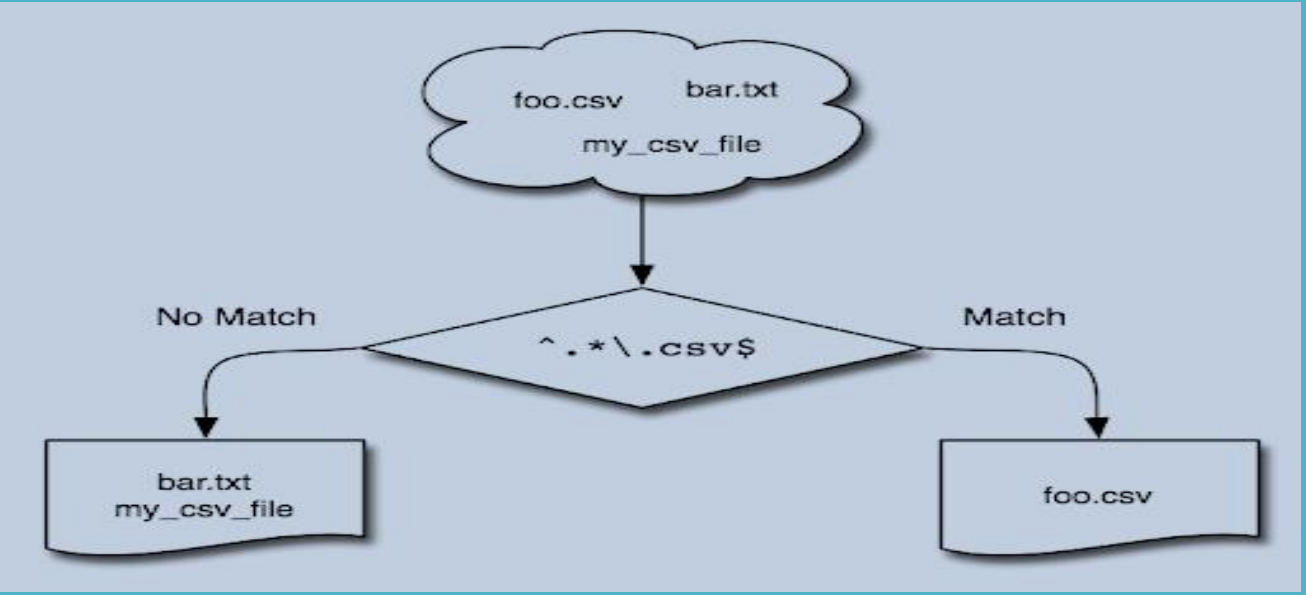
Κανονικές Εκφράσεις

Κανονικές Εκφράσεις

- Επιτρέπουν τον καθορισμό οποιουδήποτε συνδυασμού χαρακτήρων με συμβολικό και συνοπτικό τρόπο. Αποτελούνται από συνδυασμούς χαρακτήρων και μεταχαρακτήρων.

- Η βασική τους χρησιμότητα είναι η αναζήτηση αλφαριθμητικών σε αρχεία κειμένου. Η κάθε λέξη του αρχείου ελέγχεται εάν μπορεί να δημιουργηθεί από την κανονική έκφραση και εάν τα πράγματα είναι όντως έτσι, τότε επιστρέφεται στην έξοδο της εντολής.

- Αποτελούν αναπόσπαστο συστατικό των μεταγλωττιστών και των διερμηνευτών και σχετίζονται με ειδικές δομές που ονομάζονται μηχανές πεπερασμένων καταστάσεων.



Κανονικές Εκφράσεις

- Οι χαρακτήρες που χρησιμοποιούνται στις κανονικές εκφράσεις, περιλαμβάνουν το σύνολο των πεζών και κεφαλαίων γραμμάτων, τα ψηφία, και άλλους συχνά χρησιμοποιούμενους χαρακτήρες όπως είναι οι

~ ‘ ! @ # _ - = : ; /

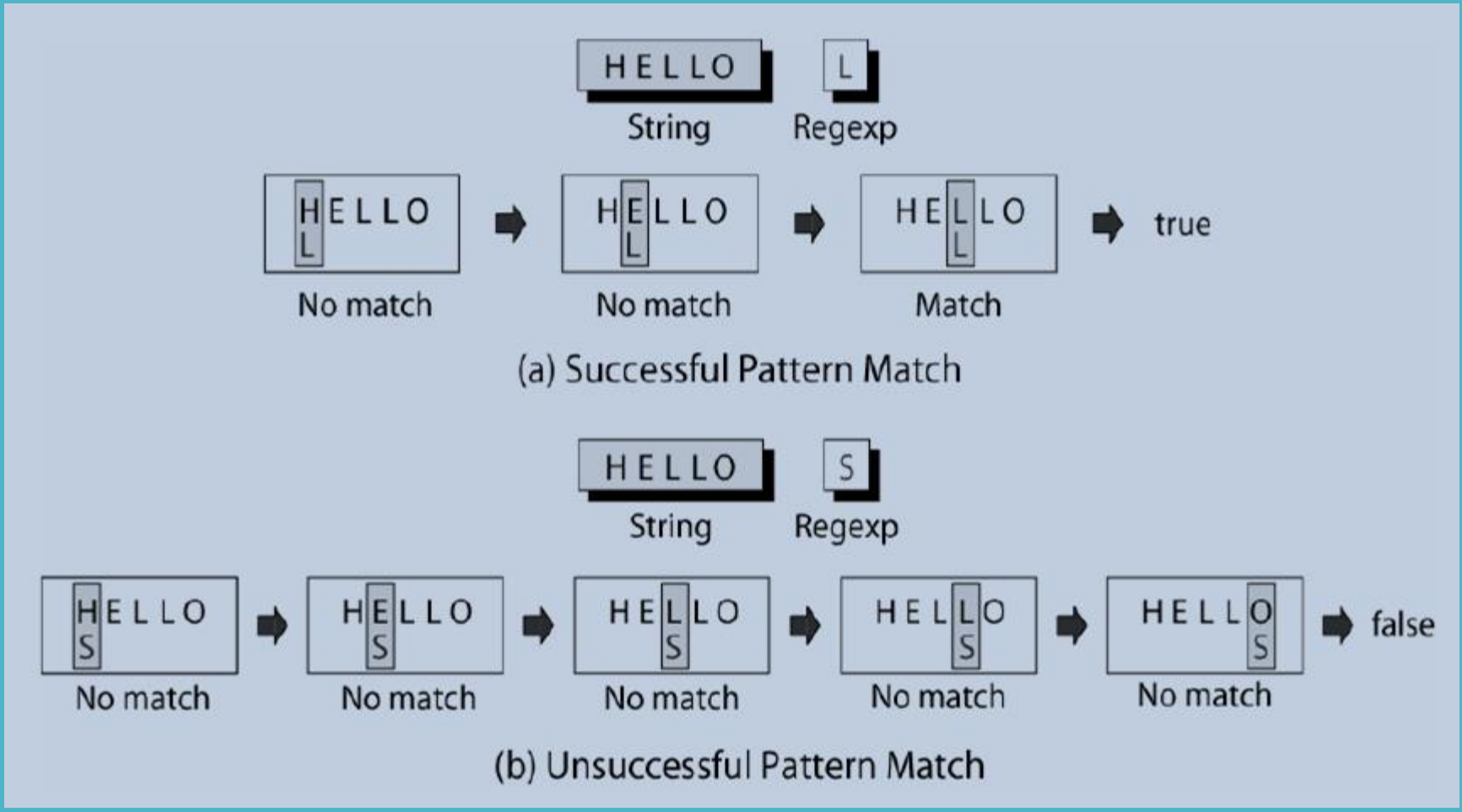
- Οι μεταχαρακτήρες που χρησιμοποιούνται στις κανονικές εκφράσεις είναι οι

\ . * [^ $]

Μία κανονική έκφραση αποτελείται από άτομα (απλούς χαρακτήρες, τελείες, anchors κλπ) που συνδυάζονται μεταξύ τους με τελεστές, με τον ίδιο ακριβώς τρόπο με τον οποίο στις αλγεβρικές εκφράσεις, τα μαθηματικά σύμβολα συνδυάζονται με τους αριθμητικούς τελεστές.

Κανονικές Εκφράσεις

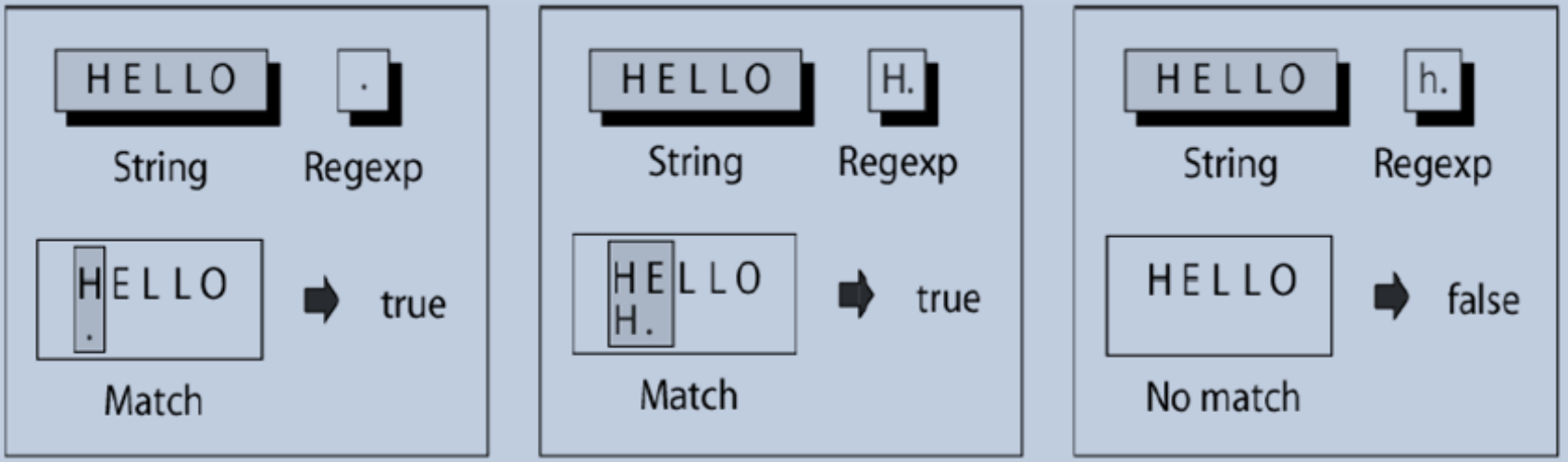
Τρόπος λειτουργίας κανονικών εκφράσεων



Κανονικές Εκφράσεις

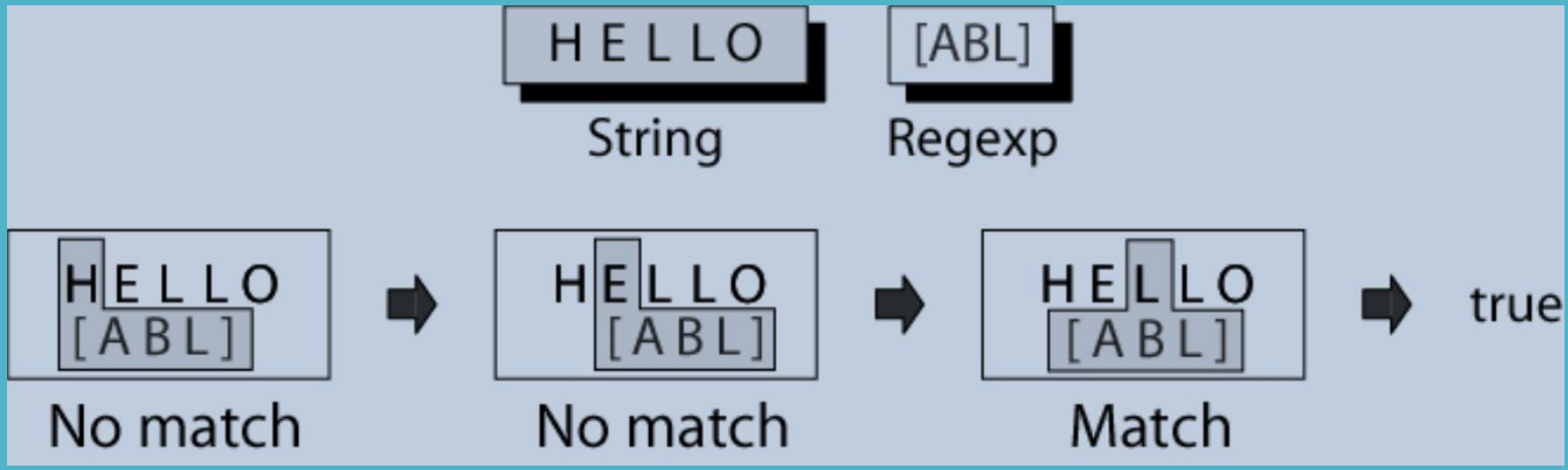
Τρόπος λειτουργίας κανονικών εκφράσεων

Μια τελεία (dot) αντιστοιχεί με οποιοδήποτε απλό χαρακτήρα εκτός από τον new line character (\n).



Κανονικές Εκφράσεις

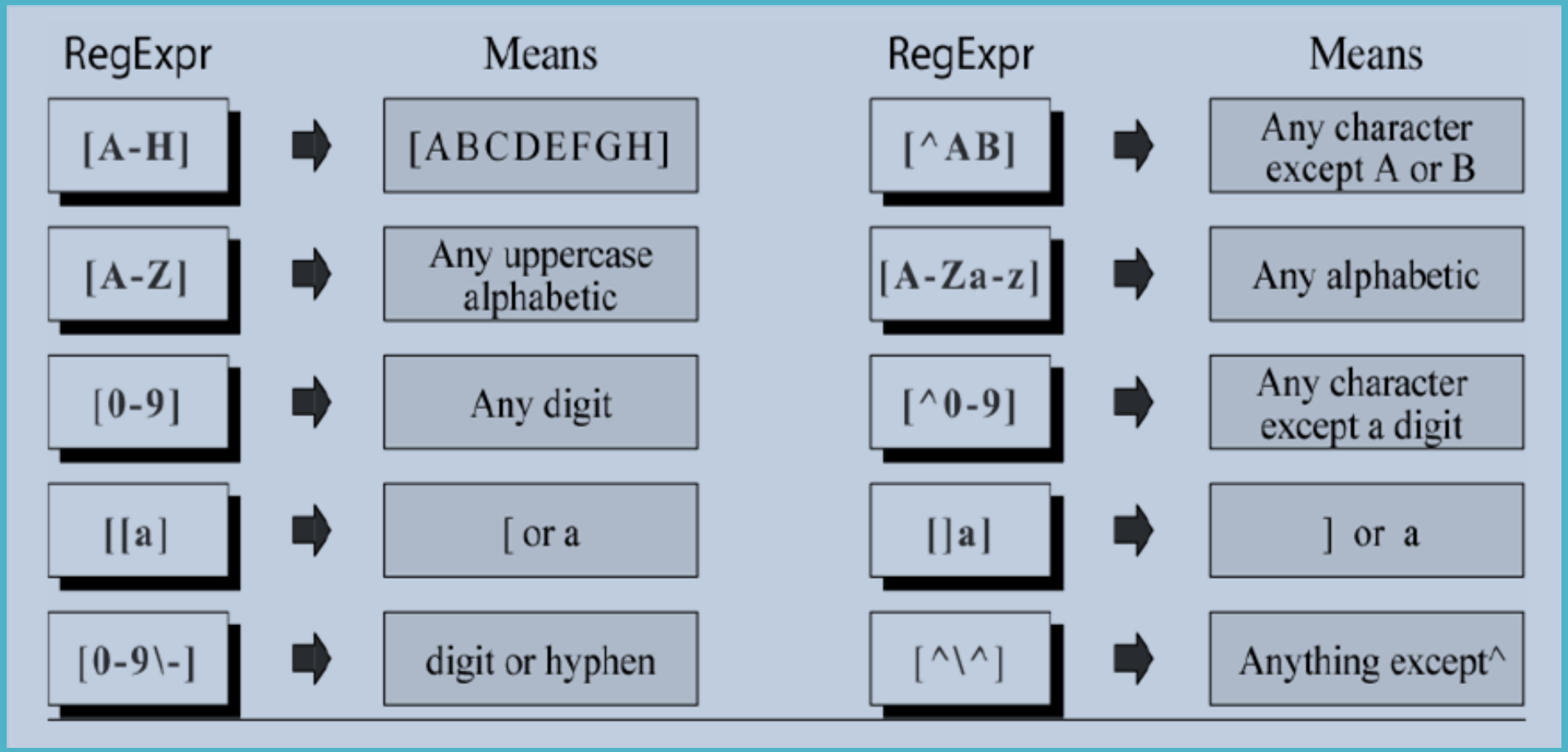
Τρόπος λειτουργίας κανονικών εκφράσεων



Οποιοσδήποτε χαρακτήρας από A έως E → [A-E]
Εναλλακτική γραφή → [ABCDE]
Οποιοσδήποτε χαρακτήρας από A έως Z και από a έως z → [A-Za-z]

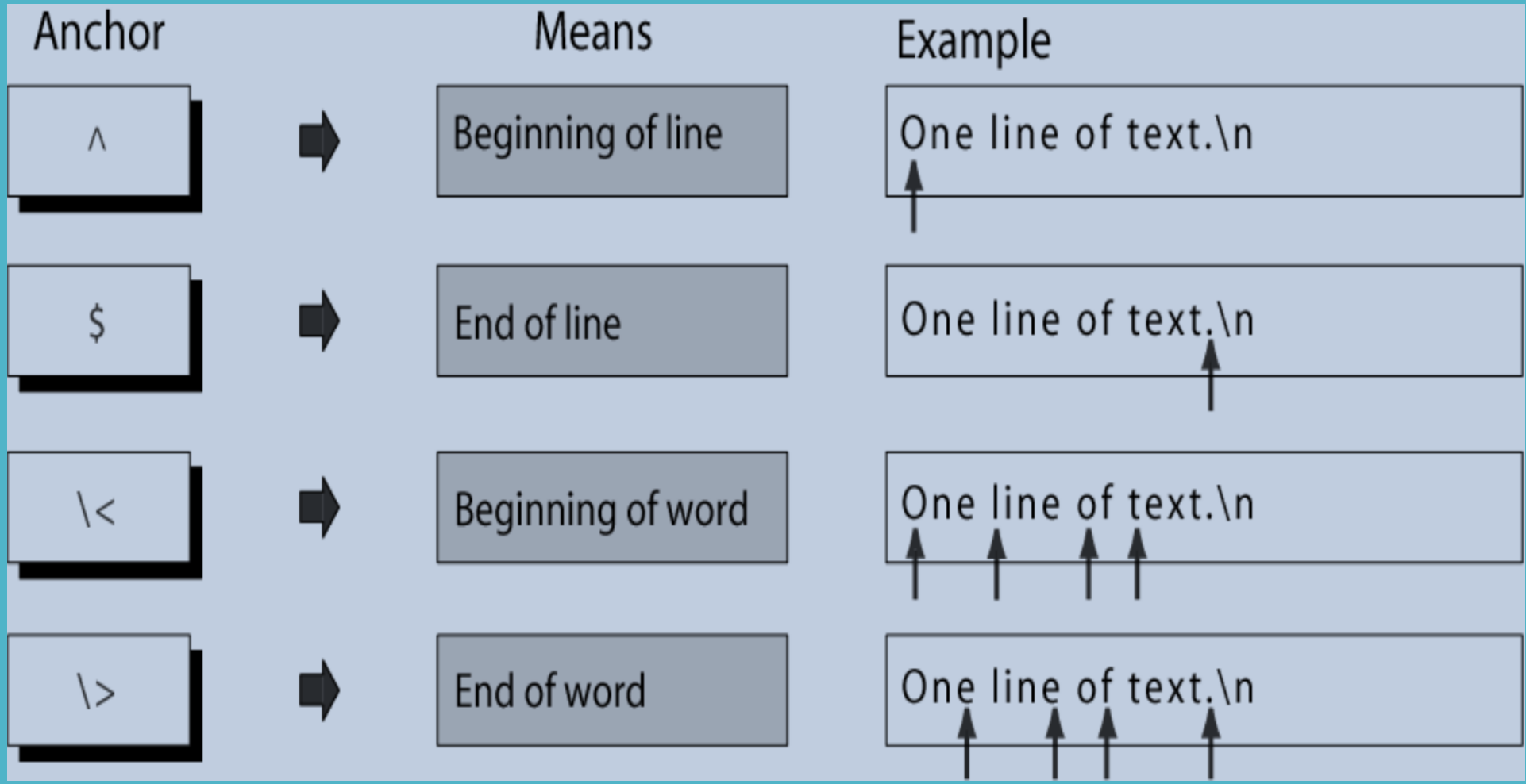
Κανονικές Εκφράσεις

Τρόπος λειτουργίας κανονικών εκφράσεων



Κανονικές Εκφράσεις

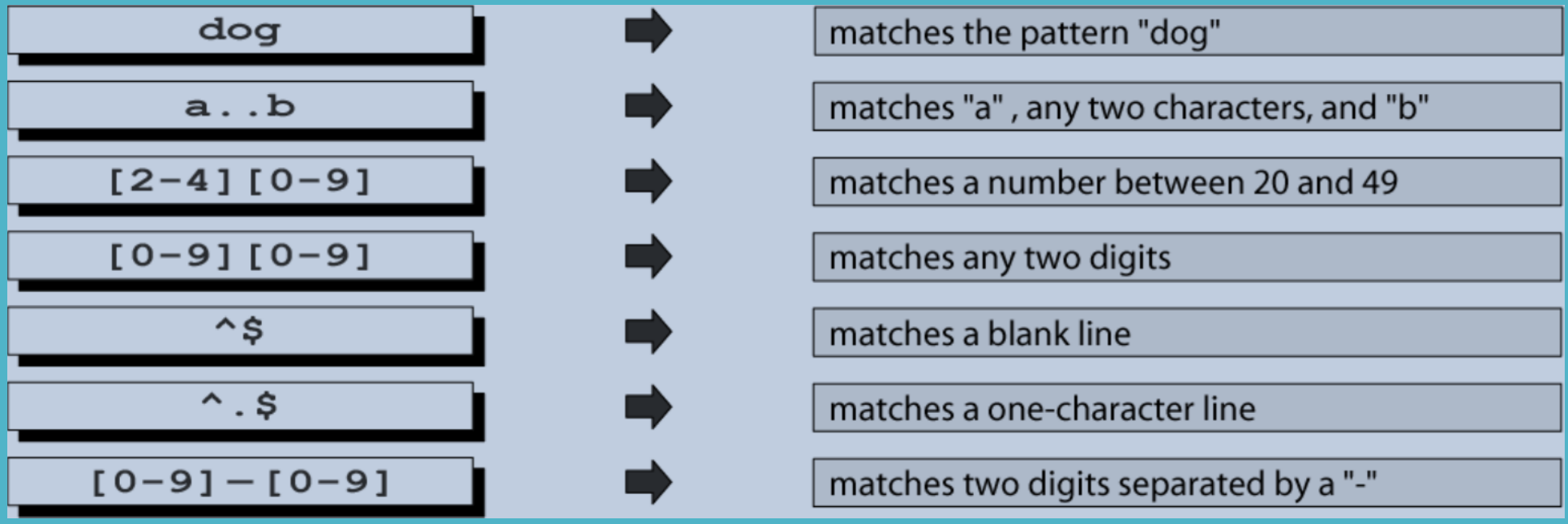
Τρόπος λειτουργίας κανονικών εκφράσεων



Κανονικές Εκφράσεις

Τελεστής ακολουθίας

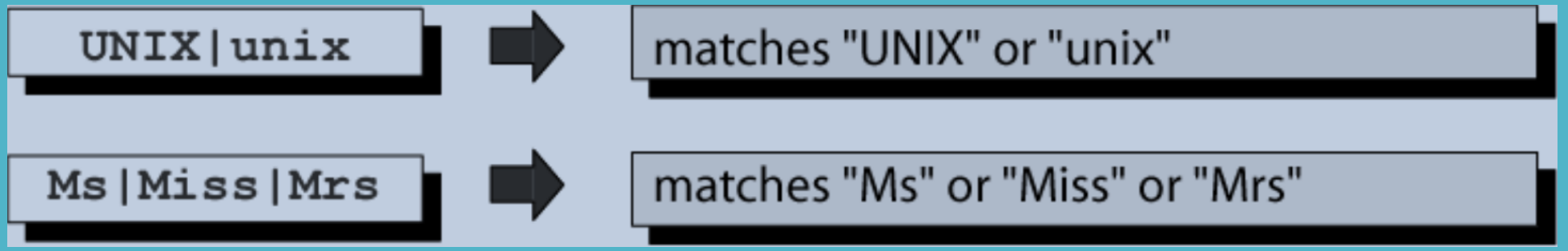
Ο **τελεστής ακολουθίας (sequence operator)** δεν υπάρχει ως σύμβολο. Αυτό σημαίνει ότι αν μια σειρά από atoms φαίνονται σε μια κανονική έκφραση, υποδηλώνεται η παρουσία ενός αόρατου sequence operator ανάμεσά τους.



Κανονικές Εκφράσεις

Τελεστής εναλλαγής

Ο **τελεστής εναλλαγής (alternation operator)** χρησιμοποιείται για να ορίσει μια ή περισσότερες εναλλακτικές περιπτώσεις.

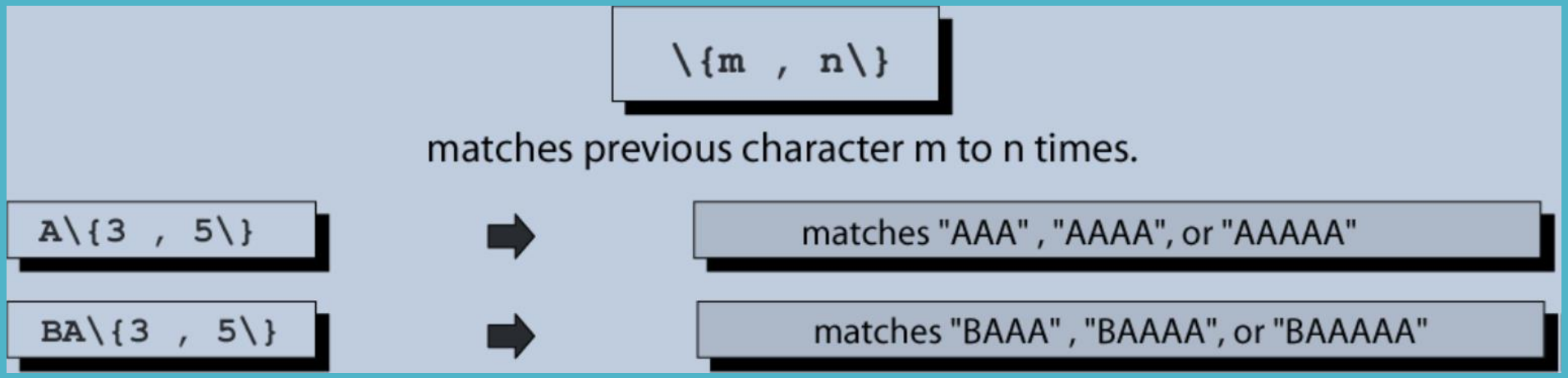


Κανονικές Εκφράσεις

Τελεστής επανάληψης

Ο **τελεστής επανάληψης (repetition operator)** καθορίζει ότι το atom ή η έκφραση που υπάρχει ακριβώς πριν από την επανάληψη μπορεί να επαναληφθεί.

m είναι ο ελάχιστος αριθμός επαναλήψεων.
n είναι ο μέγιστος αριθμός επαναλήψεων.



Κανονικές Εκφράσεις

Τελεστής επανάληψης

Formats

`\{m\}`	➡	matches previous atom exactly m times
`\{m, \}`	➡	matches previous atom m times or more
`\{, n\}`	➡	matches previous atom n times or less

Examples

`CA\{5\}`	➡	`CAAAAA`
`CA\{3,\}`	➡	`CAAA, CAAAA, CAAAAA, …`
`CA\{,2\}`	➡	`C, CA, CAA`

Κανονικές Εκφράσεις

Τελεστής επανάληψης

Formats

*	➡	special case: matches previous atom zero or more times
+	➡	special case: matches previous atom one or more times
?	➡	special case: matches previous atom 0 or one time only

Examples

BA*	➡	B, BA, BAA, BAAA, BAAAA, . . .
B.*	➡	B, BA . . . BZ, BAA . . . BZZ, BAAA . . . BZZZ, . . .
.*	➡	zero or more characters
.+	➡	one or more characters
[0-9]?	➡	zero or one digit

Κανονικές Εκφράσεις

Τελεστής ομαδοποίησης

Ο **τελεστής ομαδοποίησης (group operator)** είναι ένα ζεύγος παρενθέσεων που ανοίγουν και κλείνουν. Όταν μια ομάδα χαρακτήρων περικλείεται σε παρενθέσεις ο επόμενος τελεστής εφαρμόζεται σε όλη την ομάδα.

Η εντολή grep

(General Regular Expression Parser)

grep options RegEX [input file]

grep –e pattern ή –regexp patterns → εκτυπώνει τις γραμμές που περιέχουν το pattern
grep –f file → διαβάζει τα patterns από αρχείο
grep –i → αγνοεί τη διαφορά κεφαλαίων και μικρών γραμμάτων (ignore case)
grep –v → invert match (εκτυπώνει αυτά που δεν ταιριάζουν με το πρότυπο)
grep –w → εκτυπώνει τις γραμμές στις οποίες το ταίριασμα αφορά σε ολόκληρες λέξεις.
grep -n → μπροστά από κάθε γραμμή στην έξοδο εκτυπώνει τον αριθμό γραμμής
grep –r → διαβάζει όλα τα αρχεία ενός καταλόγου (recursively).

Η εντολή grep

ΠΑΡΑΔΕΙΓΜΑΤΑ

Εκτυπώνει τις γραμμές του αρχείου /etc/passwd που περιέχουν οπουδήποτε τη λέξη bash

```
grep bash /etc/passwd
```

Έξοδος

```
root:x:0:0:root:/root:/bin/bash
amarg:x:1000:1000:amarg:/home/amarg:/bin/bash
```

Εκτυπώνει τις γραμμές του αρχείου file.txt που ξεκινούν με τη λέξη linux

```
grep '^linux' file.txt
```

Εκτυπώνει τις γραμμές του αρχείου file.txt που τελειώνουν με τη λέξη linux

```
grep 'linux$' file.txt
```

Εκτυπώνει τις γραμμές του αρχείου file.txt που περιέχουν ΜΟΝΟ τη λέξη linux

```
grep '^linux$' file.txt
```

Η εντολή grep

ΠΑΡΑΔΕΙΓΜΑΤΑ

Εκτυπώνει τις λέξεις του αρχείου words που ξεκινούν με co περιέχουν τέσσερις οποιουσδήποτε χαρακτήρες και τελειώνουν με er

grep 'co....er' words

Εκτυπώνει τις λέξεις του αρχείου words που ξεκινούν με co περιέχουν δύο οποιουσδήποτε χαρακτήρες, ο τρίτος χαρακτήρας είναι r ή p ή q και τελειώνουν με es

grep 'co..[rpq]es' words

Εκτυπώνει τις λέξεις του αρχείου words που ξεκινούν με br, ο τρίτος χαρακτήρας μπορεί να είναι οποιοσδήποτε, ο τέταρτος χαρακτήρας ΔΕΝ μπορεί να είναι j, ακολουθεί αυθαίρετο πλήθος οποιωνδήποτε χαρακτήρων και τελειώνουν με x.

grep '^[A-Z] ' words

Εκτυπώνει τις λέξεις του αρχείου words που ξεκινούν με κεφαλαίο γράμμα

Η εντολή grep

ΠΑΡΑΔΕΙΓΜΑΤΑ

- Όλες οι γραμμές που δεν ξεκινούν από κεφαλαίο αγγλικό χαρακτήρα :
  - **grep '^[^A-Z]' file**
- Όλες οι γραμμές που περιέχουν !,& ,* :
  - **grep '([\!*\&])' file**
- Όλες οι γραμμές που περιέχουν την τιμή $1.99 :
  - **grep '\$1\.99' file**
- Όλες οι γραμμές με μήκος 2 χαρακτήρες :
  - **grep '^..$'file ή ^.\{2\}$**

Η εντολή grep

ΠΑΡΑΔΕΙΓΜΑΤΑ

- Όλες οι γραμμές που έχουν μήκος ακριβώς 17 χαρακτήρες:
  - **grep '^.\{17\}$' file**
- Όλες οι γραμμές που έχουν μήκος τουλάχιστον 25 χαρακτήρες:
  - **grep '^.\{25,\}$'**
- Όλες οι γραμμές που δεν έχουν μήκος 3 χαρακτήρες:
  - **grep -v '^...$'file**

Η εντολή grep

ΠΑΡΑΔΕΙΓΜΑΤΑ

- Όλες οι γραμμές που ξεκινούν με * :
  - **^***
- Όλες οι γραμμές που δεν περιέχουν αριθμούς :
  - **^[^0-9]*$**
- Όλες οι γραμμές που περιέχουν τα έτη 1991 έως 1995 :
  - **199[1-5]**
- Οποιαδήποτε ακολουθία χαρακτήρων δεν περιέχει ψηφία:
  - **[A-Za-z][A-Za-z]***

Η εντολή grep

ΠΑΡΑΔΕΙΓΜΑΤΑ

- Οποιοσδήποτε προσημασμένος ακέραιος :
  - **[+\-][0-9][0-9]***
- Οποιαδήποτε ακολουθία χαρακτήρων :
  - **.* (ιδιωματισμός!)**
- Οποιοδήποτε αναγνωριστικό (identifier) :
  - **[a-zA-Z_][a-zA-Z_0-9]***
- Οποιοσδήποτε πραγματικός αριθμός χωρίς πρόσημο:
  - **[0-9]+[.][0-9]+**
  - **[0-9]*[.][0-9]***

Η εντολή AWK

(Aho, Weinberger, Kerningham)

Γλώσσα ταυτοποίησης προτύπων που αναλύει αρχεία κειμένου που δέχεται στην είσοδό της και επιστρέφει τις εγγραφές που ταιριάζουν σε ένα πρότυπο.

Χρησιμοποιεί το συντακτικό της γλώσσας C και περιέχει δεσμευμένες λέξεις, σταθερές, μεταβλητές, εντολές, συναρτήσεις και πίνακες.

Κάθε γραμμή του αρχείου εισόδου μπορεί να διαχωριστεί σε πεδία που μπορούν να διαχειριστούν ανεξάρτητα ορίζοντας τον κατάλληλο διαχωριστή. Ο default separator είναι το κενό (space) ενώ για να οριστεί άλλος χαρακτήρας χρησιμοποιείται ο διακόπτης –F.

Η αναζήτηση προτύπων στο αρχείο εισόδου γίνεται με τη βοήθεια κανονικών εκφράσεων.

$ awk –F [field separator] ‘ { awk program } ’ [input_file]

$ awk –F [field separator] -f [awk script] [input_file]

Η εντολή AWK

(Aho, Weinberger, Kerningham)

Μεταβλητές

Μεταβλητές γενικού τύπου π.χ.

title="Number of students" , no=100, weight=77.9

ΕΙΔΙΚΕΣ (δεσμευμένες) ΜΕΤΑΒΛΗΤΕΣ

$n	n-οστό πεδίο στη γραμμή, $0 – ολόκληρη η γραμμή	
FS	διαχωριστής πεδίων (εξ ορισμού κενό και tab)	
OFS	διαχωριστής πεδίων αρχείου εξόδου (εξ ορισμού κενό)	
NR	αριθμός εγγραφής (γραμμής)	} Αρχικοποιούνται για κάθε γραμμή
NF	πλήθος πεδίων της γραμμής	
FILENAME	όνομα αρχείου εισόδου	
RS	διαχωριστής εγγραφών αρχείου εισόδου (εξ ορισμού new line)	
ORS	διαχωριστής εγγραφών αρχείου εξόδου (εξ ορισμού new line)	

Η εντολή AWK

Τα προγράμματα awk διαιρούνται σε τρία κύρια blocks:

BEGIN block, block επεξεργασίας, END block

Εκτός και αν ορίζεται ρητά, όλες οι εντολές εμφανίζονται στο block επεξεργασίας.

Οποιοδήποτε από τα 3 τμήματα μπορεί να παραλείπεται.

Οι εντολές διαιρούνται σε δύο τμήματα:

- Ένα κριτήριο επιλογής, που αναφέρει στην awk τι πρέπει να ταιριάξει, και
- Μια αντίστοιχη ενέργεια που αναφέρει στην awk τι θα κάνει όταν βρεθεί μια γραμμή που ταιριάζει με το συγκεκριμένο κριτήριο επιλογής.

Το τμήμα ενεργειών της εντολής βρίσκεται σε { } και μπορεί να περιέχει πολλές εντολές.

Οι εντολές που διαθέτουν κριτήριο επιλογής εφαρμόζονται σε κάθε γραμμή που αντιστοιχεί ή καθιστά αληθές το κριτήριο, ανάλογα αν αυτό είναι μια κανονική έκφραση ή μια λογική έκφραση.

Οι εντολές που δεν έχουν κριτήρια επιλογής εφαρμόζονται σε κάθε γραμμή του αρχείου εισόδου.

Η εντολή AWK

Αντιστοίχιση προτύπων

- Κάθε γραμμή πριν επεξεργαστεί μπορεί να αντιστοιχηθεί (να ταιριάξει με ένα πρότυπο). Το πρότυπο περικλείεται σε / /.
- Format :

  /pattern/ { action } εκτελείται αν η γραμμή περιέχει το πρότυπο

  !/pattern/ { action } εκτελείται αν η γραμμή ΔΕΝ περιέχει το πρότυπο
- παραδείγματα:

```
/^$/              { print "This line is blank " }
/text/            { print "This line includes text" }
/[0-9]+$/         { print "Integer:", $0 }
/[a-z]+/          { print "String:", $0 }
/^[A-Z]/          { print "start with an uppercase letter" }
```

Η εντολή AWK

```
$cat names.txt
Per Wisung 021-336699
Jan Medin 021-332211
Hans Persson 021 112233
Göran Persson 021-336666
```

```
$awk ´{print "Name: ", $1,$2, "Telephone:", $3}
´ names.txt
```

```
Name: Per Wisung        Telephone: 021-336699
Name: Jan Medin         Telephone: 021-332211
Name: Hans Persson      Telephone: 021
Name: Göran Persson     Telephone: 021-336666
```

Η εντολή AWK

```
$ cat sales
John Anderson,12,23,7,42
Joe Turner,10,25,15,50
Susan Greco,15,13,18,46
Bob Burmeister,8,21,17,46

$ awk -F, '{print $1,$5}' sales

John Anderson 42
Joe Turner 50
Susan Greco 46
Bob Burmeister 46
```

Η εντολή AWK

```
$ cat emp.data
John Anderson:sales:1980
Joe Turner:marketing:1982
Susan Greco:sales:1985
Ike Turner:pr:1988
Bob Burmeister:accounting:1991

$ awk -F: '$3 == 1980,$3 == 1985 {print $1, $3}' emp.data
John Anderson 1980
Joe Turner 1982
Susan Greco 1985
```

Η εντολή AWK

Συντακτικό των εντολών - παραδείγματα

```
if ( found == true )                    #   if (expr)
      print "Found"                     #       {action1}
else                                    #   else
      print "Not found"                 #       {action2}

while ( i <= 100)                       #   while (cond.)
      { i=i+1; print i }                #       {action}

do                                      #   do
      { i=i+1; print i }                #       {action}
while ( i<100)                          #   while (cond.)

for (i=1; i<10; i++ ) {                 #    for (set; test;increment)
      i2= i*i                           #       {action}
      printf(" %d*%d = %d\n", i, i, i2)
}
```

Η εντολή AWK

Περιεχόμενα του αρχείου /etc/passwd

```
root:x:0:0:Super-User:/:/bin/csh
sysadm:x:0:0:System V Administration:/usr/admin:/bin/sh
cmwlogin:x:0:994:CMW Login UserID:/usr/CMW:/sbin/csh
diag:x:0:996:Hardware Diagnostics:/usr/diags:/bin/csh
daemon:x:1:1:daemons:/:/dev/null
bin:x:2:2:System Tools Owner:/bin:/dev/null
uucp:x:3:5:UUCP Owner:/usr/lib/uucp:/bin/csh
sys:x:4:0:System Activity Owner:/var/adm:/bin/sh
adm:x:5:3:Accounting Files Owner:/var/adm:/bin/sh
lp:x:9:9:Print Spooler Owner:/var/spool/lp:/bin/sh
auditor:x:11:0:Audit Activity Owner:/auditor:/bin/sh
dbadmin:x:12:0:Security Database Owner:/dbadmin:/bin/sh
guest:x:998:998:Guest Account:/usr/people/guest:/bin/csh
```

Η εντολή AWK

- Δημιουργία λίστας με τους χρήστες του συστήματος με μήκος login name έως 4 χαρακτήρες

```
$ grep  "^[^:]\{1,4\}:" /etc/passwd | awk -F: '{print $5}'
Super-User
Hardware Diagnostics
System Tools Owner
UUCP Owner
System Activity Owner
Accounting Files Owner
Print Spooler Owner
```