Structures

Structures

- Group properties of objects into a data structure.

```
>>  laptop=struct('brand','apple', 'cpu',3.3, 'year',2012, 'price',1499)

laptop =
    brand: 'apple'
     year: 2012
      cpu: 3.3000
    price: 2000
```

brand:	apple
cpu:	3.3
year:	2012
price:	1499

Dot operator

```
>> laptop.brand
apple

>> laptop.brand = 'sony'
laptop =
    brand: 'sony'
      cpu: 3.3000
     year: 2012
    price: 1499
```

```
>> dynfield = 'price';

>> laptop.(dynfield)

>> laptop.(dynfield) = 1999
```

```
>> laptop.country
??? Reference to non-existent field 'country'.
```

functions for structures

```
>> fieldnames( laptop )
ans =
    'brand'
    'cpu'
    'year'
    'price'
```

```
>> isstruct(laptop)
   1
>> isfield( laptop, 'brand' )
   1
>> isfield( laptop, 'user' )
    0
```

```
>> getfield( laptop , 'brand' )

>> setfield( laptop, 'brand','sony')

>> rmfield( laptop, 'cpu')

>> laptop = rmfield(laptop, 'cpu')
```

Vectors of structures

```
>> points = struct( 'x', 0, 'y', 0);
>> points(1) = struct( 'x', 0, 'y', 0);
```

packages

	item_no	cost	price	code
1	123	19.99	39.95	g
2	456	5.99	49.99	l
3	587	11.11	33.33	w

```
>> points(2) = struct( 'x', 1, 'y', 0);
>> points(3) = struct( 'x', 6, 'y', 5);

>> points(2).y = 4;
>> points(2) = [ ];
```

- When structures are extended, uninitialized entries are empty.

```
>> points(100) = struct( 'x', 6, 'y', 5);
```

Creating Multiple Structures

- points = struct('x', {1, 3, 5} , 'y' , {10, 20, 30});
- points = struct('x', {1, 3, 5} , 'y' , {10, 20, 30}, 'z', 5);
- points = struct('x', {1, 3, 5} , 'y' , {10, 20, 30}, 'z', {5});

- points = struct('x', {1, 3, 5} , 'y' , {{10,20,30}});

- points = struct('x', [1, 3, 5] , 'y' , [10 20 30]);

- points = struct('x', {} , 'y' , {});

Fields of structure arrays

```
>> points = struct( 'x', 0, 'y', 0);
>> points(2) = struct( 'x', 1, 'y', 4);

>> points(1).x

>> points.x

>> [ points.x ]

>> { points.x }
```

Fields of vector of structures

```
>>  laptops = struct('brand','apple', 'cpu',3.3, 'year',2012, 'price',1499)
>> laptops(2).brand = 'sony';
>> laptops(2).cpu = 3.8;

>> [ laptops.cpu ]

>> { laptops.cpu }

>> [ laptops.year]

>> { laptops.year }
```

Exercise

- Add up all the x axis of the points.

```
s = 0;
for i=1:numel(points)
  s = s + points(i).x;
end
```

- Alternative:

```
s = sum ( [ points.x ] )
```

Structures can contain any data type.

```
>> laptops(1).price = [1499 1599 1999];
>> laptops(1).cpu = struct('brand','intel', 'speed', 3.3);
```

- It is good practice to store the same data type in each field of a structure vector.
  - E.g., the following is allowed, but not recommended:

```
>> laptops(1).brand = {'apple', 'macbookpro'}
>> laptops(2).brand = 'sony';
```

The following is better:

```
>> laptops(2).brand = {'sony', ''} ;
```

Nested structures

```
>> lineseg = struct( ...
'endpoint1', struct('x',2,'y',4) ...
, 'endpoint2',struct('x',1,'y',6) ...
)

>> point1 = struct('x',2,'y',4) ;
>> point2 = struct('x',1,'y',6) ;
>> lineseg = struct( ...
'endpoint1', point1, 'endpoint2', point2 );

>> lineseg.endpoint1.x =3

>> lineseg.endpoint1
ans =
  x: 3
  y: 4

>> point1.x = 5
>> lineseg.endpoint1
```

lineseg

endpoint1		endpoint2	
x	y	x	y
2	4	1	6