

# מבוא לתכנות מתמטי

תרגול 3

# מנהלות

- יש להגיש את השיעורי בית דרך האתר:
- <https://submit.cs.biu.ac.il/cgi-bin/welcome.cgi>
- יש להיכנס עם השם משתמש וסיסמא שקיבלתם מהחובשים

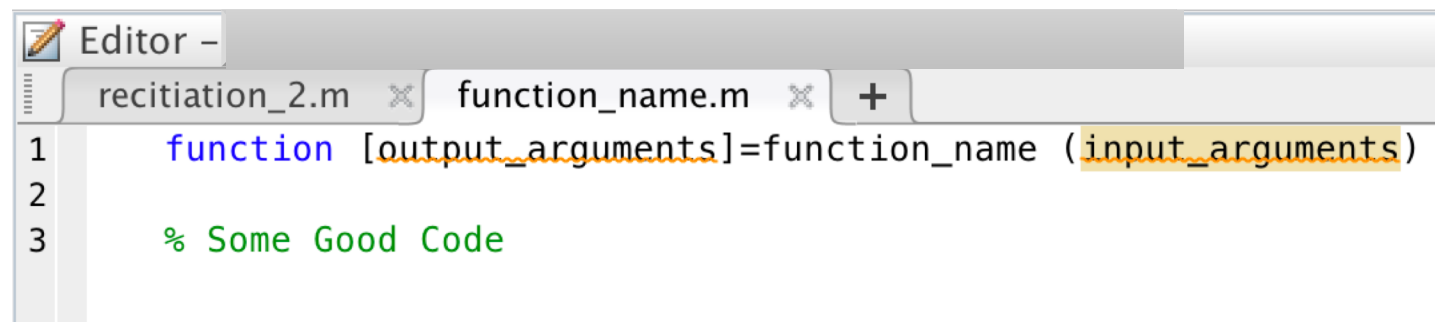
# פונקציות



# פונקציות

- מחלק את המשימה למשימות קטנות ועצמיות
- מונע שכפול קוד
- קל יותר לקריאה
- קל יותר לדבג (למצוא שגיאות בקוד)
- שימוש בקוד חוזר
- אל תשכחו את הפונקציות הבנויות של מטלאב!

# פונקציות



The image shows a screenshot of a MATLAB editor window. The window title is "Editor -". There are two tabs open: "recitation\_2.m" and "function\_name.m". The "function\_name.m" tab is active. The code in the editor is as follows:

```
1 function [output_arguments]=function_name (input_arguments)
2
3 % Some Good Code
```

```
% one return value -----  
% function sum = func_4(x, y)  
% sum = x + y;  
% end
```

```
% more than one -----  
% function [sum ,mul] = func_4(x, y)  
% sum = x + y;  
% mul = x*y;  
% end
```

# תרגיל

כתבו פונקציה שמקבלת  $x, y$  ובודקת האם

$$y = x^2 + x - 3$$

# פתרון

```
function answer = is_valid_point(x_val, y_val)
    if x_val^2 + x_val - 3 == y_val
        answer = true;
    else
        answer = false;
    end
end
```

```
>> is_valid_point(1,-1)
```

```
ans =
```

הקריאה לפונקציה

logical

1



# תרגיל

- כתבו פונקציה המקבלת וקטור של נקודת  $(x,y)$  ומחשבת כמה נקודות מקיימות

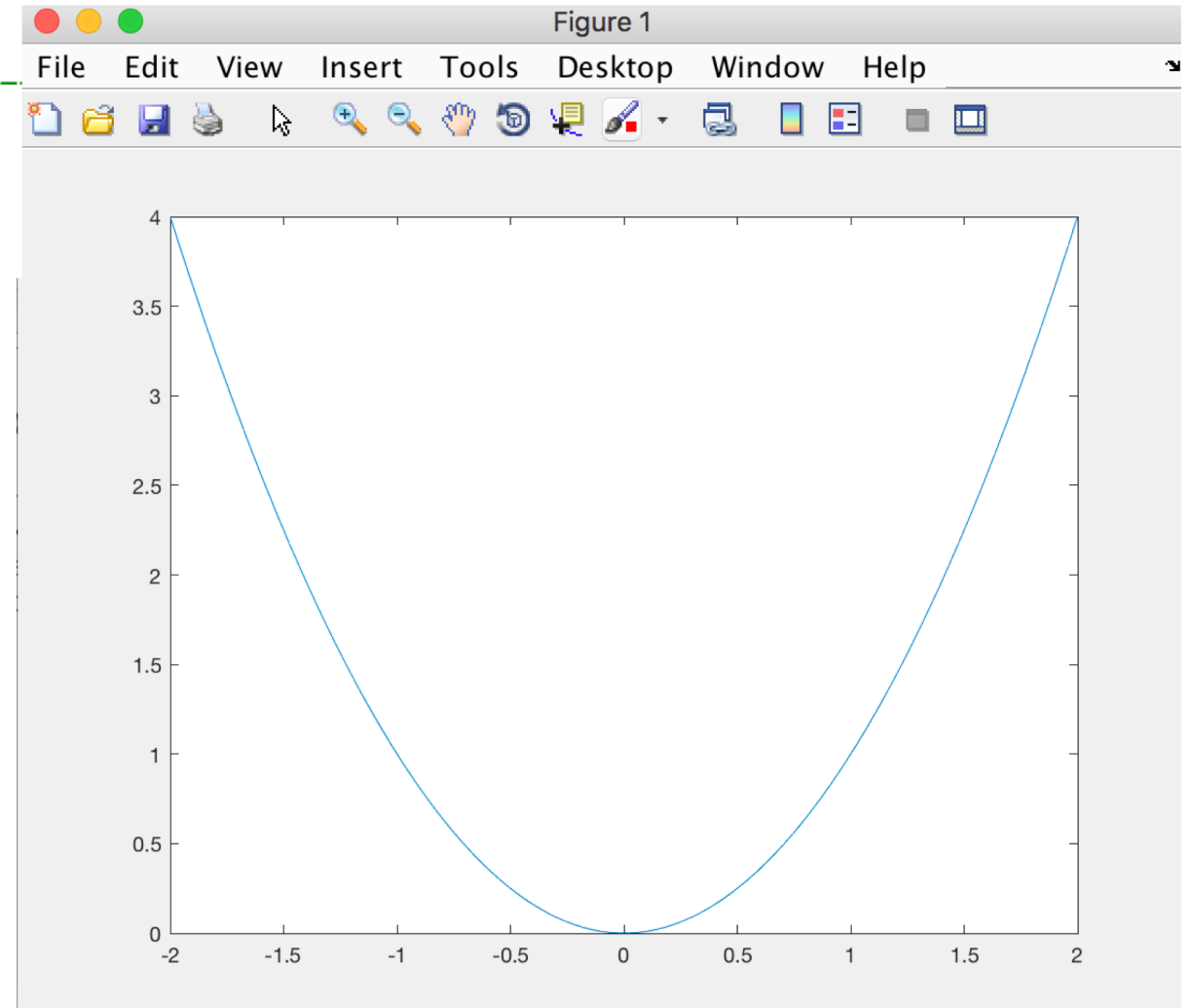
$$y = x^2 + x - 3$$

# פתרון

```
function sum = func_4(x_axis, y_axis)
sum = 0;
for i = 1:length(x_axis)
    if is_valid_point(x_axis(i), y_axis(i))
        sum = sum + 1;
    end
end
end|
```

# Anonymous function

```
% anonymous function -----  
% func_name =@ (input args) mathematical expression;  
sum_1 =@ (x, y) x + y;  
  
% plot y=x^2 -----  
power2 =@ (x) x.^2;  
x_axis = linspace(-2,2,100);  
plot(x_axis, power2(x_axis))
```



# מיון בועות

```
% ----- buble sort -----  
% write a function that given an array a, returns sorted a  
% -----  
% function sorted = func_5(array)  
%     sorted = array;  
%     len = length(array);  
%     stop = len - 1;  
%     swapped = true;  
%     while swapped  
%         swapped = false;  
%         for i = 1:stop  
%             if sorted(i) > sorted(i+1)  
%                 temp = sorted(i);  
%                 sorted(i) = sorted(i+1);  
%                 sorted(i+1) = temp;  
%                 % rais flag for swapped  
%                 swapped = true;  
%             end  
%         end  
%         stop = stop - 1;  
%     end  
% end
```

2 3 4 5 1

unsorted

2 3 4 5 1

$2 < 3$ , ok

2 3 4 5 1

$3 < 4$ , ok

2 3 4 5 1

$4 < 5$ , ok

2 3 4 5 1

$5 > 1$ , swap

2 3 4 1 5

$2 < 3$ , ok

2 3 4 1 5

$3 < 4$ , ok

2 3 4 1 5

$4 > 1$ , swap

2 3 1 4 5

$2 < 3$ , ok

2 3 1 4 5

$3 > 1$ , swap

2 1 3 4 5

$2 > 1$ , swap

1 2 3 4 5

sorted

# יעילות

- נרצה בעצם לדבר על (סדר) של הפעולות שנשתמש ובזיכרון שנשתמש.
- תלוי בקלט.
- נסתכל על המקרה הגרוע
- מיון בועות?
- חיפוש בינארי?

# תרגיל

- כתבו פונקציה שמקבלת מערך **ממוין** ומספר ומחזירה שני איברים מהמערך שסכומם המספר הנ"ל.
- הציעו פתרון ב $O(n)$

# פתרון - הרעיון

```
def n(arr,num)
```

```
    a=1
```

```
    b=len(arr)
```

```
    while(a<b):
```

```
        if arr(a)+arr(b)=n
```

```
            Return a,b
```

```
        else if arr(a)+arr(b)<num
```

```
            a=a+1
```

```
        else if arr(a)+arr(b)>num
```

```
            b=b-1
```

```
    Return false
```



# פתרון במטלעב

```
function [num1,num2]= two_sum(arr,sum_val)
    a=1;
    b=length(arr);
    num1=-1;
    num2=-1;
    while a<b
        if arr(a)+arr(b) == sum_val
            num1=a;
            num2=b;
            break
        elseif arr(a)+arr(b) <sum_val
            a=a+1;
        else
            b=b-1;
        end
    end
end
```

# תרגיל

- כתבו פונקציה שמקבלת מערך **ממוין** ומספר ומחזירה שני איברים מהמערך שסכומם המספר הנ"ל.
- הציעו פתרון ב  $O(n)$

והאם המערך לא היה ממוין?

