

Computing Fundamentals

Structs and Cell Arrays

Salvatore Filippone

`salvatore.filippone@uniroma2.it`

2014–2015

Arrays are collections of values with a *uniform* type; structures and cell arrays can hold *heterogeneous* data.

- 1 Cell arrays are accessed with *indices*;
- 2 Structures are accessed with *names*.



Cell Arrays

- A cell array is an array, so it can be indexed numerically;
- Each element of a cell array is a “container”;
- Each container holds arbitrary data.

```
>> A{1} = 42  
A = [42]
```

```
>> B(1) = {[4 6]};  
B = [1x2 double]
```

```
>> C = {3, [1,2,3], 'abcde'}  
C = [3] [1x3 double] 'abcde'
```

```
>> D = [A B C {'xyz'}]  
D = [42] [1x2 double] [3] [1x3 double] 'abcde' 'xyz'
```



Access with parentheses gives the *container*

```
>> E = D(2) % parentheses – a container  
E = [4 6]
```

Access with braces gives the *contents*

```
>> D{2} % braces – the contents  
ans = 4 6
```



You can use them:

- As lists of values in a case inside a switch

```
switch y
  case 1
    disp(" One" );
  case num2cell(2:4)
    disp(" Between 2 and 4" );
  otherwise
    disp(" Something else" );
end
```

- As argument lists for functions

```
B={1,2,3,4};
[x1,x2]=quadratic(B{1:3})
```



Processing cell arrays:

```
< Have a cell array , e.g. V={1,2,3,4...} >  
for k=1:length(V)  
    <extract an element item=V{k}>  
    <check its type>  
    <process according to type>  
end
```

Type checking options:

- Use a switch on `class(item)`
- Use `iscell`, `islogical`, `ischar`, `isnumeric`
- Use `isa(item,<classname>);`

Structures are collections of (heterogeneous) entities whose individual fields are indexed by *name*.

- Fields may contain any Octave/Matlab object;
- Field names have the same rules as variable names;
- Since they are indexed by name, fields in a structure are usually not too many

Building and accessing structures:

```
octave:72> a.nome="Giovanni";  
octave:73> a.cognome="Paisiello";  
octave:74> a.matricola="1234567";  
octave:75> a  
a =
```

scalar structure containing the fields:

```
nome = Giovanni  
cognome = Paisiello  
matricola = 1234567  
octave:76> a.matricola  
ans = 1234567  
octave:77> strcmp(a.cognome,"Verdi")  
ans = 0
```


you can create arrays of structures, either directly or using the `struct` function. Manipulating field names:

```
octave:79> fieldnames(a)
ans =
{
  [1,1] = nome
  [2,1] = cognome
  [3,1] = matricola
}
octave:80> names=fieldnames(a);
octave:81> names{1}
ans = nome
```

“Indirect” name access:

```
octave:82> a.(names{2})
ans = Paisiello
```

It is best to use “constructor” functions

```
function ans=student(name,surname,id)
    ans.name=name;
    ans.surname=surname;
    ans.id=id,
end
```

Advantages:

- 1 Less risk of misspelling;
- 2 Consistent treatment of multiple items;
- 3 Easy reorganization if needed;