

ΔΗΜΙΟΥΡΓΙΑ ΣΥΝΑΡΤΗΣΕΩΝ

Υπάρχει ένα σύνολο από ασκήσεις που μας ζητάει να δημιουργήσουμε μία συνάρτηση στη γλώσσα python η οποία να επιστρέφει ή να εμφανίζει κάποιες τιμές.

Η δημιουργία μιας συνάρτησης είναι πάρα πολύ κοντά στη δημιουργία ενός κανονικού προγράμματος που να κάνει την ίδια “δουλειά”.

Και πάλι θα πρέπει να έχουμε πολύ καλή γνώση των εντολών της python και **να διαβάσουμε πάρα πολύ καλά** την εκφώνηση της άσκησης. Αυτή (Η εκφώνηση) θα μας καθοδηγήσει στο πως θα την ολοκληρώσουμε με επιτυχία.

ΔΗΜΙΟΥΡΓΙΑ ΣΥΝΑΡΤΗΣΕΩΝ

Άσκηση 8Α

Άσκηση 8Α:

Να γράψετε συνάρτηση SYN που να δέχεται έναν αριθμό και να επιστρέφει το διπλάσιό του, αν είναι μικρότερος του 10, διαφορετικά να επιστρέφει το τριπλάσιό του.

(ΘΕΜΑ Β3 2017)

ΔΗΜΙΟΥΡΓΙΑ ΣΥΝΑΡΤΗΣΕΩΝ**Άσκηση 8Α:**

Να γράψετε συνάρτηση **SYN** που να δέχεται έναν αριθμό και να επιστρέφει το διπλάσιό του, αν είναι μικρότερος του 10, διαφορετικά να επιστρέφει το τριπλάσιό του.

(ΘΕΜΑ Β3 2017)

Άσκηση 8Α

Το πρώτο πράγμα που κάνω είναι να γράψω τη λέξη **def** και το όνομα της συνάρτησης αν αυτό μου δίνεται. Αν δεν μου δινόταν θα μπορούσα να βάλω εγώ ότι όνομα θέλω.

ΛΥΣΗ:

def SYN(

ΔΗΜΙΟΥΡΓΙΑ ΣΥΝΑΡΤΗΣΕΩΝ**Άσκηση 8A:**

Να γράψετε συνάρτηση SYN που να δέχεται έναν αριθμό και να επιστρέφει το διπλάσιό του, αν είναι μικρότερος του 10, διαφορετικά να επιστρέφει το τριπλάσιό του.

(ΘΕΜΑ Β3 2017)

Άσκηση 8A

Ρωτάω πόσα πράγματα να δέχεται; ένα. Τι ένα; Ένα αριθμό. Άρα χρειάζομαι μία μεταβλητή την οποία ονομάζω όπως εγώ θέλω και παίζει το ρόλο της παραμέτρου. π.χ. a

ΛΥΣΗ:

def SYN(a):

Σημαντική σημείωση:

Η παραπάνω μεταβλητή (a) θα ενημερωθεί με τιμή όταν θα καλέσουν τη συνάρτηση. Εγώ μέσα στη συνάρτηση θεωρώ ότι είναι μία μεταβλητή που έχει ήδη τιμή (π.χ. η a έχει ήδη έναν αριθμό μέσα της).

ΔΗΜΙΟΥΡΓΙΑ ΣΥΝΑΡΤΗΣΕΩΝ

Άσκηση 8Α

Άσκηση 8Α:

Να γράψετε συνάρτηση SYN που να δέχεται έναν αριθμό και να επιστρέφει το διπλάσιό του, αν είναι μικρότερος του 10, διαφορετικά να επιστρέφει το τριπλάσιό του.

(ΘΕΜΑ Β3 2017)

Μεταφέρω στην Python: Αν ο a είναι μικρότερος του 10 να επιστρέφει το διπλάσιό του ($2*a$)

ΛΥΣΗ:

```
def SYN(a):  
    if a<10:  
        return 2*a
```

Σημαντική σημείωση:

Δεν είναι υποχρεωτικό μία συνάρτηση να επιστρέφει μία τιμή. Μπορεί απλά να την εμφανίζει.(π.χ. με print)

ΔΗΜΙΟΥΡΓΙΑ ΣΥΝΑΡΤΗΣΕΩΝ**Άσκηση 8Α:**

Να γράψετε συνάρτηση SYN που να δέχεται έναν αριθμό και να επιστρέφει το διπλάσιό του, αν είναι μικρότερος του 10, **διαφορετικά να επιστρέφει το τριπλάσιό του.**

(ΘΕΜΑ Β3 2017)

Άσκηση 8Α

Μεταφέρω στην Python: Διαφορετικά (αλλιώς) να επιστρέφει το τριπλάσιό του

ΛΥΣΗ:

```
def SYN(a):  
    if a<10:  
        return 2*a  
    else:  
        return 3*a
```

Αυτή είναι και η λύση της Άσκησης 8Α.

ΔΗΜΙΟΥΡΓΙΑ ΣΥΝΑΡΤΗΣΕΩΝ

Άσκηση 8B

Άσκηση 8B:

Να γράψετε συνάρτηση `EMBADO_TRIG` που να δέχεται τη βάση και το ύψος ενός τριγώνου και να επιστρέφει εμβαδόν του.

Στη συνέχεια χρησιμοποιώντας την παραπάνω συνάρτηση να υπολογίσετε και να εμφανίσετε το εμβαδόν ενός τριγώνου με βάση 4 εκ. και ύψος 3 ε.

ΔΗΜΙΟΥΡΓΙΑ ΣΥΝΑΡΤΗΣΕΩΝ

Άσκηση 8B

Άσκηση 8B:

Να γράψετε συνάρτηση **EMBADO_TRIG** που να δέχεται τη βάση και το ύψος ενός τριγώνου και να επιστρέφει εμβαδόν του.

Στη συνέχεια χρησιμοποιώντας την παραπάνω συνάρτηση να υπολογίσετε και να εμφανίσετε το εμβαδόν ενός τριγώνου με βάση 4 εκ. και ύψος 3 ε.

Το πρώτο πράγμα που κάνω είναι να γράψω τη λέξη **def** και το όνομα της συνάρτησης αν αυτό μου δίνεται. Αν δεν μου δινόταν θα μπορούσα να βάλω εγώ ότι όνομα θέλω.

ΛΥΣΗ:

```
def EMBADO_TRIG(
```


ΔΗΜΙΟΥΡΓΙΑ ΣΥΝΑΡΤΗΣΕΩΝ**Άσκηση 8B:**

Να γράψετε συνάρτηση `EMBADO_TRIG` που να δέχεται τη βάση και το ύψος ενός τριγώνου και να επιστρέφει εμβαδόν του.

Στη συνέχεια χρησιμοποιώντας την παραπάνω συνάρτηση να υπολογίσετε και να εμφανίσετε το εμβαδόν ενός τριγώνου με βάση 4 εκ. και ύψος 3 ε.

Άσκηση 8B

Ρωτάω πόσα πράγματα να δέχεται; Δύο. Τι δύο; Τη βάση και το ύψος ενός τριγώνου. Δύο αριθμούς Άρα χρειάζομαι δύο μεταβλητές τις οποίες ονομάζω όπως εγώ θέλω και παίζουν το ρόλο των παραμέτρων. πχ. `b` (βάση) και `u` (ύψος)

ΛΥΣΗ:

```
def EMBADO_TRIG( b,u):
```

Σημαντική σημείωση:

Οι παραπάνω μεταβλητές (`b,u`) θα ενημερωθούν με τιμές όταν κληθεί η συνάρτηση. **Εγώ μέσα στη συνάρτηση δουλεύω θεωρώντας ότι οι παράμετροι `b,u` έχουν ήδη τιμές (Δεν χρειάζονται input για να πάρουν τιμές)**

ΔΗΜΙΟΥΡΓΙΑ ΣΥΝΑΡΤΗΣΕΩΝ**Άσκηση 8B:**

Να γράψετε συνάρτηση EMBADO_TRIG που να δέχεται τη βάση και το ύψος ενός τριγώνου και **να επιστρέφει εμβαδόν του**.

Στη συνέχεια χρησιμοποιώντας την παραπάνω συνάρτηση να υπολογίσετε και να εμφανίσετε το εμβαδόν ενός τριγώνου με βάση 4 εκ. και ύψος 3 ε.

Άσκηση 8B

Το εμβαδόν ενός τριγώνου προκύπτει από τον τύπο: $E=(b * u)/2$. Άρα πρώτα θα το υπολογίσω και μετά θα το “επιστρέψω”. Είπαμε στην προηγούμενη διαφάνεια ότι τα b και u θεωρούνται γνωστά (έχουν ήδη τιμή), άρα απλά υπολογίζω το εμβαδόν και το “επιστρέφω” (return)

ΛΥΣΗ:

```
def EMBADO_TRIG( b,u):  
    e=(b * u)/2.0  
    return e
```

Η συνάρτησή μας είναι έτοιμη!
Μία άλλη λύση θα μπορούσε να είναι :

```
def EMBADO_TRIG2( b,u):  
    return (b * u)/2.0
```

ΔΗΜΙΟΥΡΓΙΑ ΣΥΝΑΡΤΗΣΕΩΝ

Άσκηση 8B

Άσκηση 8B:

Να γράψετε συνάρτηση EMBADO_TRIG που να δέχεται τη βάση και το ύψος ενός τριγώνου και να επιστρέφει εμβαδόν του.

Στη συνέχεια χρησιμοποιώντας την παραπάνω συνάρτηση να υπολογίσετε και να εμφανίσετε το εμβαδόν ενός τριγώνου με βάση 4 εκ. και ύψος 3 ε.

Έχω δημιουργήσει μία δικιά μου συνάρτηση που υπολογίζει το εμβαδόν ενός τριγώνου. Αυτή τη συνάρτηση τη χρησιμοποιώ σαν να είναι μία συνάρτηση που προϋπήρχε στην Python. Αρκεί να την καλέσω γράφοντας το όνομά της και δίνοντας δύο αριθμούς, έναν για βάση και έναν για ύψος και αυτή θα μου επιστρέψει το εμβαδόν. Χρησιμοποιώ μία νέα μεταβλητή για το εμβαδόν στο κυρίως πρόγραμμά μου την emb.

ΛΥΣΗ:

```
def EMBADO_TRIG( b,u):  
    e=(b * u)/2.0  
    return e
```

```
emb=EMBADO_TRIG( 4,3)  
print emb
```

ΔΗΜΙΟΥΡΓΙΑ ΣΥΝΑΡΤΗΣΕΩΝ

Άσκηση 8Γ

Άσκηση 8Γ:

Να γράψετε συνάρτηση MAX3 που να δέχεται τρεις αριθμούς και να επιστρέφει το μέγιστο.

ΔΗΜΙΟΥΡΓΙΑ ΣΥΝΑΡΤΗΣΕΩΝ**Άσκηση 8Γ:**

Να γράψετε **συνάρτηση MAX3** που να δέχεται τρεις αριθμούς και να επιστρέφει το μέγιστο.

Άσκηση 8Γ

Το πρώτο πράγμα που κάνω είναι να γράψω τη λέξη `def` και το όνομα της συνάρτησης αν αυτό μου δίνεται. Αν δεν μου δινόταν θα μπορούσα να βάλω εγώ ότι όνομα θέλω.

ΛΥΣΗ:

```
def MAX3(
```

ΔΗΜΙΟΥΡΓΙΑ ΣΥΝΑΡΤΗΣΕΩΝ**Άσκηση 8Γ:**

Να γράψετε συνάρτηση MAX3 που να δέχεται τρεις αριθμούς και να επιστρέφει το μέγιστο.

Άσκηση 8Γ

Ρωτάω πόσα πράγματα να δέχεται; Τρία. Τι τρία; τρεις αριθμούς Άρα χρειάζομαι τρεις μεταβλητές τις οποίες ονομάζω όπως εγώ θέλω και παίζουν το ρόλο των παραμέτρων. πχ. a,b,c.

ΛΥΣΗ:

```
def MAX3(a,b,c):
```

ΔΗΜΙΟΥΡΓΙΑ ΣΥΝΑΡΤΗΣΕΩΝΆσκηση 8Γ:

Να γράψετε συνάρτηση MAX3 που να δέχεται τρεις αριθμούς και **να επιστρέφει το μέγιστο**.

Άσκηση 8Γ

Υπάρχουν διάφοροι τρόποι να βρούμε το μέγιστο από 3 αριθμούς. Εγώ θα επιλέξω τον πιο εύκολο. Αν έχω 3 αριθμούς **θεωρώ ότι ο μέγιστος είναι ο Πρώτος**. Αν ο Δεύτερος είναι μεγαλύτερος από το μέγιστο που έχω εκείνη τη στιγμή τότε ο νέος μέγιστος είναι ο δεύτερος. Αν ο Τρίτος είναι μεγαλύτερος από το μέγιστο που έχω εκείνη τη στιγμή τότε ο νέος μέγιστος είναι ο Τρίτος. Θα χρειαστώ μία μεταβλητή MAX για να κρατάω τη μέγιστη τιμή κάθε στιγμή.

ΛΥΣΗ:

```
def MAX3(a,b,c):  
    MAX=a
```

ΔΗΜΙΟΥΡΓΙΑ ΣΥΝΑΡΤΗΣΕΩΝ**Άσκηση 8Γ:**

Να γράψετε συνάρτηση MAX3 που να δέχεται τρεις αριθμούς και **να επιστρέφει το μέγιστο**.

Άσκηση 8Γ

Υπάρχουν διάφοροι τρόποι να βρούμε το μέγιστο από 3 αριθμούς. Εγώ θα επιλέξω τον πιο εύκολο. Αν έχω 3 αριθμούς θεωρώ ότι ο μέγιστος είναι ο Πρώτος. **Αν ο Δεύτερος είναι μεγαλύτερος από το μέγιστο που έχω εκείνη τη στιγμή τότε ο νέος μέγιστος είναι ο δεύτερος.** Αν ο Τρίτος είναι μεγαλύτερος από το μέγιστο που έχω εκείνη τη στιγμή τότε ο νέος μέγιστος είναι ο Τρίτος. Θα χρειαστώ μία μεταβλητή MAX για να κρατάω τη μέγιστη τιμή κάθε στιγμή.

ΛΥΣΗ:

```
def MAX3(a,b,c):  
    MAX=a  
    if b>MAX:  
        MAX=b
```


ΔΗΜΙΟΥΡΓΙΑ ΣΥΝΑΡΤΗΣΕΩΝΆσκηση 8Γ:

Να γράψετε συνάρτηση MAX3 που να δέχεται τρεις αριθμούς και να επιστρέφει το μέγιστο.

Άσκηση 8Γ

Υπάρχουν διάφοροι τρόποι να βρούμε το μέγιστο από 3 αριθμούς. Εγώ θα επιλέξω τον πιο εύκολο.

Αν έχω 3 αριθμούς θεωρώ ότι ο μέγιστος είναι ο Πρώτος. Αν ο Δεύτερος είναι μεγαλύτερος από το μέγιστο που έχω εκείνη τη στιγμή τότε ο νέος μέγιστος είναι ο δεύτερος. Αν ο Τρίτος είναι μεγαλύτερος από το μέγιστο που έχω εκείνη τη στιγμή τότε ο νέος μέγιστος είναι ο Τρίτος. Θα χρειαστώ μία μεταβλητή MAX για να κρατάω τη μέγιστη τιμή κάθε στιγμή.

ΛΥΣΗ:

```
def MAX3(a,b,c):  
    MAX=a  
    if b>MAX:  
        MAX=b  
    if c>MAX:  
        MAX=c
```

ΔΗΜΙΟΥΡΓΙΑ ΣΥΝΑΡΤΗΣΕΩΝ

Άσκηση 8Γ

Άσκηση 8Γ:

Να γράψετε συνάρτηση MAX3 που να δέχεται τρεις αριθμούς και να επιστρέφει το μέγιστο.

ΛΥΣΗ:

```
def MAX3(a,b,c):  
    MAX=a  
    if b>MAX:  
        MAX=b  
    if c>MAX:  
        MAX=c  
    return MAX
```

Αυτή είναι και η λύση της άσκησης.
Εναλλακτικά δίνεται μία ακόμη λύση στην επόμενη διαφάνεια.

ΔΗΜΙΟΥΡΓΙΑ ΣΥΝΑΡΤΗΣΕΩΝ**Άσκηση 8Γ:**

Να γράψετε συνάρτηση MAX3 που να δέχεται τρεις αριθμούς και να επιστρέφει το μέγιστο.

Άσκηση 8Γ

ΛΥΣΗ (β):

```
def MAX3(a,b,c):  
    if a>=b and a>=c:  
        MAX=a  
    elif b>=a and b>=c:  
        MAX=b  
    elif c>=a and c>=b:  
        MAX=c  
    return MAX
```