

Λογικές Εκφράσεις

Μια **λογική έκφραση** είναι ένα ερώτημα αν κάτι ισχύει ή όχι, δηλαδή αν είναι **ΑΛΗΘΗΣ** ή **ΨΕΥΔΗΣ**. Στη python αυτά ονομάζονται οι **λογικές τιμές True** (ΑΛΗΘΗΣ) ή **False** (ΨΕΥΔΗΣ).

Πχ η λογική έκφραση **5 < 9** “ρωτάει” αν το 5 είναι μικρότερο του 9, κάτι το οποίο ισχύει (είναι αληθές), άρα η απάντηση στο ερώτημα αν το $5 < 9$ είναι “True”. Αντίστοιχα αν $\alpha = 10$, $\beta = 20$, το ερώτημα $\alpha > \beta$ απαντάει με “False” αφού το 10 δεν είναι μεγαλύτερο από το 20.

Σημείωση: Στη python οι λογικές τιμές γράφονται **True** και **False**, όχι true, TRUE, false, FALSE κτλ.

ΠΙΝΑΚΑΣ ΑΛΗΘΕΙΑΣ - Οι Λογικοί Τελεστές and, or, not (στα ελληνικά και, ή, όχι)

Μπορούμε να συνδυάσουμε απλές λογικές εκφράσεις για να δημιουργήσουμε πιο σύνθετες λογικές εκφράσεις. Αν **A** και **B** είναι δύο λογικές εκφράσεις, δηλαδή το **A** είναι True ή False όπως και το **B** είναι True ή False, τότε:

and: Η έκφραση **A and B** “ρωτάει” αν και το **A** και το **B** ταυτόχρονα είναι True.

or: Η έκφραση **A or B** “ρωτάει” αν οποιοδήποτε από τα **A** ή **B** είναι True.

not: Η έκφραση **not A** “ρωτάει” ποιο είναι το αντίθετο του **A**, δηλαδή αν το **A** είναι True απαντάει με False, αν το **A** είναι False απαντάει με True.

A	B	A and B (και)	A or B (ή)	not A (όχι)
TRUE ΑΛΗΘΗΣ	TRUE ΑΛΗΘΗΣ	TRUE ΑΛΗΘΗΣ	TRUE ΑΛΗΘΗΣ	FALSE ΨΕΥΔΗΣ
TRUE ΑΛΗΘΗΣ	FALSE ΨΕΥΔΗΣ	FALSE ΨΕΥΔΗΣ	TRUE ΑΛΗΘΗΣ	FALSE ΨΕΥΔΗΣ
FALSE ΨΕΥΔΗΣ	TRUE ΑΛΗΘΗΣ	FALSE ΨΕΥΔΗΣ	TRUE ΑΛΗΘΗΣ	TRUE ΑΛΗΘΗΣ
FALSE ΨΕΥΔΗΣ	FALSE ΨΕΥΔΗΣ	FALSE ΨΕΥΔΗΣ	FALSE ΨΕΥΔΗΣ	TRUE ΑΛΗΘΗΣ

Η προτεραιότητα των πράξεων γίνεται ως εξής:

α. Αριθμητικοί Τελεστές	1. (...) πράξεις σε παρενθέσεις 2. ** δύναμη 3. * πολ/σμός, / διαίρεση, // πηλίκο, % υπόλοιπο 4. + πρόσθεση, - αφαίρεση
β. Συγκριτικοί Τελεστές	1. < μικρότερο, > μεγαλύτερο, == ίσο, != διαφορετικό, <= μικρότερο ή ίσο, >= μεγαλύτερο ή ίσο
γ. Λογικοί Τελεστές	1. not (όχι) 2. and (και) 3. or (ή)

Γίνονται πρώτα οι αριθμητικές πράξεις, μετά οι συγκρίσεις και τέλος οι λογικές πράξεις. Σε κάθε κατηγορία υπάρχει επίσης προτεραιότητα πράξεων, πχ στις αριθμητικές πράξεις προηγούνται οι παρενθέσεις, μετά οι δυνάμεις κτλ. Πράξεις στο ίδιο επίπεδο γίνονται με τη σειρά που εμφανίζονται.

Παράδειγμα: στην λογική έκφραση **10 < 1 + 6 or not 3**2 > 9**, θα γίνουν πρώτα οι αριθμητικές πράξεις, αρχικά το 3^{**2} και μετά η πρόσθεση $1 + 6$. Οπότε η έκφραση γίνεται **10 < 7 or not 9 > 9**. Στη συνέχεια θα υπολογιστούν οι συγκρίσεις $10 < 7$ (που είναι False) και $9 > 9$ (που είναι False) άρα η έκφραση τώρα γίνεται **False or not False**. Έχουν μείνει μόνο οι λογικοί τελεστές οπότε προηγείται το **not** του **or**, άρα η έκφραση γίνεται **False or True**, άρα το τελικό αποτέλεσμα είναι **True**.

Αντίθεση με χρήση του not

Αν συγκρίνουμε ένα αριθμό **α** με έναν αριθμό **β**, ισχύει πάντα μόνο ένα από τα παρακάτω:

- Το α να είναι μικρότερο του β ($\alpha < \beta$)
- Το α να είναι μεγαλύτερο του β ($\alpha > \beta$)
- Το α να είναι ίσο με το β ($\alpha = \beta$)

Πχ στην έκφραση **not $\alpha < \beta$** , αν το $\alpha < \beta$ είναι **True**, το **not** το αλλάζει σε **False**, δηλαδή δεν ισχύει το $<$, συνεπώς θα ισχύει το $>$ ή το $=$, δηλαδή οι άλλες δύο περιπτώσεις. Οπότε μπορούμε να εξαλείψουμε το **not** και να δημιουργήσουμε μια ισοδύναμη έκφραση, δηλαδή **not $\alpha < \beta \Leftrightarrow \alpha \geq \beta$** . Αντίστοιχα **not $\alpha \leq \beta \Leftrightarrow \alpha > \beta$** , **not $\alpha = \beta \Leftrightarrow \alpha \neq \beta$** κτλ.

Σημείωση: Στη python ο έλεγχος ισότητας συμβολίζεται με **==** και της ανισότητας με **!=**. Συνεπώς στη python γράφουμε **not $\alpha == \beta$** που είναι ισοδύναμο με το **$\alpha != \beta$** .

Επίσης αν A, B είναι λογικές εκφράσεις, ισχύει:

- **not (A or B) $\Leftrightarrow$ not A and not B**
- **not (A and B) $\Leftrightarrow$ not A or not B**
- **not (not A) $\Leftrightarrow$ A**

Παραδείγματα:

- **not not $x < 5$** γίνεται **$x < 5$** , γιατί **not not $x < 5 \Leftrightarrow$ not $x \geq 5 \Leftrightarrow x < 5$** .
- **not ($\beta < 0$ and not $\gamma \geq 10$)** γίνεται **not $\beta < 0$ or not not $\gamma \geq 10$** (δηλαδή το **not** έξω από την παρένθεση πάει μπροστά από κάθε σύγκριση μέσα στη παρένθεση και αλλάζουμε το **and** με το **or**). Με τους παραπάνω κανόνες έχουμε τελικά **$\beta \geq 0$ or $\gamma \geq 10$** .

Άσκηση 1 - Απαντήστε στις παρακάτω ερωτήσεις με τη λέξη **Σωστό** αν κρίνετε ότι η πρόταση είναι ορθή ή τη λέξη **Λάθος** αν πιστεύετε ότι είναι λανθασμένη.

1. Η συνθήκη **$A \geq 5$** μπορεί να γραφτεί και ως **($A > 5$) and ($A = 5$)**
2. Η συνθήκη **$A \geq 5$** μπορεί να γραφτεί και ως **($A > 5$) or ($A = 5$)**
3. Η συνθήκη **$A = 5$** μπορεί να γραφτεί και ως **(not $A > 5$) and (not $A < 5$)**
4. Αν **$\alpha = 2$, $\beta = 2$, $\gamma = 4$** και **$\Delta = \text{True}$** , η τιμή της έκφρασης **($\beta * \gamma > \alpha + \beta$) and (not(Δ))** είναι **True**.
5. Οι λογικές τιμές είναι οι εξής: **or**, **and**, **not**.
6. Η έκφραση **$A < B$ and $A > B$** είναι πάντα **False**.
7. Η λογική πράξη **and** έχει μεγαλύτερη προτεραιότητα από τη πράξη **not**.
8. Μια **λογική έκφραση** περιέχει **υποχρεωτικά τουλάχιστον ένα** λογικό τελεστή.

Άσκηση 2 – Η τιμή A της βαθμολογίας σε ένα θέμα μπορεί να πάρει τις τιμές από 0 μέχρι και 20 (το μηδέν και το 20 είναι επιτρεπτές τιμές). Ποια από τις παρακάτω λογικές εκφράσεις ελέγχει αυτή τη συνθήκη;

- α. **$A \geq 0$ or $A \leq 20$**
- β. **$A > 0$ and $A < 20$**
- γ. **$A \geq 20$ and $A \leq 0$**
- δ. **$A \geq 0$ and $A \leq 20$**

Άσκηση 3 – Για να ελέγξουμε τη συνθήκη $0 \leq \beta \leq 100$ μπορούμε να γράψουμε στη python (επιλέξτε όλες τις σωστές απαντήσεις, είναι παραπάνω από μία):

- α. **$0 \leq \beta \leq 100$**
- β. **$\beta \geq 0$ and $\beta \leq 100$**
- γ. **$0 \leq \beta$ and $100 \geq \beta$**
- δ. **$\beta \leq 100$ and $\beta > 0$**
- ε. **not ($\beta < 0$ and $\beta > 100$)**
- στ. **not $\beta < 0$ and $\beta \leq 100$**