

Εισαγωγή στον Προγραμματισμό Η/Υ

Διδάσκοντες: Ι. Πολενάκης, Κ. Σκιαδόπουλος, Δ. Μουρατίδου

Τμήμα Πληροφορικής, Ιόνιο Πανεπιστήμιο



2. Δομές Ελέγχου

➤ Λήψη αποφάσεων: if

❑ Εκτέλεση κώδικα βάσει συνθήκης

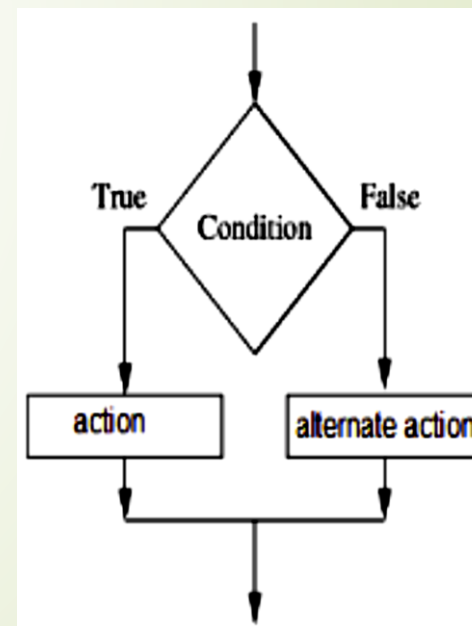
```
If: if <συνθήκη>:  
    <action>
```

❑ Εκτελεί κώδικα μιας ενέργειας αν η συνθήκη είναι True

```
if...else: if <συνθήκη>:  
    <action 1>  
else:  
    <action 2>
```

❑ Εκτελεί μια ενέργεια αν η συνθήκη είναι True, ή εκτελεί μια άλλη ενέργεια αν είναι False

```
>>> grade = 6.5  
>>> if grade >= 5.0:  
    print('Passed')  
else:  
    print('Failed')  
Passed
```



2. Δομές Ελέγχου

➤ Λήψη αποφάσεων: if

❑ Απλή έκδοση

```
if condition:  
    command
```

```
>>> if number1 == number2:  
    print(number1, 'is equal to', number2)
```

❑ «Full» έκδοση

```
>>> x = int(input("Enter an integer: "))  
>>> if x < 0:  
    print ('Negative number, transforming into positive')  
    x = -x #make positive  
elif x == 0:  
    print('Zero')  
elif x == 1:  
    print('One')  
else:  
    print('Great than 1')
```

2. Δομές Ελέγχου

► Παράδειγμα: Πράξεις δύο ακεραίων αριθμών

```
# Είσοδος δεδομένων

a = int(input("Δώσε τον πρώτο ακέραιο αριθμό: "))
b = int(input("Δώσε τον δεύτερο ακέραιο αριθμό: "))

# Υπολογισμοί

athroisma = a + b

diafora = a - b

ginomeno = a * b

# Εμφάνιση αποτελεσμάτων

print("\nΑποτελέσματα:")

print("Αθροισμα:", athroisma)

print("Διαφορά:", diafora)

print("Γινόμενο:", ginomeno)

# Έλεγχος για διαίρεση

if b != 0:

    piliko = a / b

    print("Πηλίκο: {:.2f}".format(piliko))

else:

    print("Δεν μπορεί να γίνει διαίρεση με το μηδέν!")
```

2. Δομές Ελέγχου

► Παράδειγμα: Πράξεις δύο ακεραιών αριθμών

```
# Είσοδος δεδομένων
a = int(input("Δώσε τον πρώτο ακέραιο αριθμό: "))
b = int(input("Δώσε τον δεύτερο ακέραιο αριθμό: "))

# Υπολογισμοί
athroisma = a + b
diafora = a - b
ginomeno = a * b

# Εμφάνιση αποτελεσμάτων
print("\nΑποτελέσματα:")
print("Άθροισμα:", athroisma)
print("Διαφορά:", diafora)
print("Γινόμενο:", ginomeno)

# Έλεγχος για διαίρεση
if b != 0:
    piliko = a / b
    print("Πηλίκο: {:.2f}".format(piliko))
else:
    print("Δεν μπορεί να γίνει διαίρεση με το μηδέν!")
```

- $a = \text{int}(\text{input}(\dots))$ Η συνάρτηση `input()` διαβάζει δεδομένα ως **κείμενο (string)**. Με τη `int()` μετατρέπουμε το κείμενο σε **ακέραιο αριθμό**.
- Υπολογίζονται οι πράξεις:
 - $\text{athroisma} = a + b$
 - $\text{diafora} = a - b$
 - $\text{ginomeno} = a * b$
- Τα αποτελέσματα εμφανίζονται με την εντολή `print()`.
- Η δομή **if** ελέγχει αν ο δεύτερος αριθμός είναι μηδέν:
 - Αν **δεν είναι μηδέν**, εκτελείται η διαίρεση και εμφανίζεται το αποτέλεσμα με 2 δεκαδικά ψηφία (`{:.2f}`).
 - Αν **είναι μηδέν**, εμφανίζεται κατάλληλο μήνυμα.

2. Δομές Ελέγχου

► Παράδειγμα: Πράξεις δύο ακεραιών αριθμών

```
# Είσοδος δεδομένων
a = int(input("Δώσε τον πρώτο ακέραιο αριθμό: "))
b = int(input("Δώσε τον δεύτερο ακέραιο αριθμό: "))

# Υπολογισμοί
athroisma = a + b
diafora = a - b
ginomeno = a * b

# Εμφάνιση αποτελεσμάτων
print("\nΑποτελέσματα:")
print("Άθροισμα:", athroisma)
print("Διαφορά:", diafora)
print("Γινόμενο:", ginomeno)

# Έλεγχος για διαίρεση
if b != 0:
    piliko = a / b
    print("Πηλίκο: {:.2f}".format(piliko))
else:
    print("Δεν μπορεί να γίνει διαίρεση με το μηδέν!")
```

Εκτέλεση 1:

Δώσε τον πρώτο ακέραιο αριθμό: 10
Δώσε τον δεύτερο ακέραιο αριθμό: 3

Αποτελέσματα:
Άθροισμα: 13
Διαφορά: 7
Γινόμενο: 30
Πηλίκο: 3.33

2. Δομές Ελέγχου

► Παράδειγμα: Πράξεις δύο ακεραιών αριθμών

```
# Είσοδος δεδομένων
a = int(input("Δώσε τον πρώτο ακέραιο αριθμό: "))
b = int(input("Δώσε τον δεύτερο ακέραιο αριθμό: "))

# Υπολογισμοί
athroisma = a + b
diafora = a - b
ginomeno = a * b

# Εμφάνιση αποτελεσμάτων
print("\nΑποτελέσματα:")
print("Άθροισμα:", athroisma)
print("Διαφορά:", diafora)
print("Γινόμενο:", ginomeno)

# Έλεγχος για διαίρεση
if b != 0:
    piliko = a / b
    print("Πηλίκο: {:.2f}".format(piliko))
else:
    print("Δεν μπορεί να γίνει διαίρεση με το μηδέν!")
```

Εκτέλεση 2:

Δώσε τον πρώτο ακέραιο αριθμό: 8
Δώσε τον δεύτερο ακέραιο αριθμό: 0

Αποτελέσματα:

Άθροισμα: 8

Διαφορά: 8

Γινόμενο: 0

Δεν μπορεί να γίνει διαίρεση με το μηδέν!

2. Δομές Ελέγχου

➡ Εντολές επανάληψης

Δομές επανάληψης (iteration statements)

Επαναληπτική εκτέλεση κώδικα

- **while** - Επανάληψη ενέργειας όσο η συνθήκη παραμένει True

```
>>> studyPctg = 20
>>> while studyPctg < 65:
    studyPctg = studyPctg + 20
>>> studyPctg
```

80

- **for** - Επανάληψη ενέργειας για κάθε αντικείμενο σε μια ακολουθία αντικειμένων

```
>>> for character in 'Η Python τα σπάει': print(character, end=' ')
'Η P y t h o n   τ α   σ π ά ε ι '
```

2. Δομές Ελέγχου

➡ Λέξεις κλειδιά

- ❑ Δεσμευμένες λέξεις για λειτουργικότητα
- ❑ Δεν μπορείτε να τα χρησιμοποιήσετε σαν αναγνωριστικά

Λ.χ. όνομα μεταβλητής -> συντακτικό σφάλμα

- ❑ Συλλαβισμός ακριβώς έτσι (πεζά-κεφαλαία)

•and	•def	•from	•nonlocal	•while
•as	•del	•global	•not	•with
•assert	•elif	•if	•or	•yield
•async	•else	•import	•pass	
•await	•except	•in	•raise	
•break	•False	•is	•return	
•class	•finally	•lambda	•True	
•continue	•for	•None	•try	

2. Δομές Ελέγχου

► Tips...

❑ Κάθε έκφραση μπορεί να ερμηνευτεί ως True ή False

- Μη μηδενικές -> True, δηλ. όλες οι συμβολοσειρές που έχουν χαρακτήρες
- Μηδενικές -> False, δηλ. οι συμβολοσειρές που δεν έχουν χαρακτήρες
- “==” ≠ “=”
- “==”: τελεστής ισότητας -> επιστρέφει True ή False
- “=”: εντολή εκχώρησης -> αναθέτει τιμή σε ένα αντικείμενο

```
>>> grade = 6.5
>>> if grade = 4.0:
    print('Failed')
else:
    print('Passed')
SyntaxError: invalid syntax
```

❑ Προσοχή στις εσοχές

Το block εκτέλεσης if, if...else, if...elif...else, for, while, κλπ προσδιορίζεται από το βάθος εσοχής

```
>>> grade = 100
>>> if grade >= 60:
    print('Passed')
else:
    print('Failed')
    print('You must take this course again')
```

2. Δομές Ελέγχου

► Tips...

- ☐ Ονοματισμένα ορίσματα
- ☐ Η δεσμευμένη λέξη-όρισμα (keyword ή named arguments)
- ☐ Ορίσματα συναρτήσεων: ονόματα ρητά προσδιορισμένα από τη συνάρτηση

```
>>> for character in 'Η Python τα σπάει':    print(character, end=' ')
```

docs.python.org/3/library/functions.html#print

```
print(*objects, sep=' ', end='\n', file=sys.stdout, flush=False)
```

Print *objects* to the text stream *file*, separated by *sep* and followed by *end*. *sep*, *end*, *file* and *flush*, if present, must be given as keyword arguments.

- Αντίστοιχα μπορούμε να ορίσουμε το “sep”
- Η προεπιλεγμένη τιμή για τις “sep” & “end” είναι επίσης στον ορισμό της συνάρτησης

```
>>> print(10,20,30, sep="; ")
```

```
'10; 20; 30'
```

```
print(*objects, sep=' ', end='\n', file=sys.stdout, flush=False)
```