



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΧΗΜΙΚΩΝ ΜΗΧΑΝΙΚΩΝ  
ΥΠΟΛΟΓΙΣΤΙΚΟ ΚΕΝΤΡΟ

# Προγραμματισμός και Χρήση Ηλεκτρονικών Υπολογιστών - Βασικά Εργαλεία Λογισμικού

## Μάθημα 8ο

Ανώνυμες συναρτήσεις (anonymous functions)

Συγγραφή συναρτήσεων, ορίσματα, συναρτήσεις ως ορίσματα συναρτήσεων

Πεδίο μεταβλητών (variable scope)

# Ανώνυμες συναρτήσεις (Anonymous functions)

Συναρτήσεις που ορίζονται σε μία εντολή

---

Παράδειγμα

Ορισμός της συνάρτησης  $f(x) = x^2 + 5$

```
>> fun_handle = @(x) x^2 + 5
```

το όνομα της μεταβλητής στην οποία θα αποθηκευτεί η συνάρτηση.  
Αυτές οι μεταβλητές ονομάζονται **function handles**

```
>> fun_handle(10)
```

ans =

105

όρισμα (argument)

# Άσκηση 1

---

Σχεδιάστε τη γραφική παράσταση της συνάρτησης  $f(x) = x^2 + 5$  για  $x=[-10\ 10]$   
Χρησιμοποιώντας τις συναρτήσεις:

- 1) `ezplot`
- 2) `plot`
- 3) `fplot`

```
>> ezplot('x^2+5', [-10 10])
```

```
>> plot(x, y)
```

```
>> fplot(fun_handle, [-10 10])
```

## Άσκηση 2

---

Σχεδιάστε τη γραφική παράσταση της συνάρτησης  $f(x) = \sin(x)$  για  $x=[0 \ 2\pi]$  χρησιμοποιώντας τη συνάρτηση `fplot`:

```
>> fplot(fun_handle, [0 2*pi])
```

ή

```
>> fplot(@sin, [0 2*pi])
```

# Συναρτήσεις (functions)

---

Τα **αρχεία συναρτήσεων** είναι τα m-files που ξεκινούν με τη λέξη **function**. Σε αντίθεση με τα αρχεία script, η εκτέλεση τους απαιτεί την **είσοδο μεταβλητών** (ορίσματα της συνάρτησης) και επιστρέφουν αποτελέσματα στις **μεταβλητές εξόδου** σε αντιστοιχία με τα functions σε γλώσσες προγραμματισμού όπως η Fortran ή C.

# Γενική μορφή μιας συνάρτησης του MATLAB

---

```
function [outvars] = function_name(inputvars)
    % comments
    % comments
        εντολές → τιμές
        outvars = τιμές
end
```

outvars: τα ονόματα των μεταβλητών εξόδου χωρισμένα με κόμμα , ή κενό

function\_name: όνομα της συνάρτησης. **Πρέπει να είναι το ίδιο με το όνομα του αρχείου.** Δηλαδή, το m-file πρέπει να αποθηκευτεί ως `function_name.m`

inputvars: τα **ορίσματα (arguments)** της συνάρτησης. Είναι τα ονόματα των μεταβλητών εισόδου χωρισμένα με κόμμα

comments: σχόλια σχετικά με τους υπολογισμούς και τα αποτελέσματα του function. Τα σχόλια τυπώνονται με την εντολή `help function_name`

εντολές: εντολές που υπολογίζουν τις τιμές που θα εκχωρηθούν στις μεταβλητές outvars.

## Παράδειγμα (1/2)

---

Να φτιάξετε μια συνάρτηση για τον υπολογισμό της:

$$f(x) = \begin{cases} \frac{5}{x+2} & \text{για } x \geq 1 \\ \sqrt{|x|+1} & \text{για } x < 1 \end{cases}$$

1. Επιλογή του ονόματος της συνάρτησης: `my_first_function`  
Το ίδιο όνομα θα πρέπει να έχει και το αρχείο στο οποίο θα αποθηκεύσουμε τη συνάρτηση.
2. Μεταβλητές εισόδου (ορίσματα): `x`
3. Μεταβλητές εξόδου: `y`

## Παράδειγμα (2/2)

---

```
>> edit my_first_function.m
```

```
function [y] = my_first_function(x)

if(x>=1)
    y=5/(x+2);
end
if(x<1)
    y=sqrt(abs(x)+1);
end

end
```

```
>> my_first_function(5)
```

```
ans =
```

```
0.7143
```



## Άσκηση 3

---

1. Να φτιάξετε μια **ανώνυμη συνάρτηση** για τη συνάρτηση  $f(x)=x^2 + 2x -1$
2. Φτιάξτε τη γραφική παράσταση της  $f$  χρησιμοποιώντας τη συνάρτηση `fplot`
3. Προσεγγίστε γραφικά τις ρίζες της εξίσωσης  $x^2 + 2x - 1 = 0$   
(εντολή `grid on` μετά την `fplot`)
4. Να φτιάξετε `script` που θα υλοποιεί τα ερωτήματα 1, 2 και 3

## Άσκηση 4

---

1. Να φτιάξετε μια **ανώνυμη συνάρτηση** για τη συνάρτηση  $f(x)=ax^2 + bx + c$ .
2. Προσεγγίστε γραφικά τις ρίζες της εξίσωσης  $ax^2 + bx + c = 0$ .  
Δίνονται:  $a=1$   $b=2$   $c=-1$
3. Να φτιάξετε script που θα υλοποιεί τα ερωτήματα 1 και 2

## Άσκηση 5

---

1. Να φτιάξετε μια **συνάρτηση σε αρχείο** με όνομα `myFunction`, όρισμα `x` και να επιστρέφει την τιμή της  $f(x) = x^2 + 2x - 1$
2. Προσεγγίστε γραφικά τις ρίζες της εξίσωσης  $x^2 + 2x - 1 = 0$  χρησιμοποιώντας τις συναρτήσεις `myFunction` `fplot` και `grid`
3. Να φτιάξετε αρχείο `script` που θα υλοποιεί το ερώτημα 2

## Άσκηση 6

---

1. Να φτιάξετε μια **συνάρτηση σε αρχείο** με όνομα `myFunction`, με ορίσματα `a`, `b`, `c`, `x` και να επιστρέφει την τιμή της  $f(x) = ax^2 + bx + c$
2. Προσεγγίστε γραφικά τις ρίζες της εξίσωσης  $ax^2 + bx + c = 0$  χρησιμοποιώντας τις συναρτήσεις `myFunction` `fplot` και `grid`.  
Δίνονται: `a=1` `b=2` `c=-1`
3. Να φτιάξετε αρχείο `script` που θα υλοποιεί το ερώτημα 2

## Συναρτήσεις που επιστρέφουν περισσότερες από μια τιμές

---

```
function [a c] = areacirc(rad)
% Use:[a c] = areacirc(rad)
% The function areacirc returns the area (a)
% and the circumference (c) of a circle
% Input arguments: the radius (rad) of the circle
    a = pi*rad.^2;
    c = 2*pi*rad;
end
```

```
>> areacirc(4)
ans =
    50.2655
```

```
>> [~, y]=areacirc(4)
y =
    25.1327
```

```
>> x=areacirc(4)
x =
    50.2655
```

```
>> r=1:4;
>> [x y]=areacirc(r)
x =
    3.1416    12.5664    28.2743    50.2655
```

```
>> [x y]=areacirc(4)
x =
    50.2655
y =
    25.1327
```

```
y =
    6.2832    12.5664    18.8496    25.1327
```

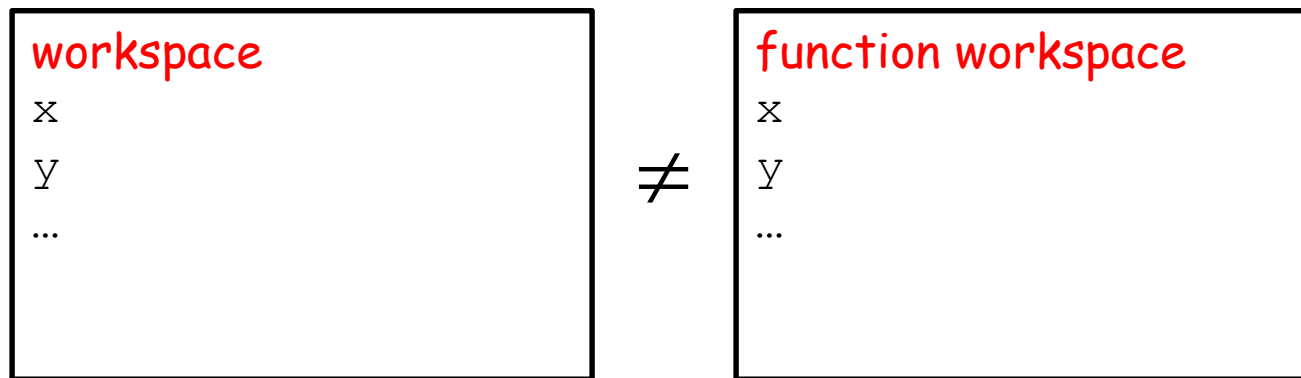
## Χώρος αποθήκευσης μεταβλητών συνάρτησης (1/2)

---

Κάθε συνάρτηση ορίζει το δικό της χώρο αποθήκευσης (**function workspace**) των μεταβλητών της.

**workspace**  $\neq$  **function workspace**

Το **function workspace** "καταστρέφεται" μετά το τέλος της συνάρτησης.



## Χώρος αποθήκευσης μεταβλητών συνάρτησης (2/2)

---

**Υποσυνάρτηση:** συνάρτηση που ορίζεται μέσα στην **κύρια συνάρτηση**

Κάθε υποσυνάρτηση έχει διαφορετικό χώρο αποθήκευσης από την κύρια συνάρτηση αλλά μπορεί να χρησιμοποιήσει και να αλλάξει τις τιμές των μεταβλητών που έχουν οριστεί στην κύρια συνάρτηση.

```
function mainFunction()  
    x=4;  
    fprintf('Value of x before calling subFunction: %d \n',x)  
    subFunction()  
    fprintf('Value of x after calling subFunction: %d \n',x)  
    function subFunction()  
        x=3;  
        fprintf('Value of x at subFunction: %d \n',x)  
    end  
end
```