Subfunctions in Matlab

✓ الدوال الفرعية (Subfunctions) فيMATLAB

الدوال الفرعية هي دوال تُعرَّف داخل ملفات MATLAB (.m) وتُستخدم لتنظيم الكود وتقسيمه إلى أجزاء صغيرة يمكن إعادة استخدامها. الدالة الفرعية تكون خاصة بالملف الذي توجد فيه ولا يمكن استدعاؤها مباشرةً من ملفات أخرى.

✓ مفهوم الدوال الفرعية

- ✓ **تعريف** :دوال تُكتب داخل ملف الدالة الرئيسية وتكون مرئية فقط داخل ذلك الملف.
- ✓ **الغرض:**
  - تقسيم الكود إلى أجزاء منطقية.
  - إعادة استخدام الكود داخل الملف دون تعريضه للوصول الخارجي.
  - تحسين القراءة والصيانة.
- ✓ **تعريف الدوال الفرعية**

الدالة الفرعية هي دالة تُكتب في نفس الملف الذي يحتوي على دالة رئيسية.

عند تشغيل الملف، يتم تنفيذ الدالة الرئيسية، وتستدعي الدوال الفرعية عند الحاجة.

- ✓ **الفرق بين الدوال الرئيسية والدوال الفرعية**
- ✓ **الدالة الرئيسية:**
  - اسمها يكون مطابقًا لاسم الملف.
  - يمكن استدعاؤها من ملفات MATLAB الأخرى.
- ✓ **الدالة الفرعية:**

- لا يمكن استدعاؤها إلا من داخل نفس الملف.
- تُستخدم لتقسيم الكود إلى أجزاء أصغر وأكثر تنظيمًا.

✓ **كيفية كتابة دالة فرعية**

❖ **القواعد:**

- تُكتب الدالة الفرعية بعد نهاية الدالة الرئيسية في نفس الملف.
- تبدأ بالكلمة المفتاحية function متبوعة بتعريف الدالة.

✓ الصيغة العامة:

```
function output = MainFunction(input)
        % كود الدالة الرئيسية
        output = SubFunction(input); % استدعاء الدالة الفرعية
end
function result = SubFunction(data)
        % كود الدالة الفرعية
        result = data * 2; % مثال: مضاعفة القيمة
end
```

Example 1: Using a subfunction to calculate the sum and square of numbers

- function main()

  numbers = [1, 2, 3, 4, 5];

  total = computeSum(numbers); % استدعاء الدالة الفرعية

  squares = computeSquares(numbers); % استدعاء دالة أخرى

  fprintf('المجموع: %d\n', total);

  fprintf('المربعات: ');

  disp(squares);

- end
  - ❖ function sumResult = computeSum(array)

    % دالة لحساب مجموع العناصر

    sumResult = sum(array);
  - ❖ end
  - ✓ function squareResult = computeSquares(array)

    % دالة لحساب مربع كل عنصر

    squareResult = array.^2;
  - ✓ end

جامعة الحياة

Example 2: Subfunctions to divide a complex task

- function main()

```
% برنامج لحساب مساحة ومحيط دائرة
radius = 5;
area = computeArea(radius);
perimeter = computePerimeter(radius);
   fprintf('2.% :مساحة الدائرةf\n', area);
fprintf('2.% :محيط الدائرةf\n', perimeter);
```

- end

➢ function area = computeArea(r)

```
% دالة لحساب مساحة الدائرة
area = pi * r^2;
```

➢ end

❖ function perimeter = computePerimeter(r)

```
% دالة لحساب محيط الدائرة
perimeter = 2 * pi * r;
```

❖ end

م: كرم الجبالي

Example 3: Subfunction to calculate the square of a number.

```
function main()
    % البرنامج الرئيسي
    number = input('أدخل عددًا: ');
    square = computeSquare(number); % استدعاء الدالة الفرعية
    fprintf('2.% مربع العدد f 2.% هو f\n', number, square);
end

% دالة فرعية لحساب مربع عدد
function result = computeSquare(x)
    result = x^2;
end
```

Example 4: Using a subfunction Check if a number is prime using a subfunction.

في هذا المثال، سنقوم بإنشاء دالة رئيسية checkPrime تستدعي دالة فرعية isPrimeNumber للتحقق مما إذا كان الرقم أوليًا.

```
clear; clc; close all;
```

- ✓ `function result = checkPrime(number) % دالة رئيسية للتحقق مما إذا كان الرقم أوليًا`

```
if isPrimeNumber(number)
  result = [num2str(number), ' هو عدد أولي.'];
```

```
        else
            result = [num2str(number), ' ليس عددًا أوليًا.'];
        end
```

- ✓ end

```
% الدالة الفرعية للتحقق من أولية الرقم
```

- function isPrime = isPrimeNumber(n)

```
        isPrime = true;
        if n <= 1
            isPrime = false;
            return;
        end
        for i = 2:sqrt(n)
            if mod(n, i) == 0
            isPrime = false;
            return;
        end
        end
```

- end

```
num = 29;
result = checkPrime(num);
disp(result);
```

د. كريم الجبالي

ملاحظات مهمة:

1. يمكن أن يكون الملف يحتوي على أكثر من دالة فرعية.
2. يُنصح باستخدام أسماء واضحة تعبر عن وظيفة كل دالة.
3. تنظيم الكود باستخدام الدوال الفرعية يجعل الكود أكثر وضوحًا وأسهل للصيانة.

م: كرم الجبالي