 'simu',
                              'log_file': False,
                              'log_file_suffix': ''}

# deep copy and update
dc_invoke_main_args = copy.deepcopy(dc_invoke_main_args_default)
dc_invoke_main_args['speckey'] = 'b_ge_s_t_bis'
dc_invoke_main_args['ge'] = True
print(f'speckey in dc_invoke_main_args_default is {dc_invoke_main_args_default["speckey"]}.')

```

```

## speckey in dc_invoke_main_args_default is ng_s_t.
print(f'speckey in dc_invoke_main_args is {dc_invoke_main_args["speckey"]}.'.')
# deep copy and update again

## speckey in dc_invoke_main_args is b_ge_s_t_bis.
dc_invoke_main_args = copy.deepcopy(dc_invoke_main_args_default)
dc_invoke_main_args['speckey'] = 'b_ge_s_t_bis_new'
dc_invoke_main_args['ge'] = False
print(f'speckey in dc_invoke_main_args is {dc_invoke_main_args["speckey"]}.'.')

## speckey in dc_invoke_main_args is b_ge_s_t_bis_new.


- copy and deepcopy
- Deep copy of a dict in python

```

1.4 Create a List of Dictionaries

```

import datetime
import pprint
ls_dc_exa = [
    {"file": "mat_matlab",
     "title": "One Variable Graphs and Tables",
     "description": "Frequency table, bar chart and histogram",
     "val": 1,
     "date": datetime.date(2020, 5, 2)},
    {"file": "mat_two",
     "title": "Second file",
     "description": "Second file.",
     "val": [1, 2, 3],
     "date": datetime.date(2020, 5, 2)},
    {"file": "mat_algebra_rules",
     "title": "Opening a Dataset",
     "description": "Opening a Dataset.",
     "val": 1.1,
     "date": datetime.date(2018, 12, 1)}
]
pprint.pprint(ls_dc_exa, width=1)

## [{'date': datetime.date(2020, 5, 2),
##   'description': 'Frequency '
##                 'table, '
##                 'bar '
##                 'chart '
##                 'and '
##                 'histogram',
##   'file': 'mat_matlab',
##   'title': 'One '
##            'Variable '
##            'Graphs '
##            'and '
##            'Tables',
##   'val': 1},
##  {'date': datetime.date(2020, 5, 2),
##   'description': 'Second '

```

```

##             'file.',
## 'file': 'mat_two',
## 'title': 'Second '
##             'file',
## 'val': [1,
##         2,
##         3]},
## {'date': datetime.date(2018, 12, 1),
##  'description': 'Opening '
##                'a '
##                'Dataset.',
##  'file': 'mat_algebra_rules',
##  'title': 'Opening '
##          'a '
##          'Dataset',
##  'val': 1.1}]

```

1.5 Iteratively Add to A Dictionary

Iteratively add additional Key and Value pairs to a dictionary.

```

ls_snm_tex = ["file1.tex", "file2.tex", "file3.tex"]
ls_snm_pdf = ["file1.pdf", "file2.pdf", "file3.pdf"]

dc_tex_pdf = {}
for tex, pdf in zip(ls_snm_tex, ls_snm_pdf):
    dc_tex_pdf[tex] = pdf

pprint.pprint(dc_tex_pdf, width=1)

## {'file1.tex': 'file1.pdf',
##  'file2.tex': 'file2.pdf',
##  'file3.tex': 'file3.pdf'}

```

1.6 Select by Keys Dictionaries from list of Dictionaries

Given a list of dictionary, search if key name is in list:

```

# string to search through
ls_str_file_ids = ['mat_matlab', 'mat_algebra_rules']
# select subset
ls_dc_selected = [dc_exa
                  for dc_exa in ls_dc_exa
                  if dc_exa['file'] in ls_str_file_ids]
# print
pprint.pprint(ls_dc_selected, width=1)

## [{'date': datetime.date(2020, 5, 2),
##   'description': 'Frequency '
##                 'table, '
##                 'bar '
##                 'chart '
##                 'and '
##                 'histogram',
##   'file': 'mat_matlab',
##   'title': 'One '

```

```

##          'Variable '
##          'Graphs '
##          'and '
##          'Tables',
##  'val': 1},
## {'date': datetime.date(2018, 12, 1),
##  'description': 'Opening '
##                'a '
##                'Dataset.',
##  'file': 'mat_algebra_rules',
##  'title': 'Opening '
##          'a '
##          'Dataset',
##  'val': 1.1}]

```

Search and Select by Multiple Keys in Dictionary. Using two keys below:

```

# string to search through
ls_str_file_ids = ['mat_matlab', 'mat_algebra_rules']
# select subset
ls_dc_selected = [dc_exa
                   for dc_exa in ls_dc_exa
                   if ((dc_exa['file'] in ls_str_file_ids)
                       and
                       (dc_exa['val']== 1))]

# print
pprint.pprint(ls_dc_selected, width=1)

```

```

## [{'date': datetime.date(2020, 5, 2),
##  'description': 'Frequency '
##                'table, '
##                'bar '
##                'chart '
##                'and '
##                'histogram',
##  'file': 'mat_matlab',
##  'title': 'One '
##          'Variable '
##          'Graphs '
##          'and '
##          'Tables',
##  'val': 1}]

```

1.7 Drop Element of Dictionary

Drop element of a dictionary inside a list:

```

# Dictionary
dc_test = [{"file": "mat_matlab_1",
            "title": "One Variable Graphs and Tables",
            "description": "Frequency table, bar chart and histogram",
            "val": 1,
            "date": datetime.date(2020, 5, 2)},
           {"file": "mat_matlab_2",
            "val": "mat_matlab_2"}]

```

```

# Drop
del dc_test[0]['val']
del dc_test[0]['file']
del dc_test[0]['description']
del dc_test[1]['val']

# Print
pprint.pprint(dc_test, width=1)

## [{'date': datetime.date(2020, 5, 2),
##   'title': 'One '
##         'Variable '
##         'Graphs '
##         'and '
##         'Tables'},
##  {'file': 'mat_matlab_2'}]

```