Cell Arrays

Cell arrays are similar to regular arrays in that they are "indexed" lists of data, with a symbolic name. Unlike traditional arrays, a cell array can contain a different data type in every "bucket". These buckets are called "cells". Generally, it is better to use the "structure" data type instead of cell arrays. One of the few times time cell arrays are useful (and in fact required) is when an array of strings is created.

Cell Arrays

Cell arrays are used when elements of differing types must be stored in a single array. There is usually **not** a good reason to do this. If you want multiple "types" of data associated with a single entity, you should use "structures".

Cell Array Syntax

Cell arrays in Matlab use the curly bracket {} notation instead of the normal parentheses (). While you may think that using () works, it in fact returns the "cell" of the array, not the "value" of the cell, which 99% of the time is not what you are looking for.

Here is an example of creating a cell array one "bucket" (or cell) at a time:

```
names{1} = 'jim';
names{2} = 'joe';
names{3} = 'jane';
names{4} = 'janet';  % notice the use of the curly brackets {}
```

Here is an example of creating the same cell array on a single line:

```
names = {'jim', 'joe', 'jane', 'janet' };
```

Warning: *using [ ] instead of { } will cause the following (sometimes useful) behavior, which is not what we want (at this point):*

```
names = ['jim', 'joe', 'jane', 'janet' ]
names =
      jimjoejanejanet

% it could be useful for the following

message = [ 'How ', 'are ', 'you ', 'today?' ]
message =
        How are you today?
```

Here is an example of different types in a single cell array:

```
mixed_values = {'jim', 89, [5 2 1] };

>> mixed_values{1}
ans =                     % a string
jim

>> mixed_values{2}
ans =                     % a number
  89

>> mixed_values{3}
ans =                     % an array of numbers
  [5 2 1]


% here we retrieve the regular array out of the
% cell array and then the first value from the regular array
>> mixed_values{3}(1)
ans =
  5

% here we print the first name in the array as a string

>> fprintf('The first student is named %s\n', mixed_values{1});
```

Cell arrays of Strings

Because of ancient history and the mechanics behind the way Matlab is written, it turns out that strings must be stored in cell arrays. This causes many problems with new students, so make a note, and commit it to memory!

PLEASE NOTE: *In Matlab, strings of different lengths are considered different types (even though we don't usually make this distinction) and thus must be stored in cell arrays. Thus 'Jim' is an array of 3 characters. 'Joan' is an array of 4 characters. These are considered different types! Thus Cell arrays must be used to contain strings, not regular arrays!*

Here are some more examples.

Remember, when using Cell arrays, you **must** use the {} brackets. Failure to do so will cause a variety of weird and hard to find logical errors in your program.

```
the_separate_parts_of_my_name = {'H.', 'James', 'de', 'St.', 'Germain' };

first_name = the_separate_parts_of_my_name{2};              % Notice the {}

first_letter_of_name = first_name(1);                       % Notice the ()

first_letter_of_name = the_separate_parts_of_my_name{1}(1); % Notice the {} ()
```

Back to Topics List