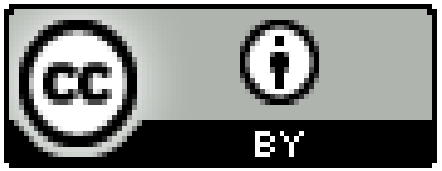
Matlab Programming

Structures and Cell Arrays



Matlab's basic data type is the double array.

Most data is numeric…

… but some is not.

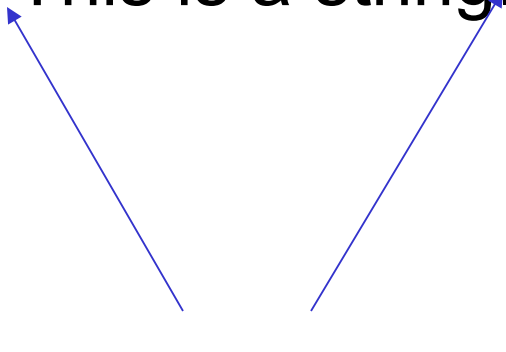
- Labels
- Textual Input
- …

Example: strings

```
>> str = ‘This is a string.’;
```

Example: strings

```
>> str = 'This is a string.';
```



Example: strings

```
>> str = ‘This is a string.’;
>> strArr = [‘string1’, ‘string2’]
strArr =
    string1string2
```

Example: strings

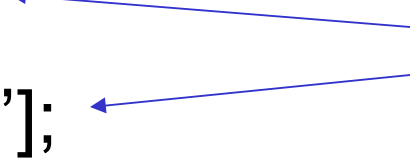
```
>> str = ‘This is a string.’;
>> strArr = [‘string1’, ‘string2’]
strArr =
    string1string2
>> strArr[2]
 ans =
    ‘t’
>> length(strArr)
    14
```

What is a string?

A string IS an array… of characters.

A string IS an array… of characters.

```
>> str = ‘Hi’;
>> str = [‘H’ ‘i’];
```

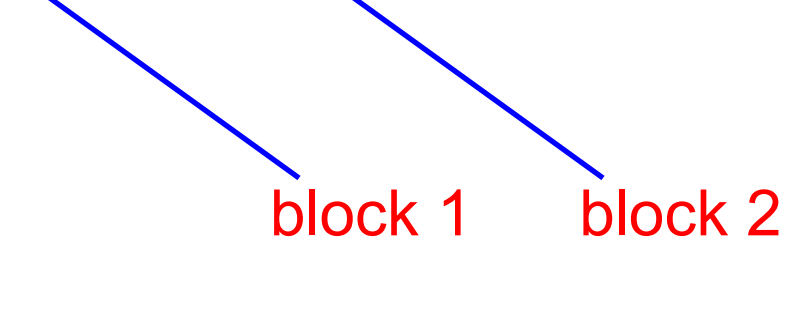


A string IS an array… of characters.

```
>> str = ‘Hi’;
>> str = [‘H’ ‘i’];
```

Arrays of strings behave like arrays:

```
>> strArr = [‘string1’, ‘string2’]
strArr =
    string1string2
```



That isn't really what we want…

That isn't really what we want…

In order to have an array of strings, we need to be able to have mixed types.

That isn't really what we want…

In order to have an array of strings, we need to be able to have mixed types.

Matlab arrays: an array of arrays is *flattened* into one big array.

Matlab cell arrays: an array of arrays *is* an array of arrays.

Two data types to handle data with mixed types:

Structure – store hierarchical data of mixed type.

Cell array – an array of containers that can contain mixed data types.

Two data types to handle non-numeric data:

Structure – store hierarchical data of mixed type.

Cell array – an array of containers that can contain mixed data types.

Major difference: cell arrays indexed by integer, structures indexed by arrays.

Colors

red

blue

green

black

orange

```
>> colors = struct();
>> colors.red = [255 0 0];
>> colors.blue = [0 0 255];
>> colors.green = [0 255 0];
>> colors.black = [255 255 255];
>> colors.orange = [255 112 0];

>> colors.orange[2]
ans =
    112
```

Structure as dictionary:

Python:

```
>>> colors = dict();
>>> colors['red'] = [255, 0, 0];
```

Matlab:

```
>> colors = struct();
>> colors.red = [255 0 0];
```

Dynamically creating key-value pairs:

```
>> s = struct();
>> a = ‘file’;
>> s.a = ‘data2_27_11.txt’;
>> s
s =
        a: ‘data_2_27_11.txt’
```

Dynamically creating key-value pairs:

```
>> s = struct();
>> a = ‘file’;
>> s.(a) = ‘data2_27_11.txt’;

>> s
s =
        file : ‘data’
```

Full set of methods to treat s as a ‘dictionary’:

```
isfield(s,’file’);  % 1 if ‘file’ is a field
fieldnames(s); % Return cell array of field names, or keys
orderfields(s); % Return ordered set of fields
rmfield(s,’file’); % Remove the field ‘file’.
```

But only strings can be field names.

Cell arrays:

Similar to floating point arrays, but…

each cell can hold a different type of data.

```
>> c = {1 'this is a string' 3 < 5};
```

Cell arrays:

Similar to floating point arrays, but…

each cell can hold a different type of data.

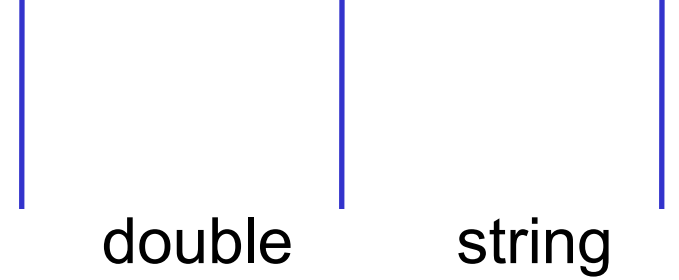
```
>> c = {1 'this is a string' 3 < 5};
```

Cell arrays:

Similar to floating point arrays, but…

each cell can hold a different type of data.

```
>> c = {1 'this is a string' 3 < 5};
```



double string Boolean

Indexing cell arrays

```
>> c(1)
ans =
        [1]
```

```
>> c{1}
ans =
     1
```

Why are the results different?

Indexing cell arrays

```
>> c(1)
ans =
        [1]
>> iscell(c(1))
ans =
      1
```

```
>> c{1}
 ans =
      1
>> iscell(c{1})
ans =
      0
```

Curly braces give the *content* of the cell.

Indexing cell arrays

```
>> c(1)
ans =
        [1]
>> iscell(c(1))
ans =
      1
```

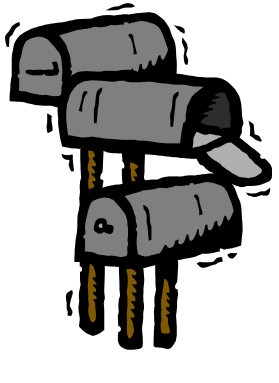
```
>> c{1}
 ans =
      1
>> iscell(c{1})
ans =
      0
```

Curly braces give the *content* of the cell.

Parentheses return a cell array with one element.

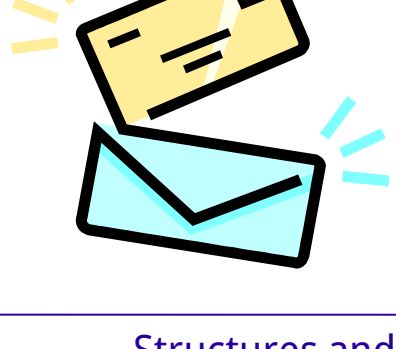
A useful analogy:

A cell array is an array of mailboxes…



A single cell is a single mailbox…

The contents of the cell can be a letter, a package, or some other type.



Helpful cell/struct functions:

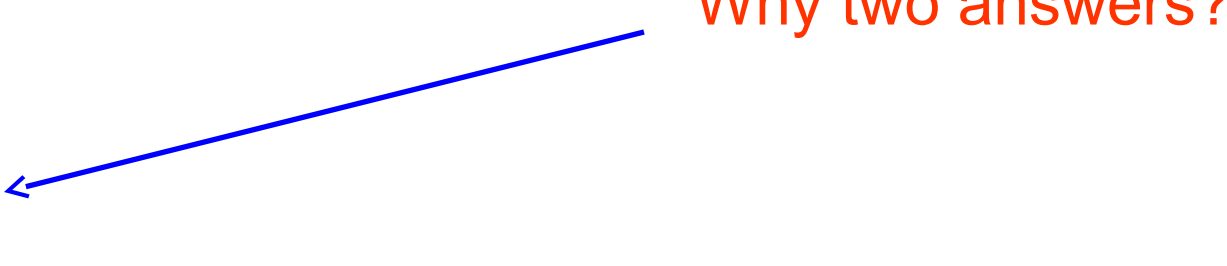
iscell

isstruct

char(str) % Convert a cell array of strings to an array of characters (i.e. a string)

Cell arrays can have more than one dimension:

```
>> c = { ‘a’ , ‘b’, ‘c’ ; ‘d’, ‘e’, ‘f’};
>> c{2,2}
ans =
e
>> c{:,1}
ans =
a
ans =
d
```



What's worse:

```
>> c = { 'a' , 'b', 'c' ; 'd', 'e', 'f'};
>> col = c{:,1};
>> col
ans =
a
```

Where is 'd'?

```
>> col = c(:,1);
>> col
    ['a']
    ['d']
```

Two element cell array

Dictionaries in Matlab

A struct is a dictionary…

… if you can live with strings as keys

A cell array is *not* a dictionary…

… since it fills out a grid with empty arrays

A cell array is more like a Python *list*

created by

Richard T. Guy

June 2011