Τμήμα Μηχανικών Σχεδίασης Προϊόντων και Συστημάτων

1101 ΠΛΗΡΟΦΟΡΙΚΗ

Δρ. Χρήστος Όροβας

ΕΔΙΠ Μαργαρίτης Δημήτριος

4. Αρχεία Συναρτήσεων

Αρχική έκδοση: Τσαγκαλίδου Ροδή

Αρχεία συναρτήσεων

Τα Αρχεία συναρτήσεων περιέχουν μια ακολουθία εντολών εισόδου με την οποία υπολογίζονται οι νέες μεταβλητές εξόδου.

Δεν «τρέχουν» από μόνα τους, αλλά καλούνται με το όνομά τους από άλλα προγράμματα ή από το command window.

Συναρτήσεις οριζόμενες από τον χρήστη

Μέχρι στιγμής έχουμε δει αρκετές συναρτήσεις που μας παρέχει το Matlab

Εκτός από αυτές υπάρχουν και οι *user-defined* που:

- Αποθηκεύονται σε ξεχωριστό m-file
- Δημιουργούνται από τον χρήστη
- Χρησιμοποιούνται όπως και οι *built in* δηλαδή:
  - μπορούν να χρησιμοποιηθούν για να αναθέσουν τιμή σε μία άλλη μεταβλητη
  - να τυπωθύν με μία εντολή disp ή fprintf

Αρχεία συναρτήσεων

Η δομή τους είναι:

function [output 1, output 2,..] =filename (input1, input2,..)

Το **filename** είναι το όνομα του function με το οποίο αποθηκεύεται στο m-file με το όνομα filename.m. Το όνομα αυτό δεν πρέπει να συμπίπτει με όνομα συνάρτησης βιβλιοθήκης.

Οι μεταβλητές εισόδου είναι σε παρενθέσεις και οι μεταβλητές εξόδου σε αγκύλες.

Δημιουργία αρχείου συνάρτησης

Επιλέγουμε από το μενού File/New/Function M-File

function[μεταβλητές εξόδου 1, μεταβλητές εξόδου 2]=filename (input 1, input 2)

δηλ.

function (output εντολες] = filename (input εντολές)

- Εντολές
- Εντολες b o d y

End

Οι μεταβλητές εισόδου είναι σε παρένθεση και οι μεταβλητές εξόδου σε αγκύλες.

Αρχεία συναρτήσεων

- Τα σχόλια είναι προαιρετικά
- Εντολές είναι ακολουθία εντολών με την οποία υπολογίζονται οι μεταβλητές εξόδου.
- Τα scripts files δε δέχονται ορίσματα εισόδου και εξόδου, ενώ τα functions files δέχονται.
- **Τα scripts files αποθηκεύουν τις μεταβλητές σε ένα workspace και τις μοιράζονται και με άλλα scripts**, ενώ τα functions αποθηκεύουν τις μεταβλητές σε ένα workspace εσωτερικά στη συνάρτηση.

Κλήση συνάρτησης – function call

- Για να χρησιμοποιηθεί μία συνάρτηση πρέπει να γίνει η κλήση της (***function call***) από ένα script. Συνήθως γίνεται σε μια εντολή ανάθεσης για να «κρατηθεί» το αποτέλεσμα ή σε μια εντολή εξόδου (π.χ disp(), fprintf()

- Όταν γίνεται ένα function call
  - Περνάει ο έλεγχος στην function
  - Οι τιμές των ορισμάτων περνάνε στην function
  - Εκτελούνται οι εντολές της function
  - Επιστρέφεται μία ή περισσότερες τιμές

Παράδειγμα

Το παρακάτω αρχείο συνάρτησης υπολογίζει το άθροισμα και το γινόμενο τριών αριθμών.

```
function [sum, prod] = sumprod(x1, x2, x3)
% όνομα συνάρτησης : sumprod
% όνομα m-file : sumprod.m
% μεταβλητές εισόδου x1, x2, x3
% μεταβλητές εξόδου sum, prod
sum=x1+x2+x3;
prod=x1*x2*x3;
end
```

Εκτέλεση του αρχείου συνάρτησης

Για να καλέσουμε το παραπάνω αρχείο θα γράψουμε στον Command Window:

```
>> [s,p] = sumprod(1, -3, 4)

s =

     2

p =

   -12
```

Για να καλέσουμε το παραπάνω αρχείο μέσω function call, θα δημιουργήσουμε ένα αρχείο script (πχ test)

```
[atrisma,ginomeno]=sumprod(1,-3,4)
```

και θα γράψουμε στον Command Window, το όνομα του αρχείου script : test

```
>>test


atrisma =

     2

ginomeno =

   -12
```

Βασικές διαφορές μεταξύ ενός *script file* και ενός *function file*:

- Script File

Περιγραφή: Ένα script file είναι ένα αρχείο που περιέχει μια σειρά από εντολές που εκτελούνται αμέσως με τη σειρά που γράφονται. Συνήθως χρησιμοποιείται για να εκτελεί απλές διαδικασίες ή διαδοχικά βήματα.

Δομή: Δεν χρειάζεται συγκεκριμένη σύνταξη. Απλώς περιέχει εντολές και ενέργειες.

Μεταβλητές: Οι μεταβλητές που δημιουργούνται σε ένα script file παραμένουν στο *workspace* και είναι προσβάσιμες μετά την εκτέλεση του script.

Χρήση: Το script file είναι ιδανικό για απλές εργασίες και για εντολές που χρειάζονται άμεση εκτέλεση, χωρίς ιδιαίτερη οργάνωση ή ανάγκη επανάχρησης του κώδικα.

Βασικές διαφορές μεταξύ ενός *script file* και ενός *function file*:

- Function File

Περιγραφή: Ένα function file περιέχει τον ορισμό μιας συνάρτησης που μπορεί να καλεστεί με διάφορες εισόδους (inputs) και να επιστρέψει εξόδους (outputs). Αυτό το είδος αρχείου επιτρέπει την επανάχρηση του κώδικα, καθώς μπορεί να καλείται από οποιοδήποτε σημείο του προγράμματος ή από άλλα αρχεία.

Δομή: Απαιτεί μια συγκεκριμένη σύνταξη που περιλαμβάνει την επικεφαλίδα της συνάρτησης.

Μεταβλητές: Οι μεταβλητές μέσα στη συνάρτηση είναι τοπικές (local) και δεν επηρεάζουν το workspace του χρήστη. Αυτό προστατεύει τις μεταβλητές από ανεπιθύμητες τροποποιήσεις.

Χρήση: Το function file χρησιμοποιείται όταν θέλουμε να δημιουργήσουμε έναν κώδικα που μπορεί να καλείται επαναλαμβανόμενα, με διαφορετικές εισόδους, χωρίς να χρειάζεται να ξαναγραφτεί.

Ανακεφαλαίωση: Πότε χρησιμοποιούμε το καθένα;

- **Script File**: Όταν η εργασία είναι απλή και δεν χρειάζεται επανάληψη ή όταν απλά θέλουμε να τρέξουμε μια σειρά από εντολές για άμεσο αποτέλεσμα.
- **Function File**: Όταν η εργασία είναι πιο σύνθετη και μπορεί να επαναχρησιμοποιηθεί ή να καλεστεί από άλλα κομμάτια του κώδικα με διαφορετικά δεδομένα.

Κατηγορίες συναρτήσεων

- Κατηγοριοποίηση συναρτήσεων matlab:
  1. Συναρτήσεις που επιστρέφουν μία τιμή
  2. Συναρτήσεις που επιστρέφουν πολλές τιμές
  3. Συναρτήσεις που δεν επιστρέφουν τίποτα (π.χ. απλώς τυπώνουν μια τιμή)
- Οι συναρτήσεις αυτές διαφέρουν:
  - Στον τρόπο κλήσης τους
  - Στο πως είναι το function header
- Όλες είναι αποθηκευμένες σε αρχεία με την επέκταση .m

Παράδειγμα συνάρτησης

- Για παράδειγμα, για μια συνάρτηση που υπολογίζει και επιστρέφει το εμβαδόν ενός κύκλου:
  - πρέπει να υπάρχει ένα όρισμα εισόδου (η ακτίνα)
  - πρέπει να υπάρχει ένα όρισμα εξόδου (το εμβαδόν)
  - **σε ένα αρχείο M** που ονομάζεται calcarea.m:

```
function area = calcarea(rad)
% Η συνάρτηση calcarea υπολογίζει το εμβαδόν ενός κύκλου
area = pi * rad * rad;
end
```

- Το **όνομα της συνάρτησης είναι ίδιο με το όνομα του αρχείου M**
- Η συνάρτηση δίνει μια τιμή στο όρισμα εξόδου, και με αυτόν τον τρόπο επιστρέφει την τιμή· στη συγκεκριμένη περίπτωση με μια εντολή ανάθεσης τιμής (*Σημείωση: η έξοδος πρέπει να αποκρύπτεται, με χρήση του ;)*
- Τα ονόματα των ορισμάτων εισόδου και εξόδου ακολουθούν τους ίδιους κανόνες με τις μεταβλητές και πρέπει να είναι μνημονικά

Κλήση της συνάρτησης

- Η συνάρτηση θα μπορούσε να κληθεί με αρκετούς τρόπους:
- >> calcarea(4)
  - Αυτή η κλήση θα αποθηκεύσει το αποτέλεσμα στην προεπιλεγμένη μεταβλητή ans
- >> myarea = calcarea(9)
  - Η συγκεκριμένη κλήση θα αποθηκεύσει το αποτέλεσμα στη μεταβλητή *myarea*

Παράδειγμα

function y=sind(x)

έχουμε ένα όρισμα εισόδου(x) και ένα εξόδου (y).

function name=area(a,b,h)

τρια ορίσματα εισόδου (a,b,h) ένα εξόδου(name).

function [a,b]=motion(v,angle)

δυο ορίσματα εισόδου(v,angle) και δυο εξόδου(a,b).

Μέρη ενός function definition

- ***Output args***: Οι μεταβλητές που θα περιέχουν τις τιμές που θα επιστρέψει η συνάρτηση
- ***funcname***: το όνομα της συνάρτησης (Πρέπει να είναι το ίδιο με το όνομα του αρχείου.Με αυτό το όνομα θα πρέπει να αποθηκευτεί το αρχείο συνάρτησης)
- ***%comment***: Ένα σχόλιο που εμφανίζει την λειτουργία της συνάρτησης (είναι αυτό που θα επιστρέφει η help)
- ***body***: όλες οι εντολές της συνάρτησης. Εδώ θα πρέπει να γίνεται ανάθεση τιμής στα output args αν υπάρχουν
- ***end***: Καθορίζει το τέλος των εντολών της συνάρτησης

Συναρτήσεις που επιστρέφουν πάνω από μία τιμή

Οι συναρτήσεις που επιστρέφουν πάνω από μία τιμή:

- στο ***output_args*** έχουν πάνω από ένα όνομα
- τα ονόματα στο output_args ***χωρίζονται μεταξύ τους με κόμμα*** και ***περιέχονται σε αγκύλες***

Κλήση συνάρτησης που επιστρέφει πολλές τιμές

- Αφού η συνάρτηση επιστρέφει πολλές τιμές θα πρέπει να φροντίσουμε να τις αποθηκεύσουμε.
- Αυτό γίνεται με την τοποθέτηση αριστερά του τελεστή ανάθεσης (operator =) πολλών μεταβλητών μέσα σε ένα vector.
- Ο αριθμός των μεταβλητών θα πρέπει να ισούται με τον αριθμό τον ορισμάτων εξόδου της συνάρτησης

Παραδείγματα function call

- Έστω ότι έχω την συνάρτηση με function header
  function [x,y,z]=fnname(a,b)

- Το παραπάνω καθορίζει ότι η συνάρτηση επιστρέφει 3 τιμές που πρέπει να τις αποθηκεύσω σε αντίστοιχες μεταβλητές όπως

  [g,h,t]=fnname(23,33);

- Μπορώ να χρησιμοποιήσω και τα ίδια ονόματα μιας και η συνάρτηση έχει δικό της workspace
  [x,y,z]=fnname(23,33);

- Αν βάλω μόνο μία μεταβλητή τότε θα αποθηκευτεί η τιμή μόνο του πρώτου ορίσματος εξόδου
  result=fnname(23,33);

Παράδειγμα συνάρτησης που επιστρέφει πολλές τιμές

Ένα παράδειγμα είναι μια συνάρτηση που επιστρέφει το εμβαδό και την περίμετρο ενός κύκλου

Πρέπει να φροντίσουμε να αποθηκεύουμε τις τιμές που επιστρέφει σε vector με δύο elements

Αλλιώς θα επιστρέφεται μόνο η τιμή μόνο του πρώτου ορίσματος

```
function [area, circum] = areacirc(rad)
% areacirc returns the area and
% the circumference of a circle
% Format: areacirc(radius)

area = pi * rad .* rad;
circum = 2 * pi * rad;
end
```

```
>> [a,c]=areacirc(4)
a =
   50.2655
c =
   25.1327
```

Συνάρτηση που επιστρέφει πολλές τιμές

- Η σειρά με την οποία γράφω τις μεταβλητές που θα λάβουν τις τιμές επιστροφής έχει σημασία;
  - ***ΝΑΙ έχει****. Η πρώτη μεταβλητή θα πάρει την τιμή του πρώτου ορίσματος επιστροφής, κτλ κτλ*

- Μπορώ να περάσω vector σαν όρισμα εισόδου στην συνάρτηση;
  - *ΝΑΙ και τότε και οι μεταβλητές που θα πάρουν τις τιμές εισόδου θα είναι η κάθε μια τους και ένα vector*

- Τι γίνεται αν θέλω να αποθηκεύσω μόνο την δεύτερη τιμή που επιστρέφει η συνάρτηση;
  - *Μπορώ για το όνομα της πρώτης μεταβλητής να χρησιμοποιήσω το ~. Αυτό είναι σαν να λέω στο Matlab αγνόησε την τιμή αυτή*

Πρόγραμμα που καλεί την συνάρτηση

Η συνάρτηση που δημιουργήσαμε μπορεί να κληθεί είτε από το Command Window είτε μέσα από ένα script όπως στο διπλανό παράδειγμα

```
% This script prompts the user for the radius of a circle,
%  calls a function to calculate and return both the area
%  and the circumference, and prints the results
% It ignores units and error-checking for simplicity

radius = input('Please enter the radius of the circle: ');
[area, circ] = areacirc(radius);
fprintf('For a circle with a radius of %.1f,\n', radius)
fprintf('the area is %.1f and the circumference is %.1f\n',...
    area, circ)
```

Συναρτήσεις που δεν επιστρέφουν τιμή

Υπάρχουν συναρτήσεις που δεν υπολογίζουν και επιστρέφουν τιμές αλλά απλώς εκτελούν μια ενέργεια π.χ. εκτύπωση

Αυτές δεν έχουν output arguments στο definition

Η γενική μορφή του definition είναι η διπλανή

```
function funcname( input_args )
%funcname Summary of this function goes here
%   Detailed explanation goes here

Statements here
end
```

Δεν μπορεί να χρησιμοποιηθεί σε εντολή ανάθεσης ή σε fprintf!!!

Προγραμματιστικές επιλογές

```
function calccircum1(radius)
% calccircum1 displays the circumference of a circle
%   but does not return the value
% Format: calccircum1(radius)

disp(2 * pi * radius)
end
```

```
function circle_circum = calccircum2(radius)
% calccircum2 calculates and returns the
%   circumference of a circle
% Format: calccircum2(radius)

circle_circum = 2 * pi * radius;
end
```

Συνάρτηση χωρίς ορίσματα εισόδου

Σε αυτή την περίπτωση στο definition το μόνο που έχουμε να γράψουμε είναι το όνομα της συνάρτησης

Υπάρχουν συναρτήσεις που δεν έχουν ορίσματα ούτε εισόδου, ούτε εξόδου

Η κλήση της μπορεί να γίνει και μόνο με το όνομα της (χωρίς παρενθέσεις)

```
function printrandnp
% printrandnp prints one random number
% Format: printrandnp or printrandnp()


fprintf('The random # is %.2f\n',rand)
end
```

Αντιστοιχία ορισμάτων στην κλήση συνάρτησης

- Κατά την κλήση μιας συνάρτησης τρία είναι τα ζητήματα που μας απασχολούν
  1. Να γράψουμε το **όνομα της συνάρτησης σωστά**!
  2. Αν «δίνουμε» τα κατάλληλα **ορίσματα** έτσι ώστε η συνάρτηση να εκτελέσει την λειτουργία τους. Είναι απαραίτητα δηλαδή ο **σωστός ΑΡΙΘΜΟΣ** ορισμάτων και η **σωστή ΣΕΙΡΑ** ΤΟΥΣ
  3. **Αν «λαμβάνουμε» τις τιμές που επιστρέφει** αυτή η συνάρτηση. Εδώ μπορούμε να «παραλείψουμε» την λήψη μιας ή περισσοτέρων τιμών αν θέλουμε

Αντιστοιχία ορισμάτων στην κλήση συνάρτησης

- Έστω ότι έχουμε το ακόλουθο function header
- Ο μ
- Ποιες από τις παρακάτω κλήσεις της συνάρτησης είναι σωστές και ποιες λάθος;

```
> function [outa, outb] = qql(x,y,z)
```

```
(a) [var1, var2]=qql(a,b,c);
(b) answer =qql(3,y,q);
(c) [a,b] = myfun(x,y,z);
(d) [outa,outb]=qql(x,y);
```