Working with Cell Arrays

We have used a ***cell array*** to store a collection of strings

```
myPets = {'cleo'  'mona'  'tammy'};
```

We can access the content of an individual location of a cell array using an index placed ***inside curly braces:***

```
>> myPets{1}
ans =
cleo
```

```
for index = 1:length(myPets)
    disp(myPets{index});
end
```

The ***real power*** of cell arrays is that they allow us to store ***multiple types of data*** in one place:

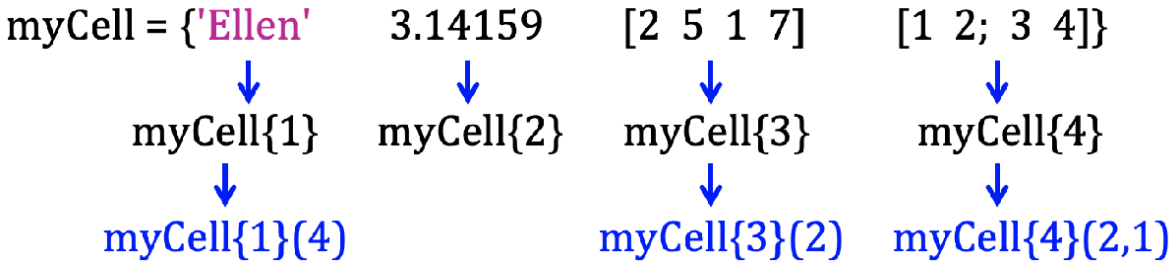
```
>> myCell = {'Ellen'  3.14159  [2  5  1  7]  [1  2;  3  4]}
myCell =
  1×4 cell array
    {'Ellen'}  {[3.1416]}  {1×4 double}  {2×2 double}
```

```
>> celldisp(myCell)
myCell{1} =
Ellen
myCell{2} =
    3.1416
myCell{3} =
    2   5   1   7
myCell{4} =
    1   2
    3   4
```

Create a cell array from scratch with the **cell** function:

```
>> newCell = cell(1,3);
>> newCell{1} = 'Stella';
>> newCell{2} = 'SCI  E122';
>> newCell{3} = stellaImage;
```

How do we refer to the content of vectors or matrices that are stored inside a cell array?



When working with data in ***external text files*** (stay tuned!), the data will automatically be stored in cell arrays when read into MATLAB, which may contain "inner" cell arrays. Our final example provides a taste of this idea.

Into Thin Air…

Let's explore the **mountainInfo.m** script…

```
% cell array of information about mountains
mountains = {{'Everest' 'K2' 'Kanchenjunga' 'Lhotse I' 'Makalu I' 'Lhotse II' ...
              'Dhaulagiri' 'Manaslu I' 'Cho Oyu' 'Nanga Parbat' 'Annapurna'} ...
             {'Himalayas' 'Karakoram' 'Himalayas' 'Himalayas' 'Himalayas' ...
            'Himalayas' 'Himalayas' 'Himalayas' 'Himalayas' 'Himalayas' 'Himalayas'} ...
             {'Nepal-China' 'Kashmir' 'Nepal-India' 'Nepal-China' 'Nepal-China' ...
              'Nepal-China' 'Nepal' 'Nepal' 'Nepal-China' 'Kashmir' 'Nepal'} ...
             [29028 28250 28208 27923 27824 27560 26810 26760 26750 …
              26660 26504]};

% ask the user to enter the name of a mountain
mount = input('Enter the name of a mountain: ',  's');

% determine the index of the user's mountain in the cell array of names
index = find(strcmpi(mountains{1}, mount));

% if the user's mountain is not found, print a message about this,
% otherwise print all the information about this mountain
if isempty(index)
   disp('your mountain was not found');
else
   disp(['name: ' mountains{1}{index}])
   disp(['range: ' mountains{2}{index}]);
   disp(['location: ' mountains{3}{index}]);
   disp(['height: ' num2str(mountains{4}(index)) ' feet']);
end
```