

Container Map Basics

back to [Fan's Intro Math for Econ](#), [Matlab Examples](#), or [MEconTools Repositories](#)

Generate a Container map with any Type of data

Create a container map with float, int, string, and matrix

```
close all;
clear all;

% Create A Map with String Keys and any values
param_map = containers.Map('KeyType','char','ValueType','any');
param_map('share_unbanked_j') = 12;
param_map('equi_r_j') = 2;
param_map('equi_w_j') = 'abc';
param_map('equi_P_j') = zeros(2,3);
disp(param_map.keys);
```

```
{'equi_P_j'}    {'equi_r_j'}    {'equi_w_j'}    {'share_unbanked_j'}
```

```
disp(param_map.values);
```

```
{2×3 double}    {[2]}    {'abc'}    {[12]}
```

Access Multiple Values of a container map

Values been accessed need to be of the same type

```
% Parameter Dealing from Map
params_group = values(param_map, {'share_unbanked_j', 'equi_r_j'});
[equi_P_j, equi_r_j] = params_group{:};
disp(['equi_P_j:' num2str(equi_P_j) ', equi_r_j:' num2str(equi_r_j)]);
```

```
equi_P_j:12, equi_r_j:2
```

```
% Access Scalar Elements of Map and Convert the Array
disp(cell2mat(values(param_map, {'share_unbanked_j', 'equi_r_j'})));
```

```
12    2
```

Create a container map of color values and generate a array of color choices:

```
% Container map with three colors
mp_colors = containers.Map('KeyType','char','ValueType','any');
mp_colors('blue') = [57 106 177]./255;
mp_colors('red') = [204 37 41]./255;
mp_colors('black') = [83 81 84]./255;
% An selection array
ar_st_colors_pick = {'blue', 'blue', 'red', 'black', 'blue'};
ar_colors = values(mp_colors, ar_st_colors_pick);
% Print selected colors
celldisp(ar_colors);
```

```

ar_colors{1} =
    0.2235    0.4157    0.6941

ar_colors{2} =
    0.2235    0.4157    0.6941

ar_colors{3} =
    0.8000    0.1451    0.1608

ar_colors{4} =
    0.3255    0.3176    0.3294

ar_colors{5} =
    0.2235    0.4157    0.6941

```

Use Stringed Numeric Array Array as Key for Container Map

The key for the container map below are char of an array, with string and array values associated with each key. Access values by converting array to string in the form of the keys.

```

% Define a Map with stringed array as keys
param_map = containers.Map('KeyType','char', 'ValueType','any');
ar_key1 = [11, 12, 21, 22, 31, 32];
ar_key2 = [10, 20, 30];
ar_key3 = [2];
ar_key4 = [1];
param_map(num2str(ar_key1)) = "nest1";
param_map(num2str(ar_key2)) = [1,3,4];
param_map(num2str(ar_key3)) = "nest3";
param_map(num2str(ar_key4)) = [2,3;4,5];

% Show keys and values
disp(param_map.keys);

```

```

{'1'}    {'10  20  30'}    {'11  12  21  22  31  32'}    {'2'}

```

```

disp(param_map.values);

```

```

{2x2 double}    {1x3 double}    {"nest1"}    {"nest3"}

```

```

% Access value via stringed arrays

```

```
disp(param_map(num2str(ar_key2)))
```

```
1      3      4
```

Container Integer Keys

Given some matrix, I want to store matrix column names as well as labels for what each row and column correspond to. Achieve this using a cell array of container maps. Cell dimensions correspond to the first, second, etc dimensions, any dimension specific information can be stored in this fashion.

Can access information associated with the label value of the row values:

```
% Define Matrix Row and Column and additional dimension information
cl_mp_datasetdesc = {};
cl_mp_datasetdesc{1} = containers.Map({'dim', 'name', 'labval'}, {1, 'kids', [0,1,2,3]});
cl_mp_datasetdesc{2} = containers.Map({'dim', 'name', 'labval'}, {2, 'age', [18,19,20]});
% get variable labels for the first dimension (rows)
disp([...
    string(['dim 1 var name:' cl_mp_datasetdesc{1}('name') ]), ...
    string(['dim 2 var name:' cl_mp_datasetdesc{2}('name') ])...
]);
```

```
"dim 1 var name:kids"      "dim 2 var name:age"
```

Is Key In Container

```
param_map_a = containers.Map('KeyType', 'char', 'ValueType', 'any');
param_map_a('fl_b_bd') = -3;
param_map_a('fl_w_max') = 50;
param_map_a('fl_kp_min') = 0;
param_map_a('it_w_i') = 100;

disp([...
    string(['has it_w_i as key? ' num2str(isKey(param_map_a, 'it_w_i'))]), ...
    string(['has it_w_i1 as key? ' num2str(isKey(param_map_a, 'it_w_i1'))]) ...
]);
```

```
"has it_w_i as key? 1"      "has it_w_i1 as key? 0"
```

Container Key Loop

Generate new container key within loop dynamically

```
param_map_a = containers.Map('KeyType', 'char', 'ValueType', 'any');
rng('default');
rng(123);
for st_cur = ["abc", "efg", "qqq"]

    if (strcmp(st_cur, "abc"))
        data = rand([1,1]);
    elseif (strcmp(st_cur, "efg"))
        data = 123.123;
    elseif (strcmp(st_cur, "qqq"))
```

```

    data = -123;
end

% common function
fl_sh_0p1pc_j = data*2 + 1;
fl_sh_5pc_j = data/2 - 1;

% generate map keys
st_key_sh_0p1pc_j = strjoin([st_cur, 'sh_0p1pc_j'], "_");
st_key_sh_5pc_j = strjoin([st_cur, 'sh_5pc_j'], "_");

% store
param_map_a(st_key_sh_0p1pc_j) = fl_sh_0p1pc_j;
param_map_a(st_key_sh_5pc_j) = fl_sh_5pc_j;

end

disp([...
    string(['param_map_a.keys:' param_map_a.keys]), ...
    string(['param_map_a.values:' string(param_map_a.values)]) ...
]);

```

Columns 1 through 7

```
"param_map_a.keys:"    "abc_sh_0p1pc_j"    "abc_sh_5pc_j"    "efg_sh_0p1pc_j"    "efg_sh_5pc_j"    "qqq_sh_0p1pc_j"
```

Columns 8 through 14

```
"param_map_a.values:"    "2.3929"    "-0.65177"    "247.246"    "60.5615"    "-245"    "-62.5"
```