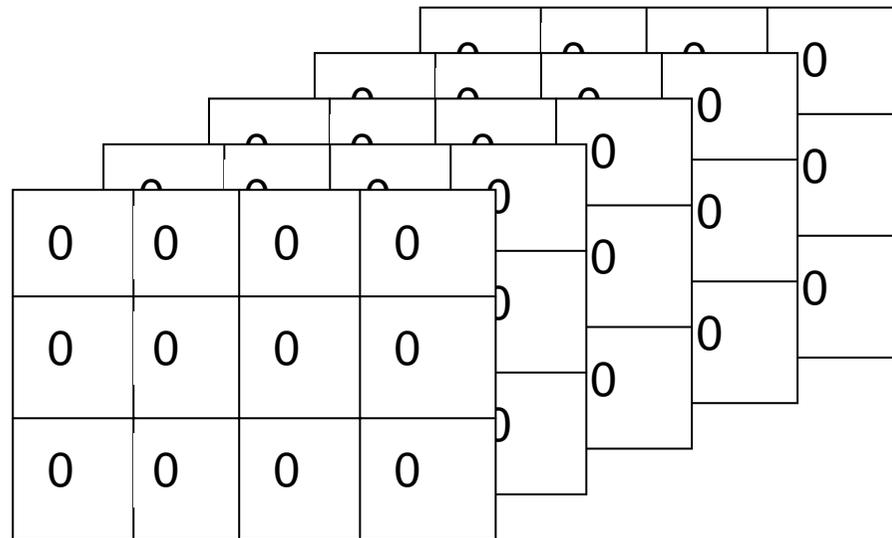


# Arrays multidimensionali:

Sono arrays con un numero di indici >2:

```
Z3_D = zeros(3,4,5)    % Z3_D è una matrice tri-dimensionale, 3x4x5  
                        % i cui elementi valgono 0
```

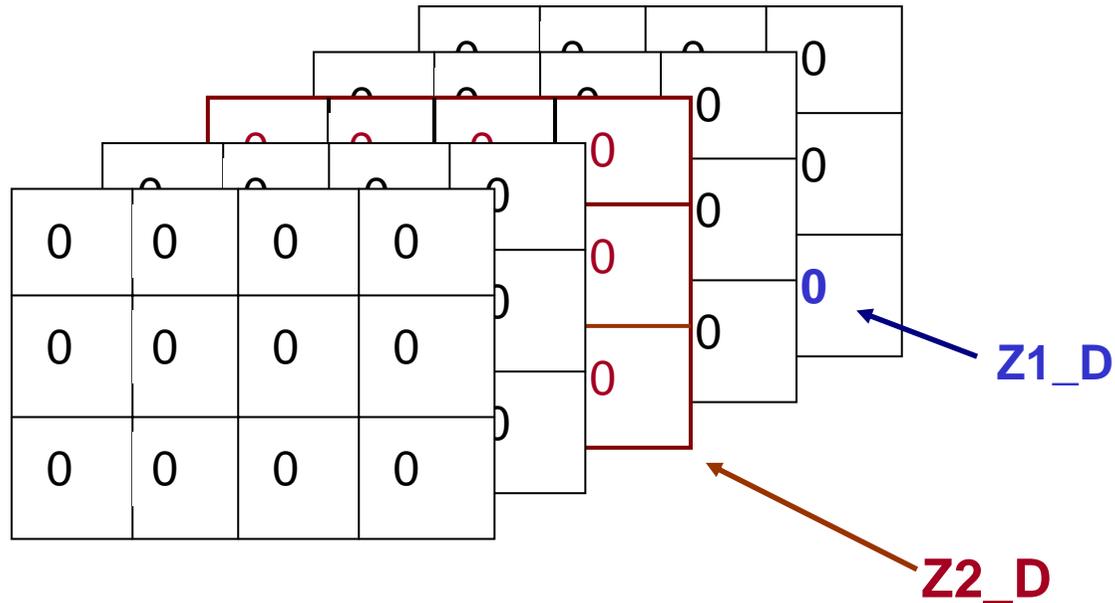


# Array tri-tidimensionale

`Z3_D = zeros(3,4,5)`    % **Z3\_D** è una matrice tri-dimensionale, 3x4x5  
                                  % i cui elementi valgono 0

`Z2_D = Z3_D(:,:,3)`    % **Z2\_D** è una matrice bi-dimensionale, 3x4,  
                                  % ottenuta da **Z3\_D**

`Z1_D = Z3_D( end,end,end)` % **Z1\_D** è uno scalare ottenuto da **Z3\_D**



# Cell array:

- sono arrays i cui elementi sono copie di altri arrays
- un cell-array è creato mettendo insieme (tra **parentesi graffe**) più variabili:

`A = [1 2 3; 4 5 6; 7 8 9]`

`C = { A sum(A) prod(prod(A)) }`

matrice 2D

vettore

scalare

C è un cell-array di 1x3 elementi, i 3 elementi sono di dimensioni diverse:



I cell-arrays possono essere utilizzati per memorizzare sequenze di matrici di dimensioni diverse

- si può creare un cell-array vuoto con la funzione `cell`

# Cell array:

**Come indirizzare elementi di un cell array:**

**A1 = C{1}    % ottengo una matrice 3x3, il primo elemento di C**

**P = C{3}    % ottengo uno scalare, il terzo elemento di C**

# Strutture

Le strutture sono array con campi che possono contenere tipi di dato differenti.

I campi di una struttura vengono specificati utilizzando un punto seguito dal nome del campo

Omar



Data\_nascita

Miglior\_classifica

Partite\_W\_L

Titoli\_Vinti

```
>> Omar.Data_Nascita='8 Maggio 1968'
```

```
Omar =
```

```
    Data_Nascita: '8 Maggio 1968'
```

```
>> Omar.Miglior_Classifica=18
```

```
Omar =
```

```
    Data_Nascita: '8 Maggio 1968'
```

```
    Miglior_Classifica: 18
```

**I dati possono essere scalari, array, celle, altre strutture**

**Omar =**

**Data\_Nascita: '8 Maggio 1968'**

**Miglior\_Classifica: 18**

**Partite\_W\_L: [150 157]**

**Titoli\_Vinti: {'Milano' 'Rotterdam'}**

**Le strutture possono essere create in modo dinamico in fase di assegnazione**

**>> insegnamento.denominazione='SASB'**

**insegnamento =**

**denominazione: 'SASB'**

**E' possibile avere strutture a più dimensioni**

```
>> insegnamento(2).denominazione='MASB'
```

```
insegnamento =
```

```
1x2 struct array with fields:  
denominazione
```

**Se aggiungiamo un campo, questo è aggiunto a tutti gli elementi**

```
>> insegnamento(2).num_stud=130
```

```
insegnamento =
```

```
1x2 struct array with fields:
```

```
denominazione  
num_stud
```

```
>> insegnamento(1)
```

```
ans =
```

```
denominazione: 'SASB'  
num_stud: []
```

**Si accede alle strutture come alle matrici**

```
>> insegnamento(1).denominazione
```

```
ans =
```

```
SASB
```

```
>> insegnamento(1).denominazione(1:2)
```

```
ans =
```

```
SA
```

Si possono creare strutture con il comando *struct* utilizzando cell array (delle stesse dimensioni) o scalari

```
>> categorie={'arte','viaggi','romanzi','cucina'}
```

```
categorie =
```

```
 'arte' 'viaggi' 'romanzi' 'cucina'
```

```
>> quantita={{5},{4},{12},{3}}
```

```
quantita =
```

```
 {1x1 cell} {1x1 cell} {1x1 cell} {1x1 cell}
```

```
>> libreria=struct('categorie',categorie,'quantita',quantita)
```

```
libreria =
```

```
1x4 struct array with fields:
```

```
 categorie
```

```
 quantita
```

<b>rmfield</b>	<b>rimozione campo</b>
<b>setfield</b>	<b>sostituzione contenuto di un campo</b>
<b>struct2cell</b>	<b>conversione da struttura a cella</b>
<b>fieldnames</b>	<b>fornisce i nomi della struttura</b>
<b>isstruct</b> <b>(0) altrimenti</b>	<b>fornisce vero (1) se applicato ad una struttura, falso</b>