Scuola universitaria professionale della Svizzera italiana
Dipartimento tecnologie innovative
Istituto sistemi informativi e networking

SUPSI

Ambienti Operativi: Espressioni Regolari

Amos Brocco, Ricercatore, DTI / ISIN

Vi ricordate'grep' ?

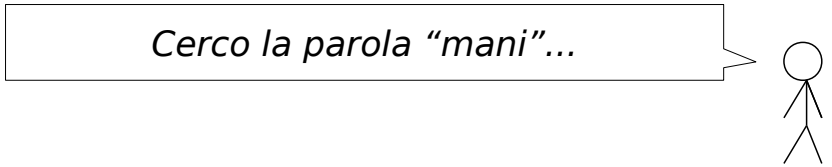
- Abbiamo visto che il comando **grep** può essere utilizzato per “estrarre” le righe di un testo o di un input (con una pipe) che contengono una determinata parola

[X] bash

```
utente@host:~/Documenti/Privato$ grep mani testo.txt
gomiti sulle sue ginocchia e con la faccia appoggiata tra le mani, stavo a sentire.
Gli Egiziani trasmisero ai Romani le preparazioni che permettevano di trasformar le
ma domani possiamo esser poveri. E non si misero in cammino a mani vuote.
```

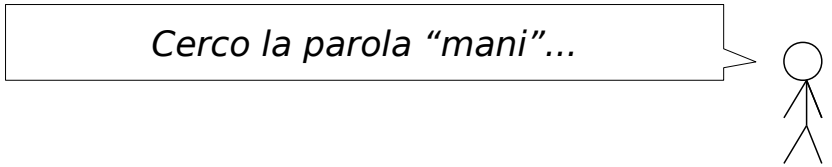
... e se volessimo visualizzare tutte le righe che contengono solo la parola “mani”
(e non Romani o domani) ?

Ricerca di una stringa di testo...



Il 16 febbraio del 1951, sedevo su un panchettino di legno, ai suoi piedi, puntavo i gomiti sulle sue ginocchia e con la faccia appoggiata tra le mani, stavo a sentire. Gli Egiziani trasmisero ai Romani le preparazioni che permettevano di trasformar le fibre vegetali del papiro in superfici pulite, bianche, pieghevoli. Oggi siamo ricchi, ma domani possiamo esser poveri. E non si misero in cammino a mani vuote.

Ricerca di una stringa di testo...

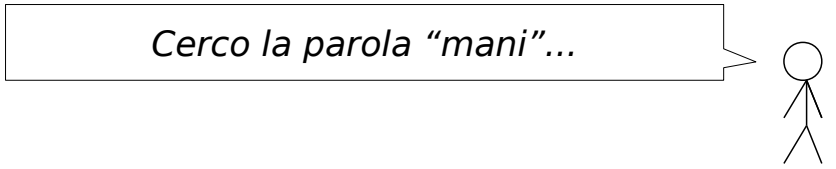


Il 16 febbraio del 1951, sedevo su un panchettino di legno, ai suoi piedi, puntavo i gomiti sulle sue ginocchia e con la faccia appoggiata tra le mani, stavo a sentire. Gli Egiziani trasmisero ai Romani le preparazioni che permettevano di trasformar le fibre vegetali del papiro in superfici pulite, bianche, pieghevoli. Oggi siamo ricchi, ma domani possiamo esser poveri. E non si misero in cammino a mani vuote.

```
T : String := "...testo...";
L : Integer := T'Length;

for I in Integer range 0 .. L-4 loop
         if T(I) = 'm' and
            T(I+1) = 'a' and
            T(I+2) = 'n' and
            T(I+3) = 'i' then
                  -- Trovato!
         end if;
end loop;
```

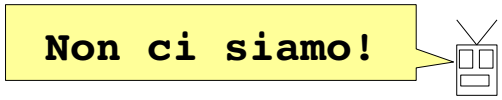
Ricerca di una stringa di testo...



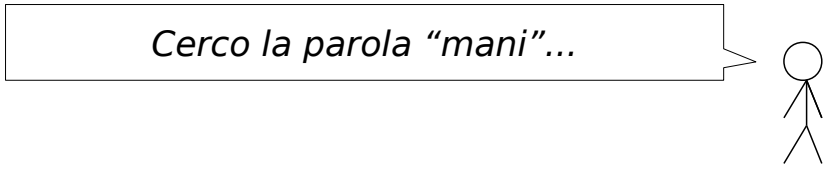
Il 16 febbraio del 1951, sedevo su un panchettino di legno, ai suoi piedi, puntavo i gomiti sulle sue ginocchia e con la faccia appoggiata tra le **mani**, stavo a sentire. Gli Egiziani trasmisero ai Ro**mani** le preparazioni che permettevano di trasformar le fibre vegetali del papiro in superfici pulite, bianche, pieghevoli. Oggi siamo ricchi, ma do**mani** possiamo esser poveri. E non si misero in cammino a **mani** vuote.

```
T : String := "...testo...";
L : Integer := T'Length;

for I in Integer range 0 .. L-4 loop
         if T(I) = 'm' and
            T(I+1) = 'a' and
            T(I+2) = 'n' and
            T(I+3) = 'i' then
                   -- Trovato!
         end if;
end loop;
```



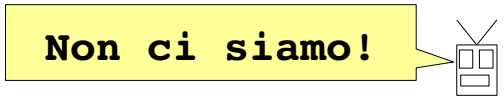
Ricerca di una stringa di testo...



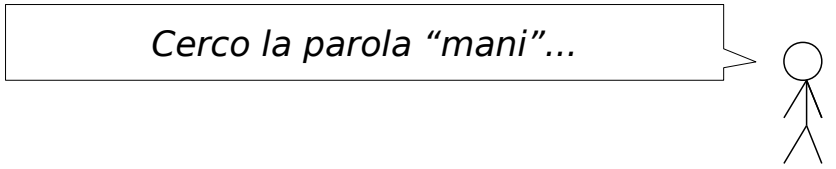
Il 16 febbraio del 1951, sedevo su un panchettino di legno, ai suoi piedi, puntavo i gomiti sulle sue ginocchia e con la faccia appoggiata tra le mani, stavo a sentire. Gli Egiziani trasmisero ai Romani le preparazioni che permettevano di trasformar le fibre vegetali del papiro in superfici pulite, bianche, pieghevoli. Oggi siamo ricchi, ma domani possiamo esser poveri. E non si misero in cammino a **mani** vuote.

```
T : String := "...testo...";
L : Integer := T'Length;

for I in Integer range 0 .. L-6 loop
         if T(I) = ' ' and
            T(I+1) = 'm' then
            T(I+2) = 'a' and
            T(I+3) = 'n' and
            T(I+4) = 'i' and
            T(I+5) = ' ' then
                   -- Trovato!
         end if;
end loop;
```



Ricerca di una stringa di testo...

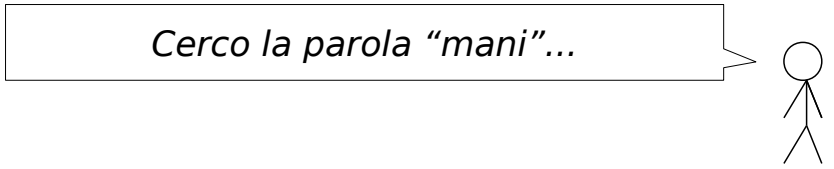


Il 16 febbraio del 1951, sedevo su un panchettino di legno, ai suoi piedi, puntavo i gomiti sulle sue ginocchia e con la faccia appoggiata tra le **mani**, stavo a sentire. Gli Egiziani trasmisero ai Romani le preparazioni che permettevano di trasformar le fibre vegetali del papiro in superfici pulite, bianche, pieghevoli. Oggi siamo ricchi, ma domani possiamo esser poveri. E non si misero in cammino a **mani** vuote.

```
T : String := "...testo...";
L : Integer := T'Length;

for I in Integer range 0 .. L-6 loop
          if T(I) = ' ' and
             T(I+1) = 'm' then
             T(I+2) = 'a' and
             T(I+3) = 'n' and
             T(I+4) = 'i' and
             (T(I+5) = ' ' or T(I+5) = ',') then
                    -- Trovato!
          end if;
end loop;
```

Ricerca di una stringa di testo...



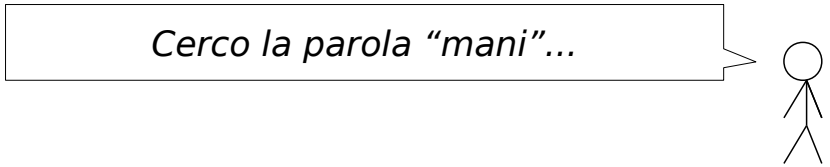
Il 16 febbraio del 1951, sedevo su un panchettino di legno, ai suoi piedi, puntavo i gomiti sulle sue ginocchia e con la faccia appoggiata tra le **mani**, stavo a sentire. Gli Egiziani trasmisero ai Romani le preparazioni che permettevano di trasformar le fibre vegetali del papiro in superfici pulite, bianche, pieghevoli. Oggi siamo ricchi, ma domani possiamo esser poveri. E non si misero in cammino a **mani** vuote.

```
T : String := "...testo...";
L : Integer := T'Length;

for I in Integer range 0 .. L-6 loop
          if T(I) = ' ' and
             T(I+1) = 'm' then
             T(I+2) = 'a' and
             T(I+3) = 'n' and
             T(I+4) = 'i' and
             (T(I+5) = ' ' or T(I+5) = ',') then
                    -- Trovato!
          end if;
end loop;
```

E se invece della virgola avessi '.' ?

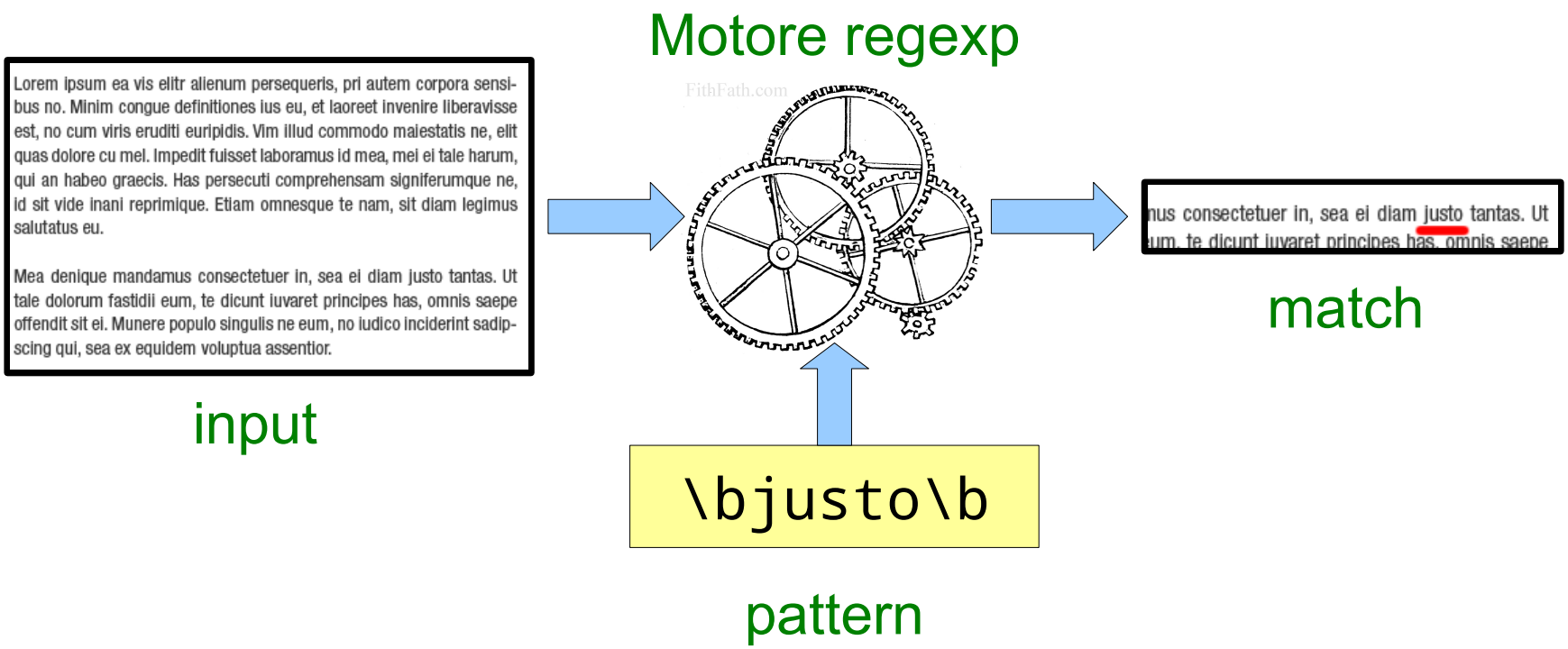
Espressione regolare




```
\bmani\b
```

Cos'è un'espressione regolare?

- È uno strumento per **ricercare sequenze di caratteri** (stringhe) **all'interno di un testo** (o flusso di caratteri)
- È un pattern o template per configurare un “motore delle espressioni regolari”

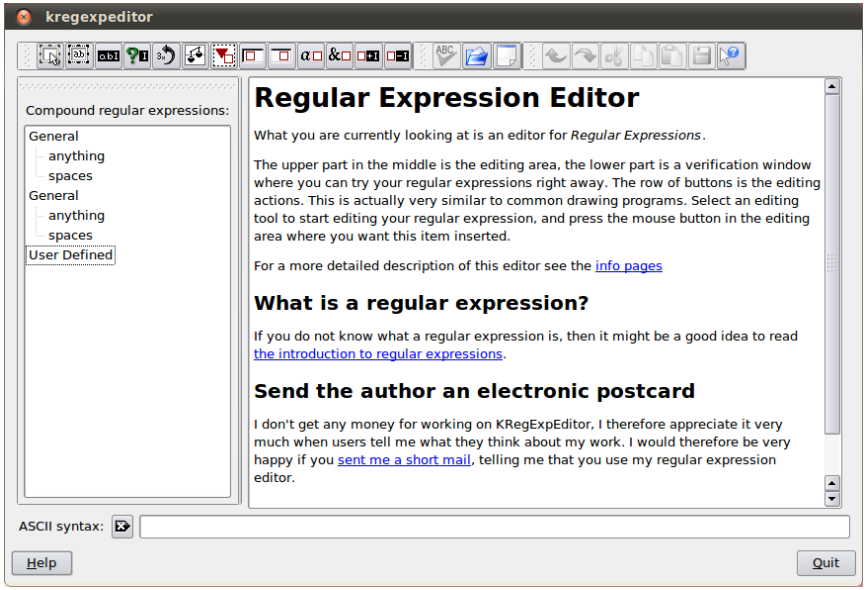


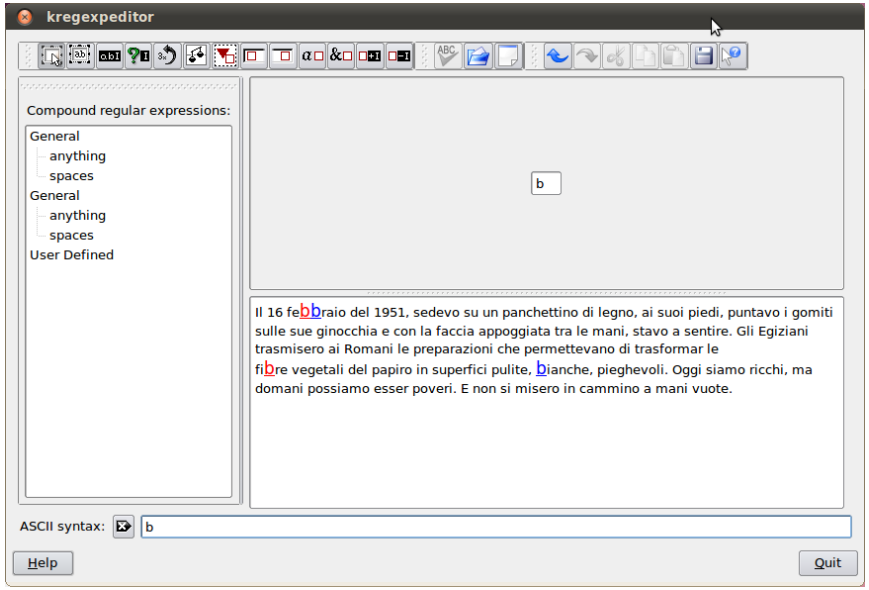
Qualche definizione

- **regexp**
  - **Reg**ular **exp**ression, espressione regolare
- **pattern**
  - Schema che descrive, utilizzando il linguaggio delle espressioni regolari, la stringa da ricercare
  - Solitamente si dice “Espressione regolare” per indicare il pattern
    - Un'espressione regolare è costituita da più **termini**, che determinano “cosa” vogliamo cercare
- **pattern matching**
  - Operazione per verificare se una stringa corrisponde a (o contiene) un pattern definito
- **pattern substitution**
  - Operazione di sostituzione di una stringa corrispondente a un pattern con un'altra stringa

kregexpeditor

- Tool grafico per sperimentare con le espressioni regolari





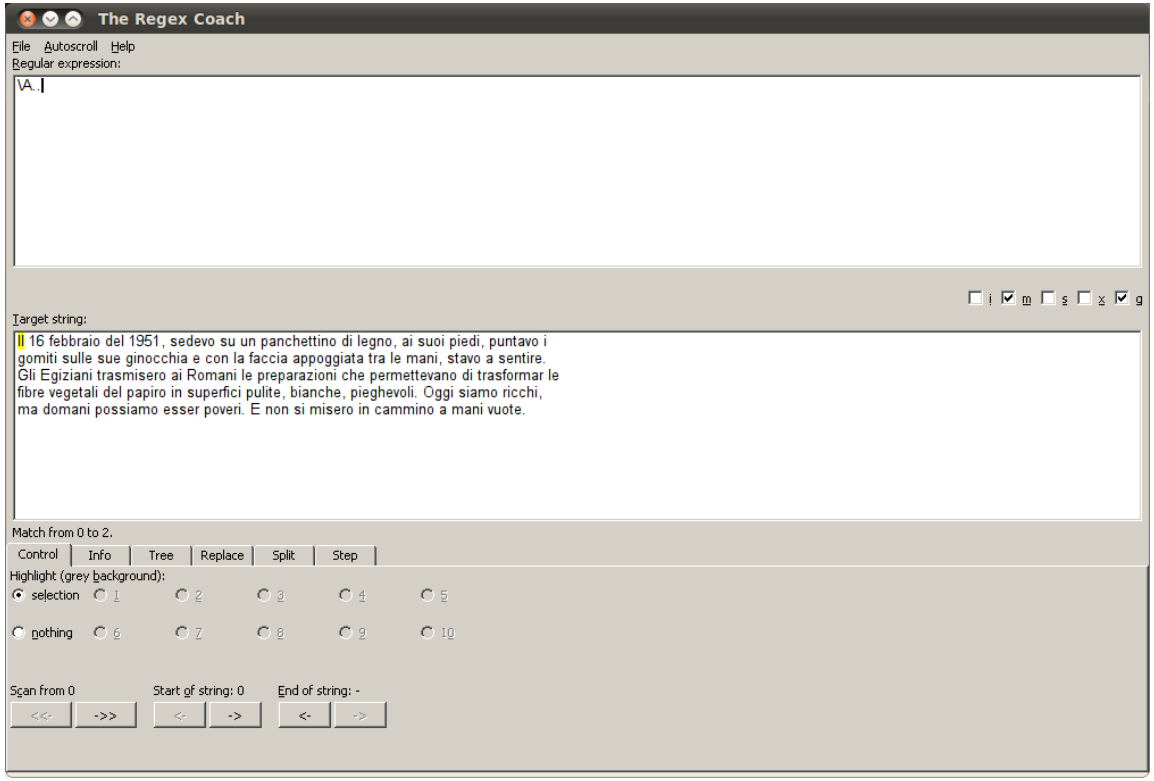
- Attenzione:
  - Il motore di kregexpeditor non è molto robusto e può generare errori con espressioni complesse
  - Non tutti i caratteri speciali sono supportati
  - le espressioni regolari riconosciute da kregexpeditor sono “case-insensitive” !

kregexpeditor

- Scaricate l'archivio **kregexpeditor.tar.gz** da Moodle, e il file **testo.txt**
- Estraete il contenuto dell'archivio nella vostra directory personale
- Eseguite il file kregexpeditor.sh
- Incollate il contenuto di **testo.txt** nel pannello inferiore

The Regex Coach

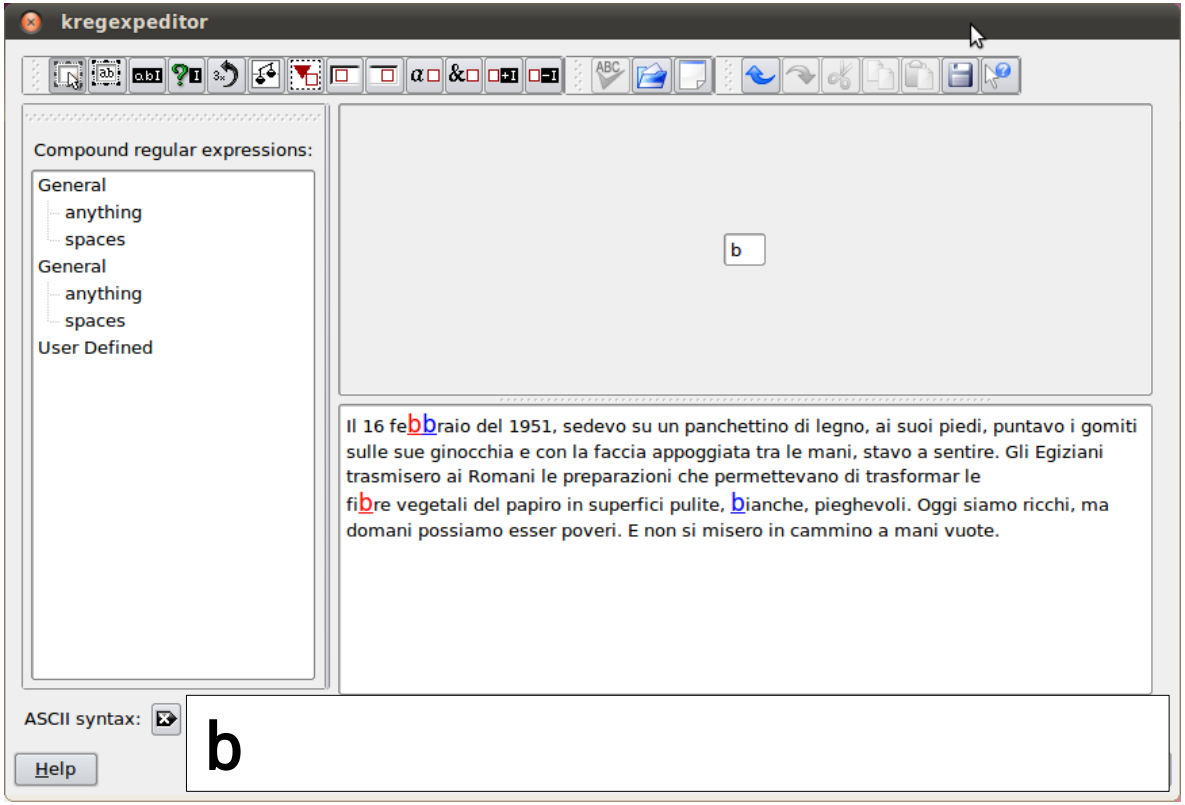
- http://weitz.de/regex-coach/
  - Download: http://weitz.de/files/regex-coach.exe
- Per Windows (funziona anche su GNU/Linux con Wine)



Pattern semplici: singoli caratteri

Cerco il carattere 'b'

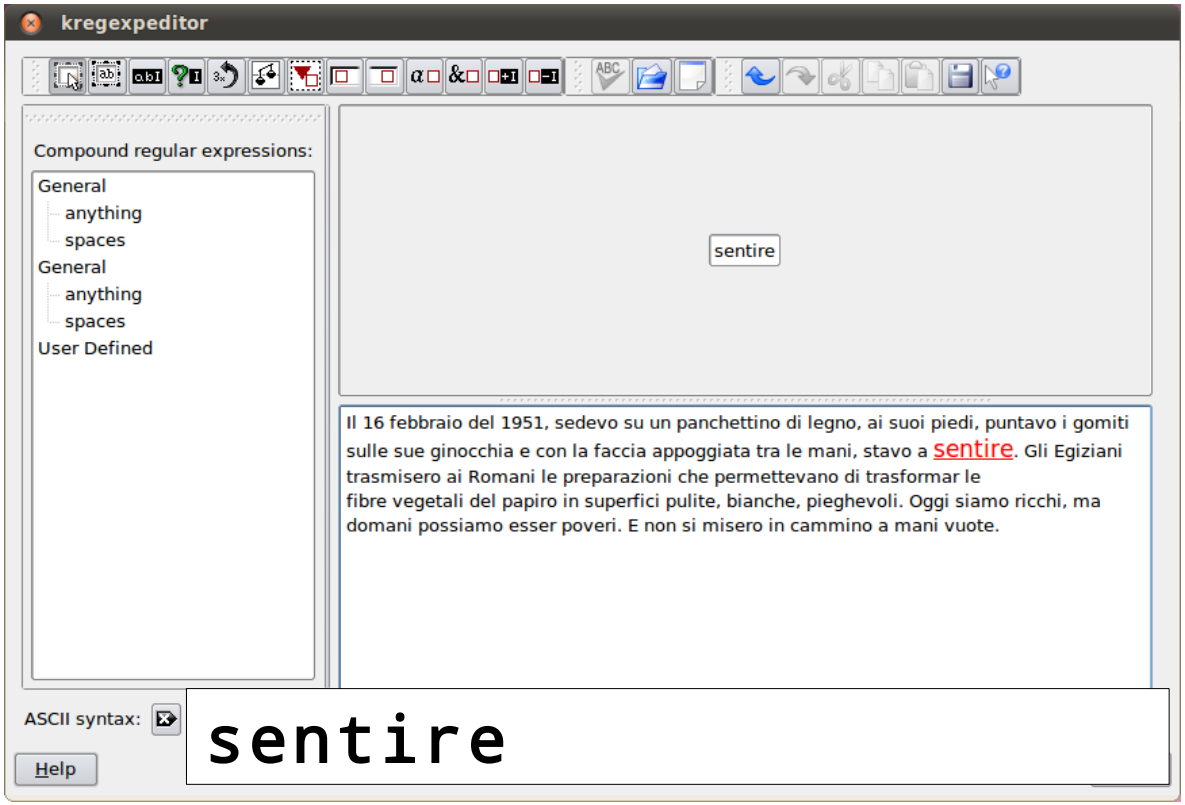
L'espressione regolare è composta dal singolo carattere da ricercare



Pattern semplici: sequenze di più caratteri (stringhe)

Cerco la parola 'sentire'

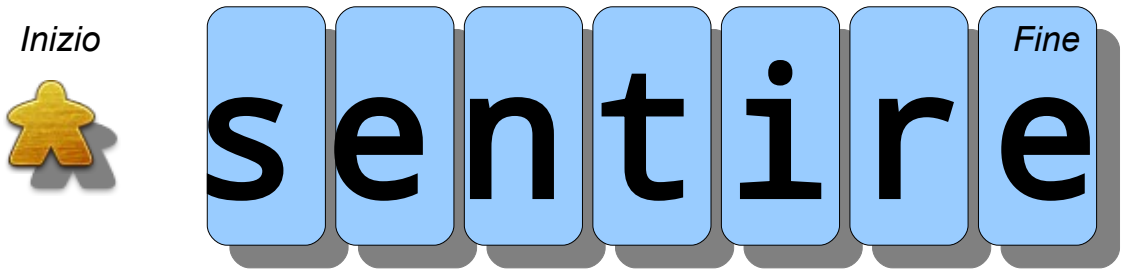
`L'espressione regolare è composta dalla sequenza di caratteri da ricercare`



Come funziona il matching

- Immaginiamo un gioco da tavolo,
  - Ogni termine dell'espressione regolare corrisponde a una casella
  - Invece di un dado, per spostarmi leggo il testo di input carattere per carattere:
    - se il carattere letto corrisponde alla casella successiva mi sposto in avanti
    - altrimenti torno all'inizio

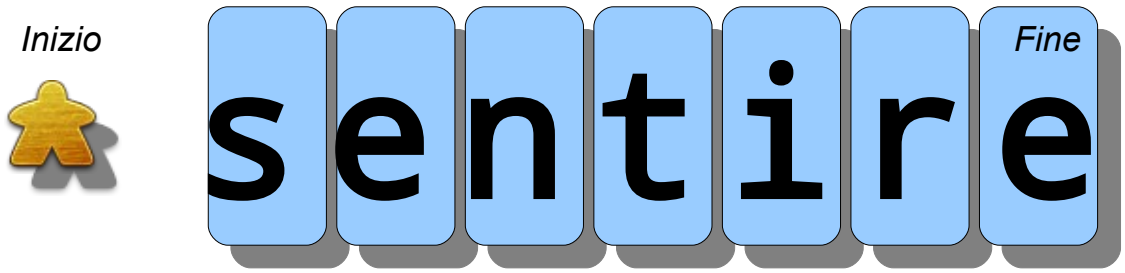
- Lo scopo del gioco è ottenere una corrispondenza arrivando alla casella “Fine”



Come funziona il matching

a vo a sentire. Gli Egiziani trasmisero ai Romani le

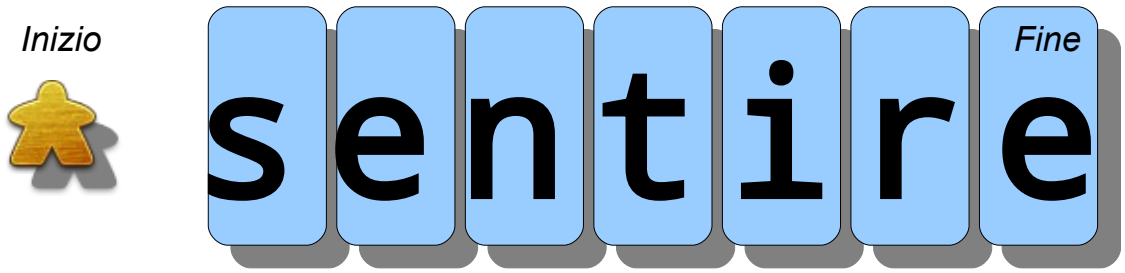
Carattere corrente



Come funziona il matching

V o a sentire. Gli Egiziani trasmisero ai Romani le

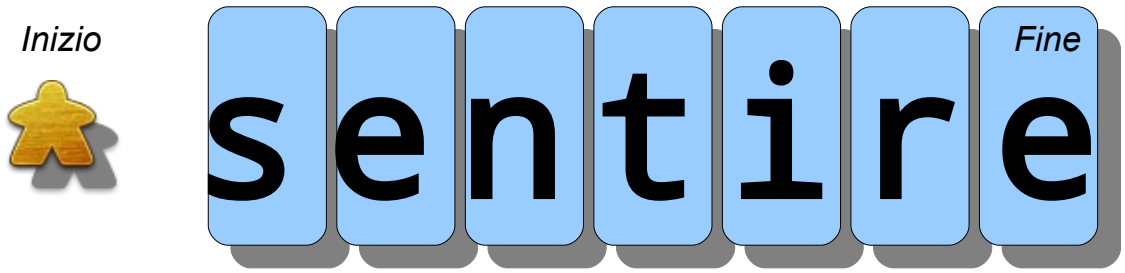
Carattere corrente



Come funziona il matching

O a sentire. Gli Egiziani trasmisero ai Romani le p

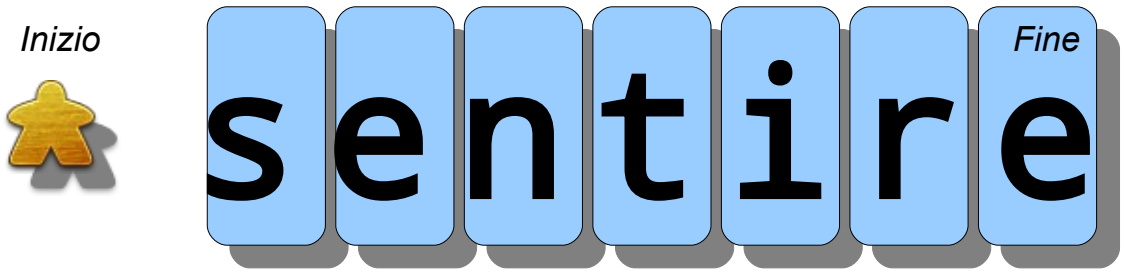
Carattere corrente



Come funziona il matching

a sentire. Gli Egiziani trasmisero ai Romani le pr

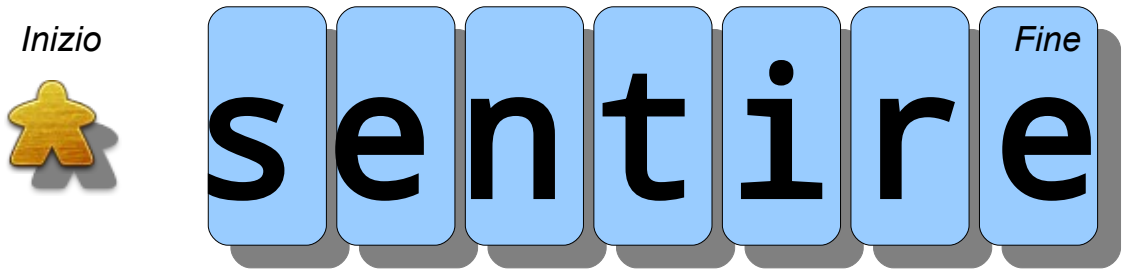
Carattere corrente



Come funziona il matching

a sentire. Gli Egiziani trasmisero ai Romani le pre

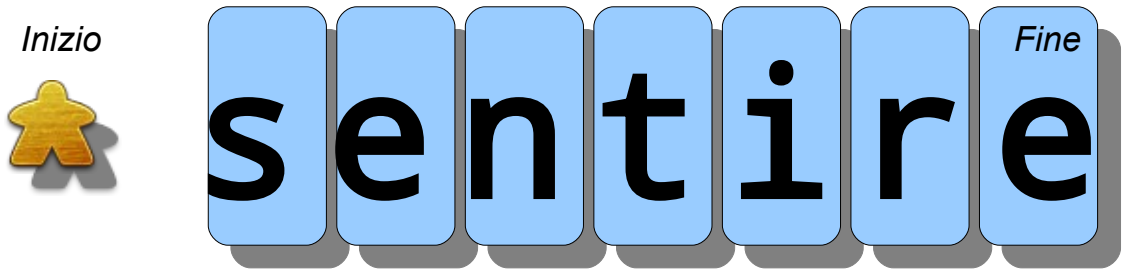
Carattere corrente



Come funziona il matching

sentire. Gli Egiziani trasmisero ai Romani le prep

Carattere corrente

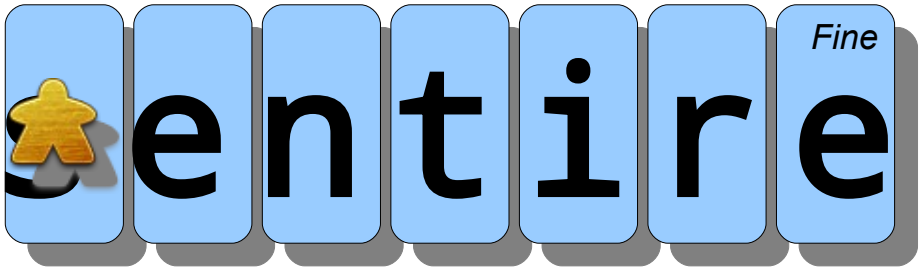


Come funziona il matching

S entire. Gli Egiziani trasmisero ai Romani le prepa

Carattere corrente

Inizio



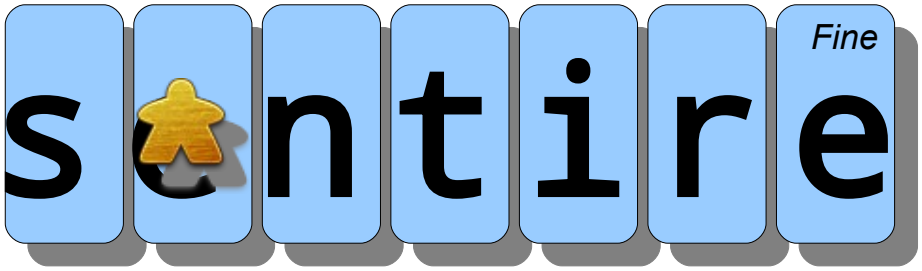
Fine

Come funziona il matching

e ntire. Gli Egiziani trasmisero ai Romani le prepar

Carattere corrente

Inizio

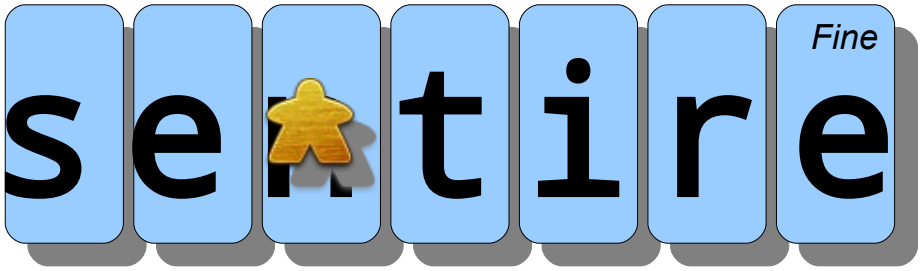


Come funziona il matching

n tire. Gli Egiziani trasmisero ai Romani le prepara

Carattere corrente

Inizio

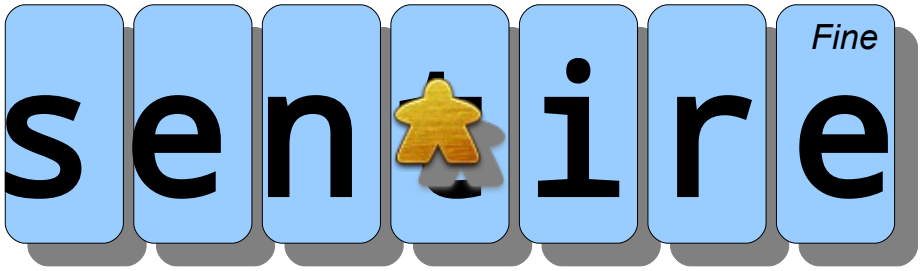


Come funziona il matching

t ire. Gli Egiziani trasmisero ai Romani le preparaz

Carattere corrente

Inizio



Come funziona il matching

i re. Gli Egiziani trasmisero ai Romani le preparazi

Carattere corrente

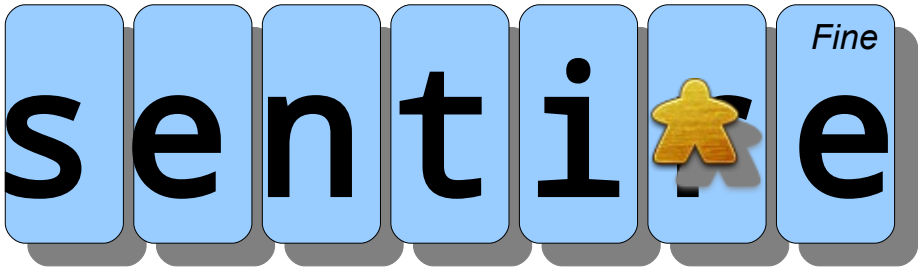
Inizio

Come funziona il matching

r e. Gli Egiziani trasmisero ai Romani le preparazio

Carattere corrente

Inizio *Fine*

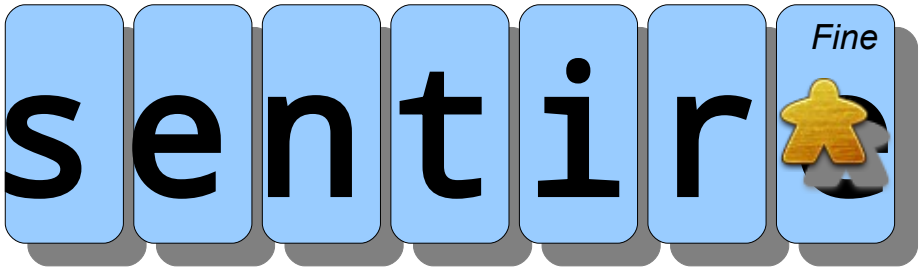


Come funziona il matching

e . Gli Egiziani trasmisero ai Romani le preparazion

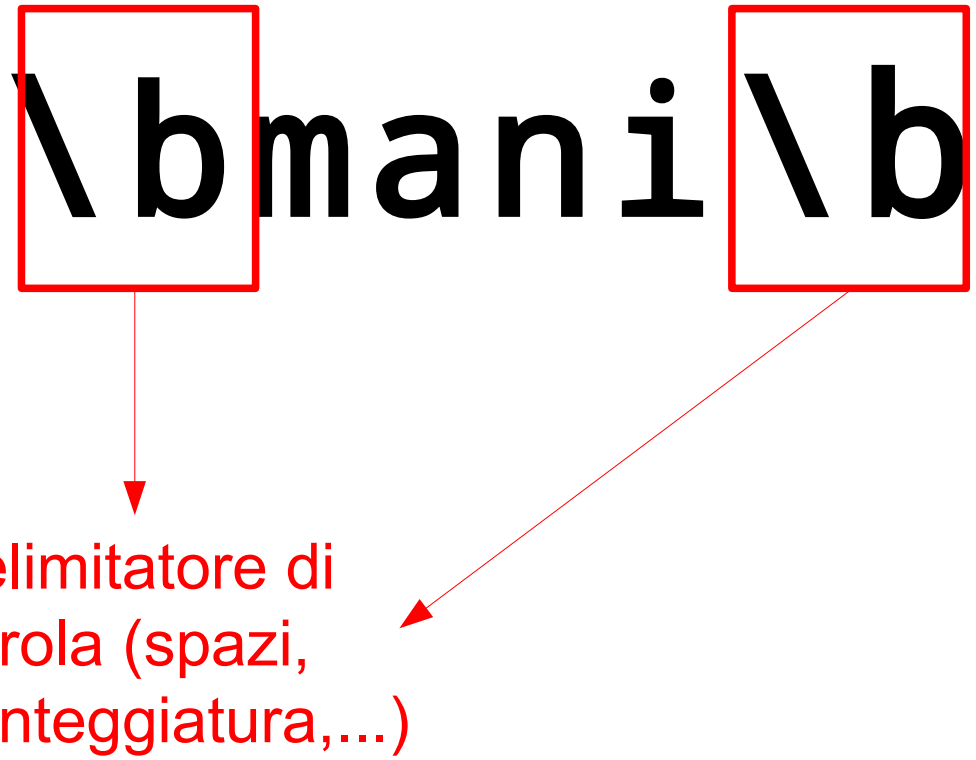
Carattere corrente

Inizio



Caratteri normali e speciali

- Alcuni caratteri assumono un significato speciale se preceduti da '\' (backslash)



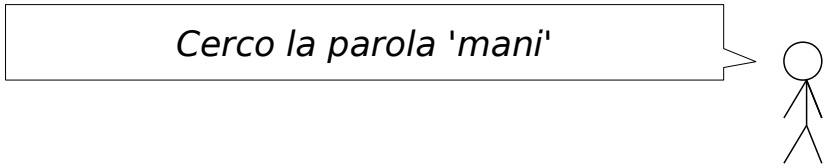
Altri caratteri speciali

\b	Delimitatore di una parola
\B	Inverso di \b (carattere che non è un delimitatore di parola)
\d	Cifra
\D	Inverso di \d (carattere che non è una cifra)
\s	Spazio
\S	Inverso di \s (tutto tranne uno spazio)
\w	Carattere di una parola
\W	Inverso di \w
\A	Inizio del testo (!)
\Z	Fine del testo (!)

Non sono inclusi nel match!

(!) non è riconosciuto da con kregexp

Delimitatore di parola '\b'



kregexpeditor

Compound regular expressions:

- General
  - anything
  - spaces
- General
  - anything
  - spaces
- User Defined

Word Boundary | mani | Word Boundary

Il 16 febbraio del 1951, sedevo su un panchettino di legno, ai suoi piedi, puntavo i gomiti sulle sue ginocchia e con la faccia appoggiata tra le mani, stavo a sentire. Gli Egiziani trasmisero ai Romani le preparazioni che permettevano di trasformar le fibre vegetali del papiro in superfici pulite, bianche, pieghevoli. Oggi siamo ricchi, ma domani possiamo esser poveri. E non si misero in cammino a mani vuote.

ASCII syntax: \bmani\b

Help

Come funziona il matching

t ra le mani, stavo a sentire. Gli Egiziani trasmisero

Carattere corrente

Come funziona il matching

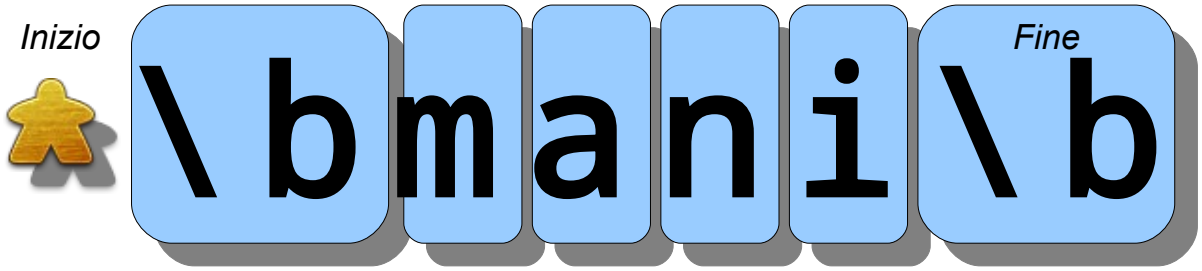
r a le mani, stavo a sentire. Gli Egiziani trasmisero

Carattere corrente

Come funziona il matching

a le mani, stavo a sentire. Gli Egiziani trasmisero a

Carattere corrente

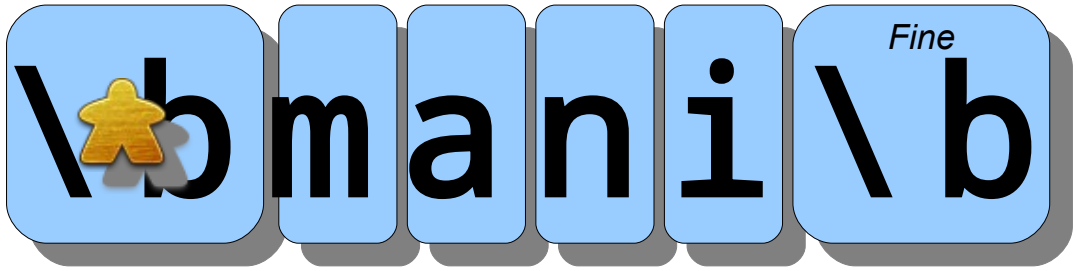


Come funziona il matching

le mani, stavo a sentire. Gli Egiziani trasmisero ai

Carattere corrente

Inizio *Fine*



Come funziona il matching

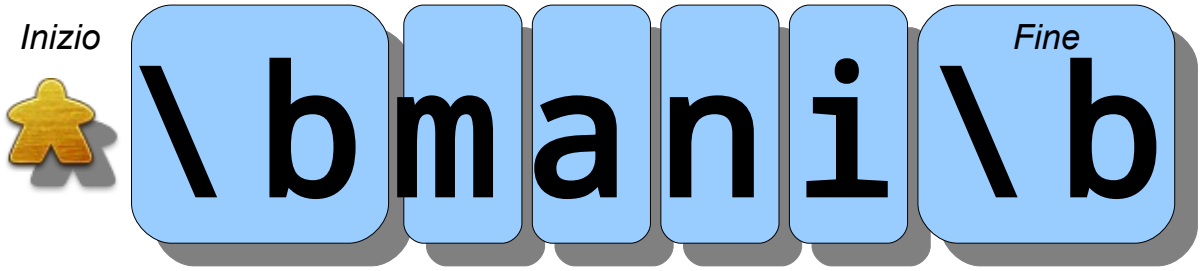
l e mani, stavo a sentire. Gli Egiziani trasmisero ai

Carattere corrente

Come funziona il matching

e mani, stavo a sentire. Gli Egiziani trasmisero ai R

Carattere corrente



Come funziona il matching

mani, stavo a sentire. Gli Egiziani trasmisero ai Ro

Carattere corrente

Come funziona il matching

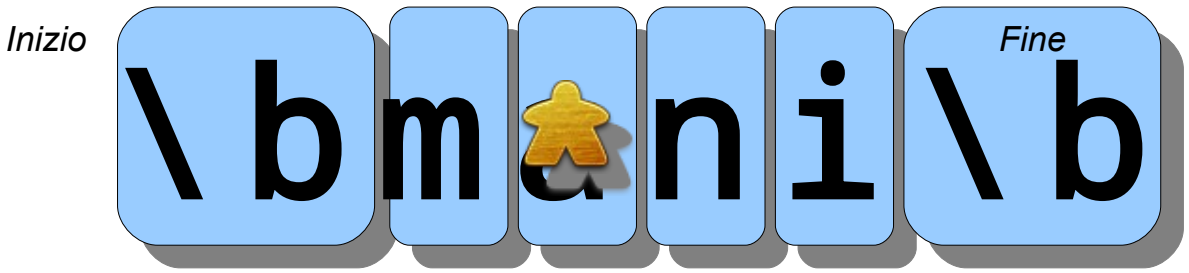
m ani, stavo a sentire. Gli Egiziani trasmisero ai Rom

Carattere corrente

Come funziona il matching

a ni, stavo a sentire. Gli Egiziani trasmisero ai Roma

Carattere corrente



Come funziona il matching

n i, stavo a sentire. Gli Egiziani trasmisero ai Roman

Carattere corrente

Come funziona il matching

i , stavo a sentire. Gli Egiziani trasmisero ai Romani

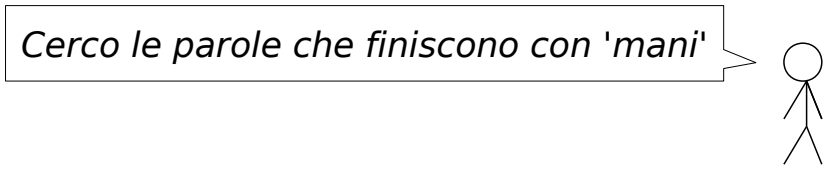
Carattere corrente

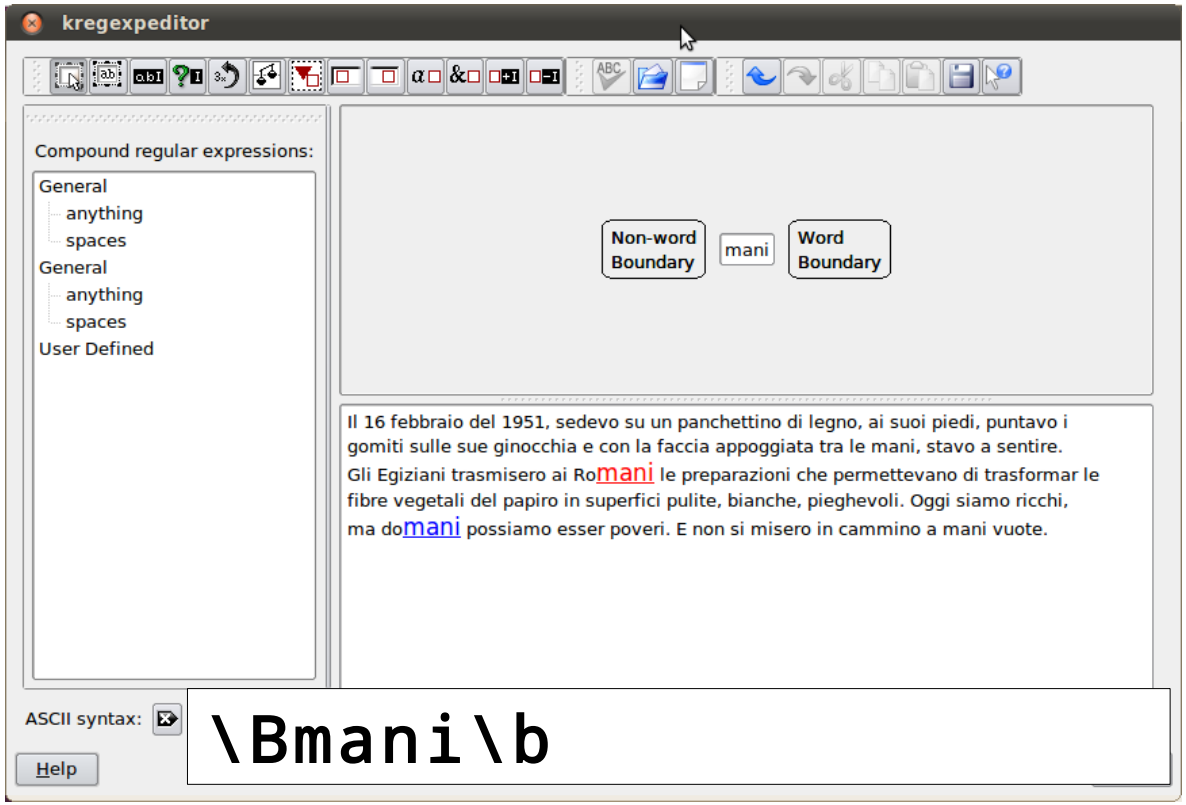
Come funziona il matching

, stavo a sentire. Gli Egiziani trasmisero ai Romani

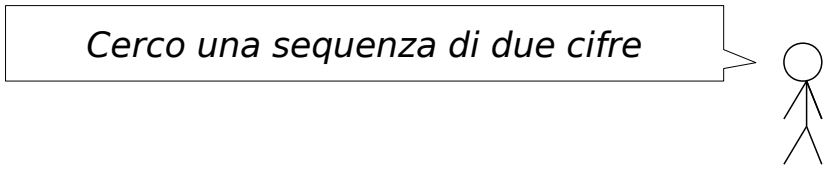
Carattere corrente

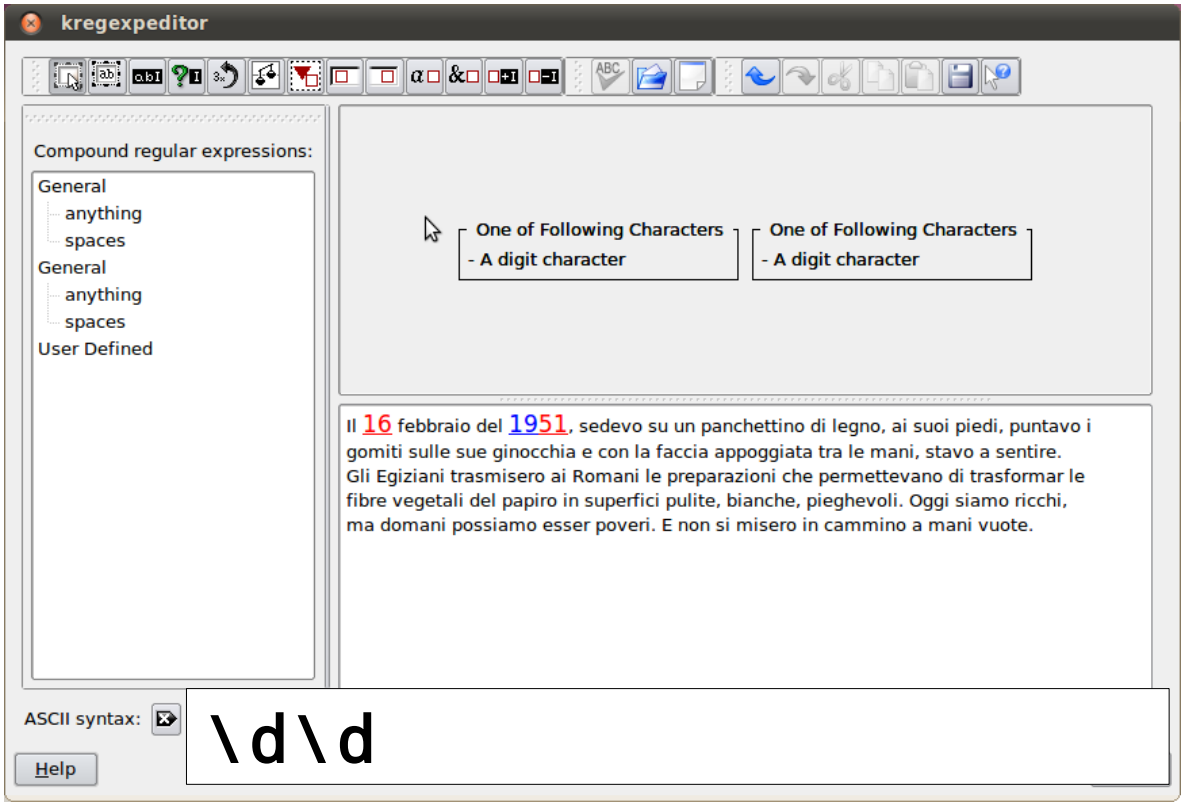
Non-delimitatore di parola '\B'



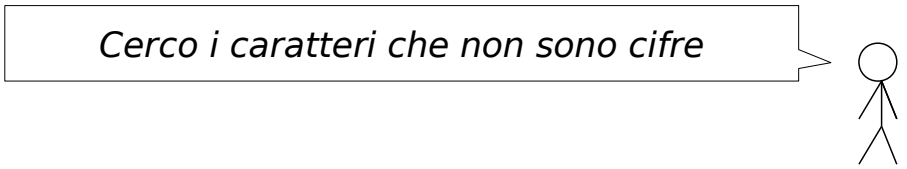


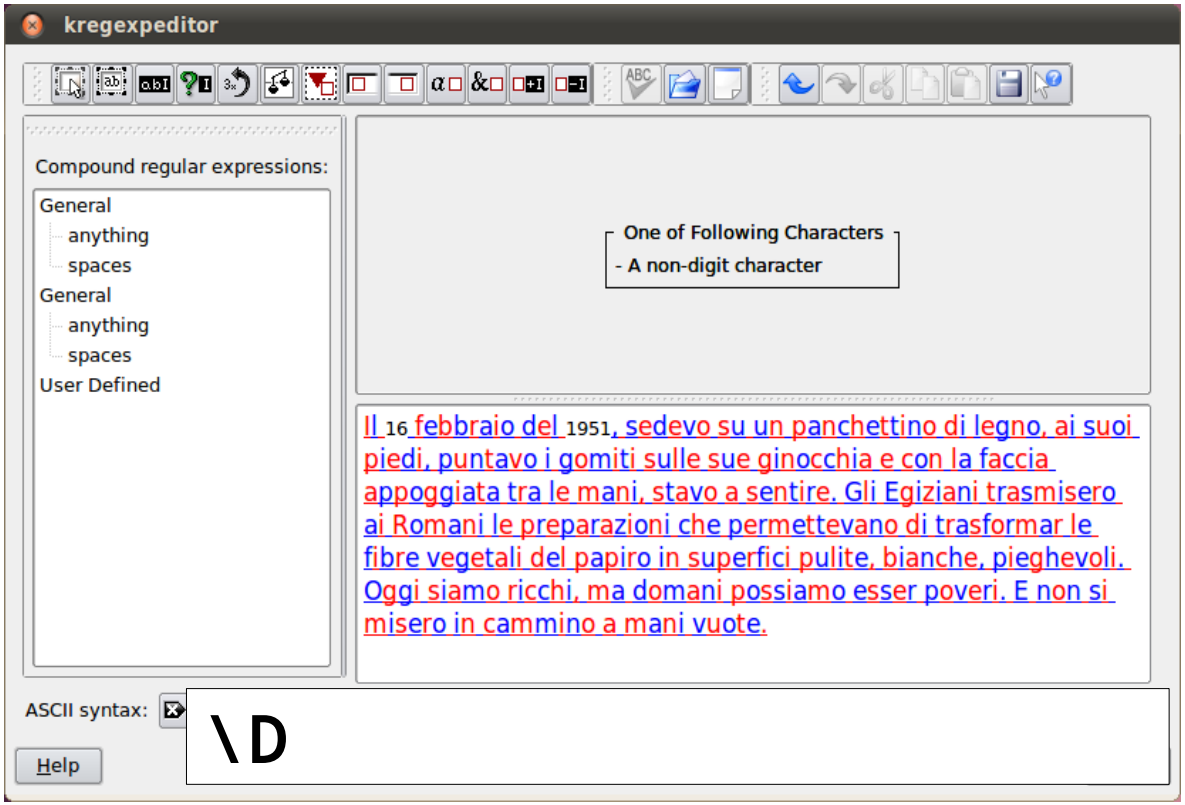
Cifre '\d'


Cerco una sequenza di due cifre

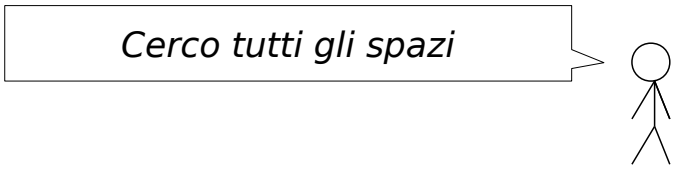

kregexpeditor
Compound regular expressions:
General
anything
spaces
General
anything
spaces
User Defined
One of Following Characters
- A digit character
One of Following Characters
- A digit character
Il 16 febbraio del 1951, sedevo su un panchettino di legno, ai suoi piedi, puntavo i gomiti sulle sue ginocchia e con la faccia appoggiata tra le mani, stavo a sentire.
Gli Egiziani trasmisero ai Romani le preparazioni che permettevano di trasformar le fibre vegetali del papiro in superfici pulite, bianche, pieghevoli. Oggi siamo ricchi, ma domani possiamo esser poveri. E non si misero in cammino a mani vuote.
ASCII syntax:
\d\d
Help

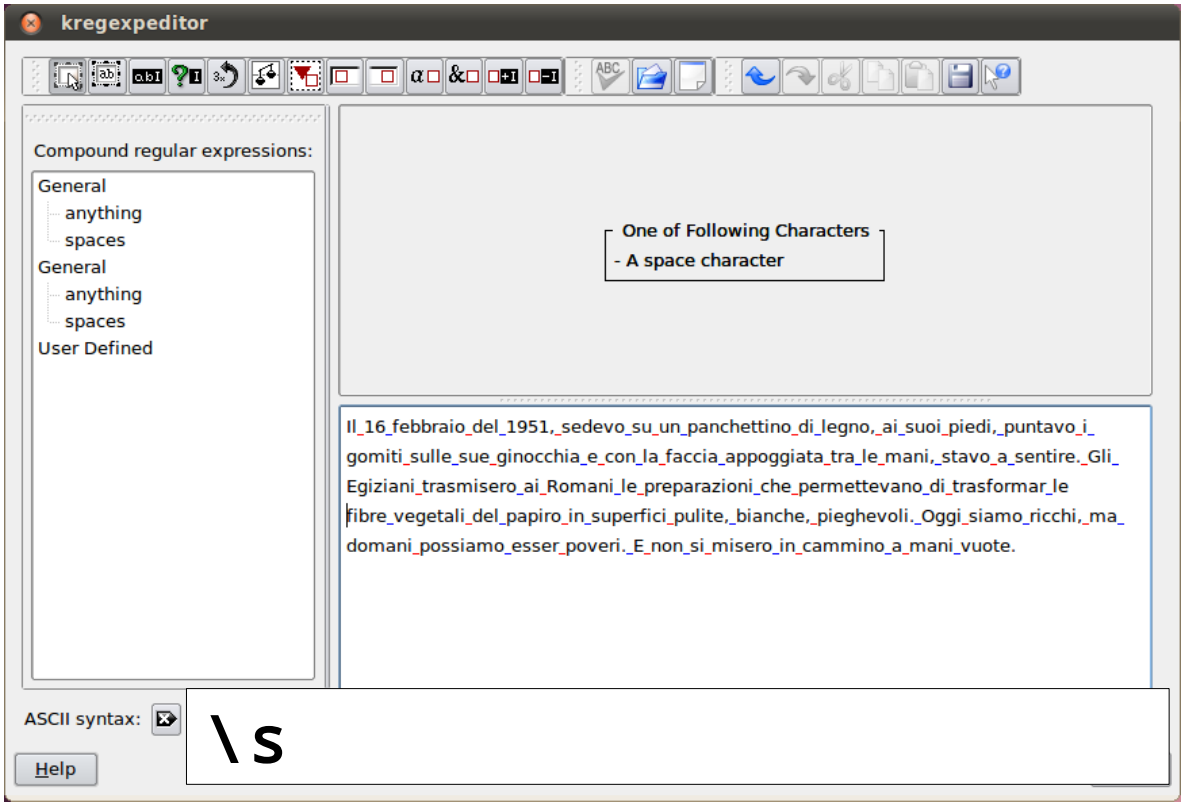
Non-cifre '\D'


Cerco i caratteri che non sono cifre

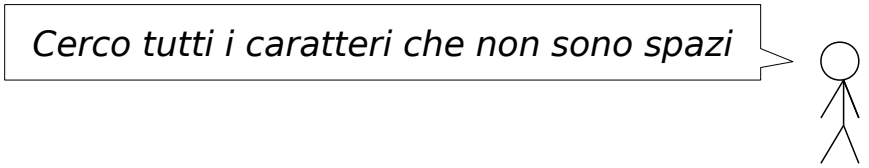

kregexpeditor
Compound regular expressions:
General
anything
spaces
General
anything
spaces
User Defined
One of Following Characters
- A non-digit character
Il 16 febbraio del 1951, sedevo su un panchettino di legno, ai suoi piedi, puntavo i gomiti sulle sue ginocchia e con la faccia appoggiata tra le mani, stavo a sentire. Gli Egiziani trasmisero ai Romani le preparazioni che permettevano di trasformar le fibre vegetali del papiro in superfici pulite, bianche, pieghevoli. Oggi siamo ricchi, ma domani possiamo esser poveri. E non si misero in cammino a mani vuote.
ASCII syntax:
\D
Help

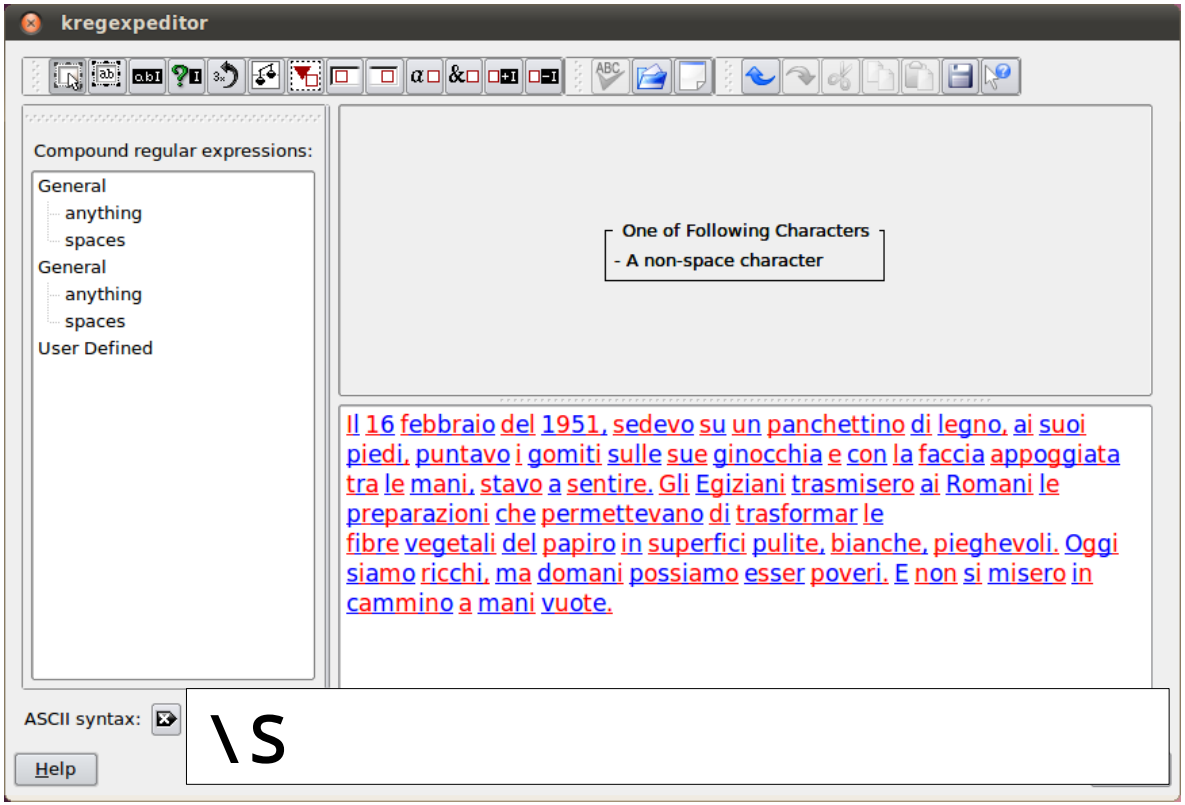
Spazi '\s'


Cerco tutti gli spazi

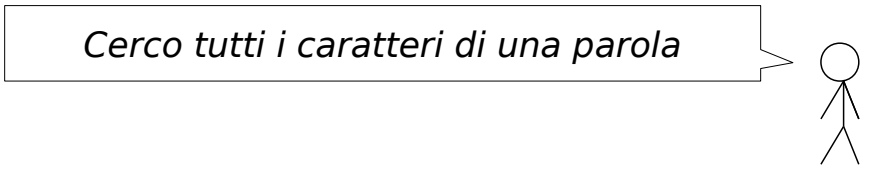

kregexpeditor
Compound regular expressions:
General
anything
spaces
General
anything
spaces
User Defined
One of Following Characters
- A space character
Il_16_febbraio_del_1951,_sedevo_su_un_panchettino_di_legno,_ai_suoi_piedi,_puntavo_i_gomiti_sulle_sue_ginocchia_e_con_la_faccia_appoggiata_tra_le_mani,_stavo_a_sentire._Gli_Egiziani_trasmisero_ai_Romani_le_preparazioni_che_permettevano_di_trasformar_le fibre_vegetali_del_papiro_in_superfici_pulite,_bianche,_pieghevoli._Oggi_siamo_ricchi,_ma_domani_possiamo_esser_poveri._E_non_si_misero_in_cammino_a_mani_vuote.
ASCII syntax:
\s
Help

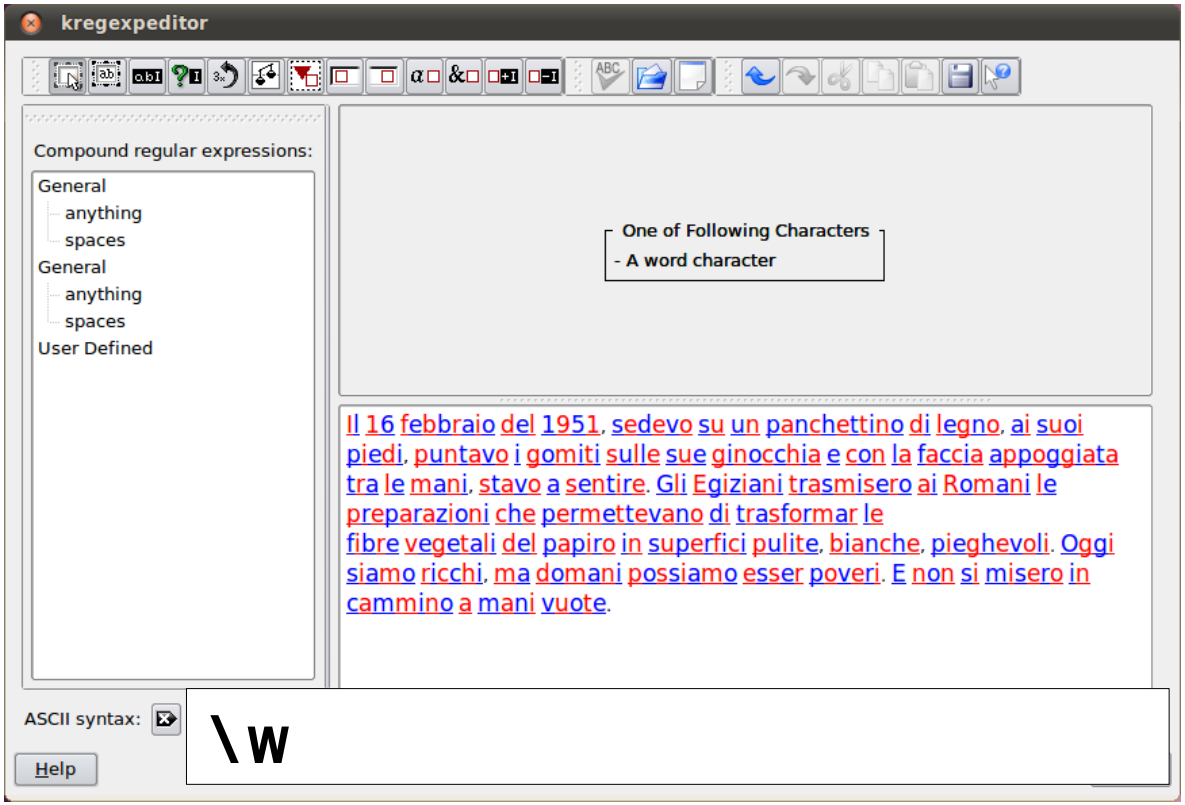
Non-spazi '\S'


Cerco tutti i caratteri che non sono spazi


kregexpeditor
Compound regular expressions:
General
anything
spaces
General
anything
spaces
User Defined
One of Following Characters
- A non-space character
Il 16 febbraio del 1951, sedevo su un panchettino di legno, ai suoi piedi, puntavo i gomiti sulle sue ginocchia e con la faccia appoggiata tra le mani, stavo a sentire. Gli Egiziani trasmisero ai Romani le preparazioni che permettevano di trasformar le fibre vegetali del papiro in superfici pulite, bianche, pieghevoli. Oggi siamo ricchi, ma domani possiamo esser poveri. E non si misero in cammino a mani vuote.
ASCII syntax:
\S
Help

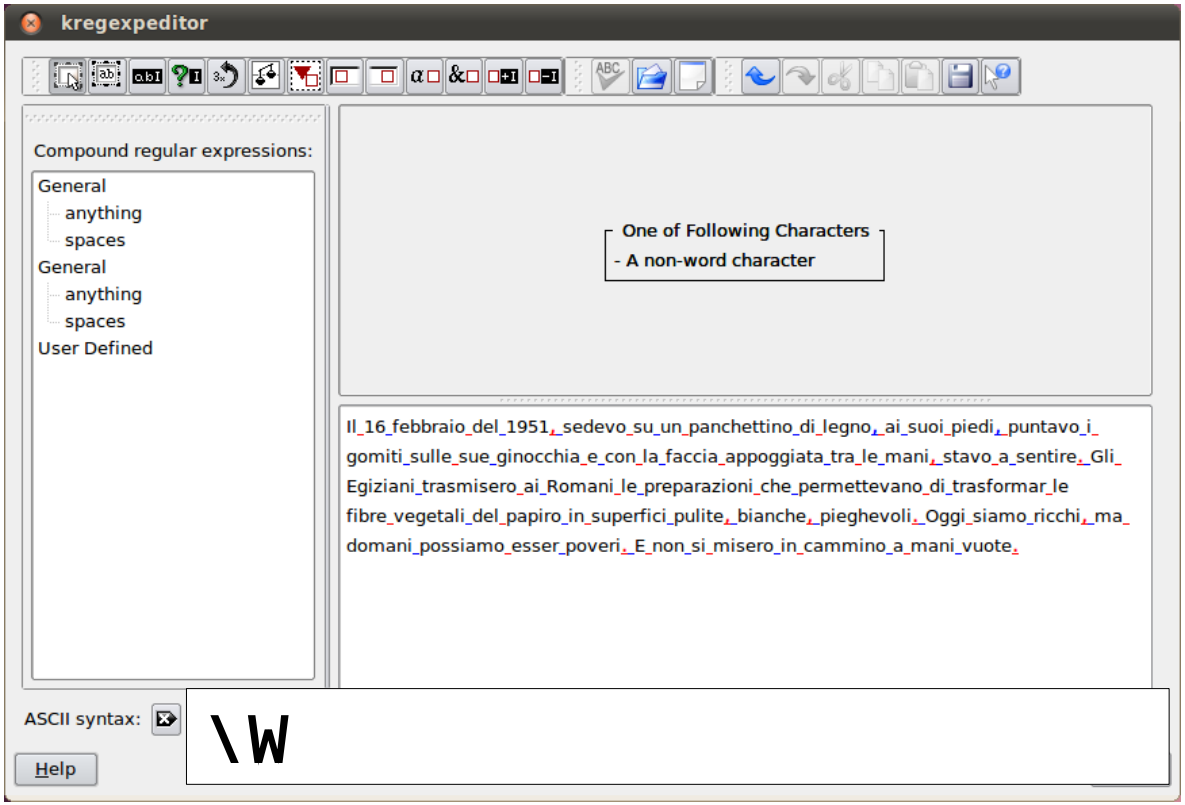
Caratteri di una parola '\w'



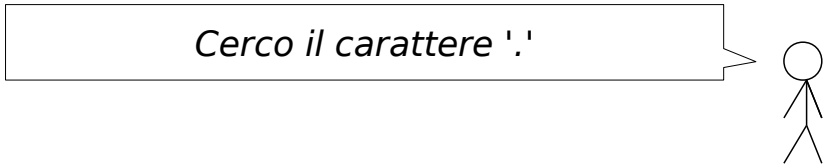


Caratteri che non sono di una parola '\W'

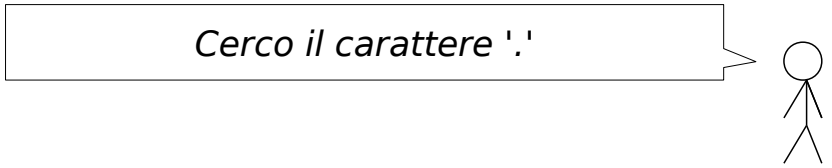
Cerco tutti i caratteri che non fanno parte di una parola

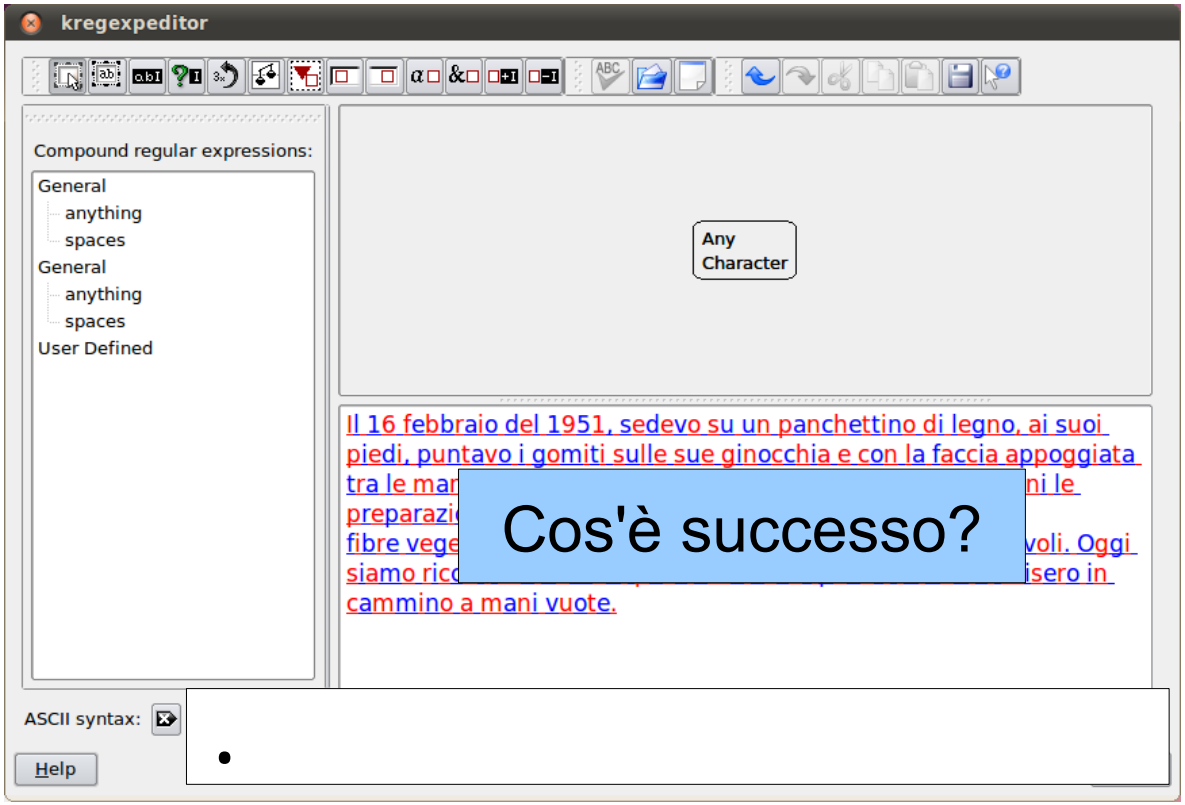


Ancora sul backslash



Ancora sul backslash





Caratteri speciali

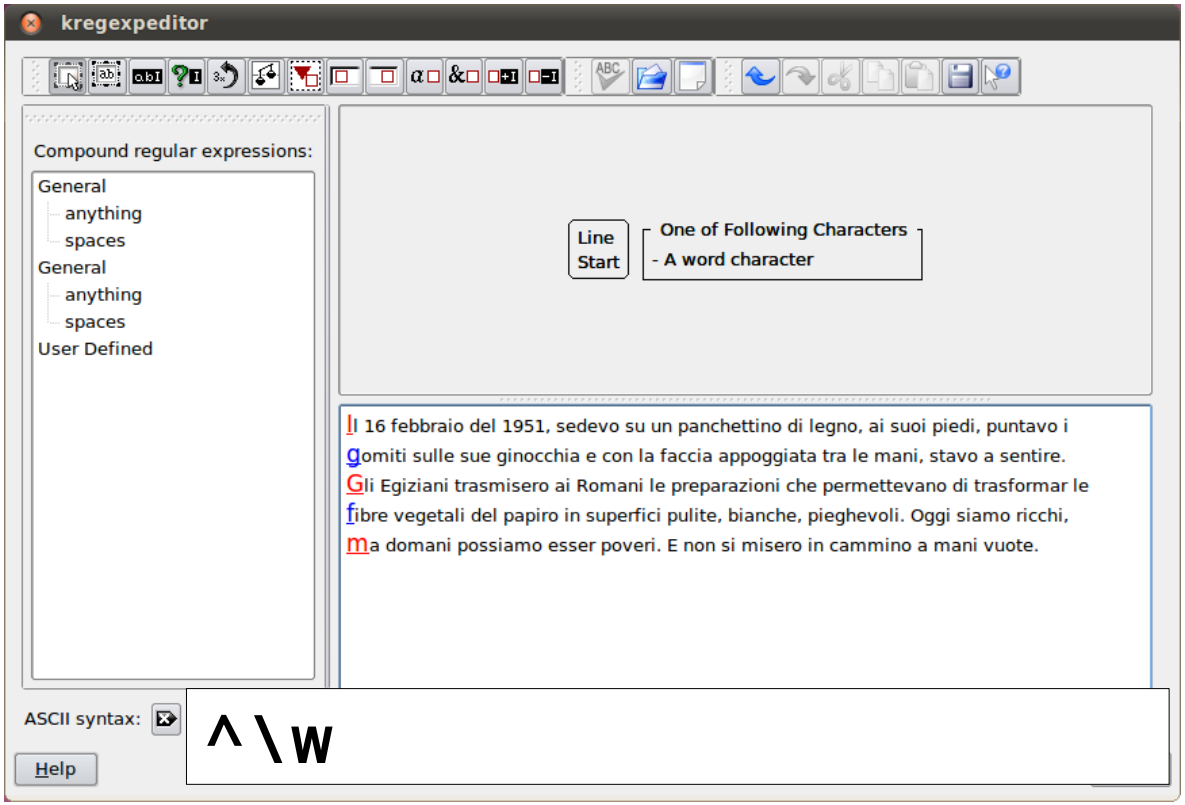
- Alcuni caratteri sono “speciali” anche senza backslash
  - **Per poterli usare come caratteri normali bisogna anteporre un backslash '\'**

^	Inizio della riga
$	Fine della riga
*	Zero o più (quantificatore)
+	Uno o più (quantificatore)
?	Zero o uno (quantificatore)
.	Qualsiasi carattere tranne ritorno a capo
( )	Delimitatori di gruppo
{ }	Delimitatori per il numero di ripetizioni
[ ]	Insiemi di caratteri
¦	Alternativa

Attenzione! $ non funziona correttamente in kregexpeditor

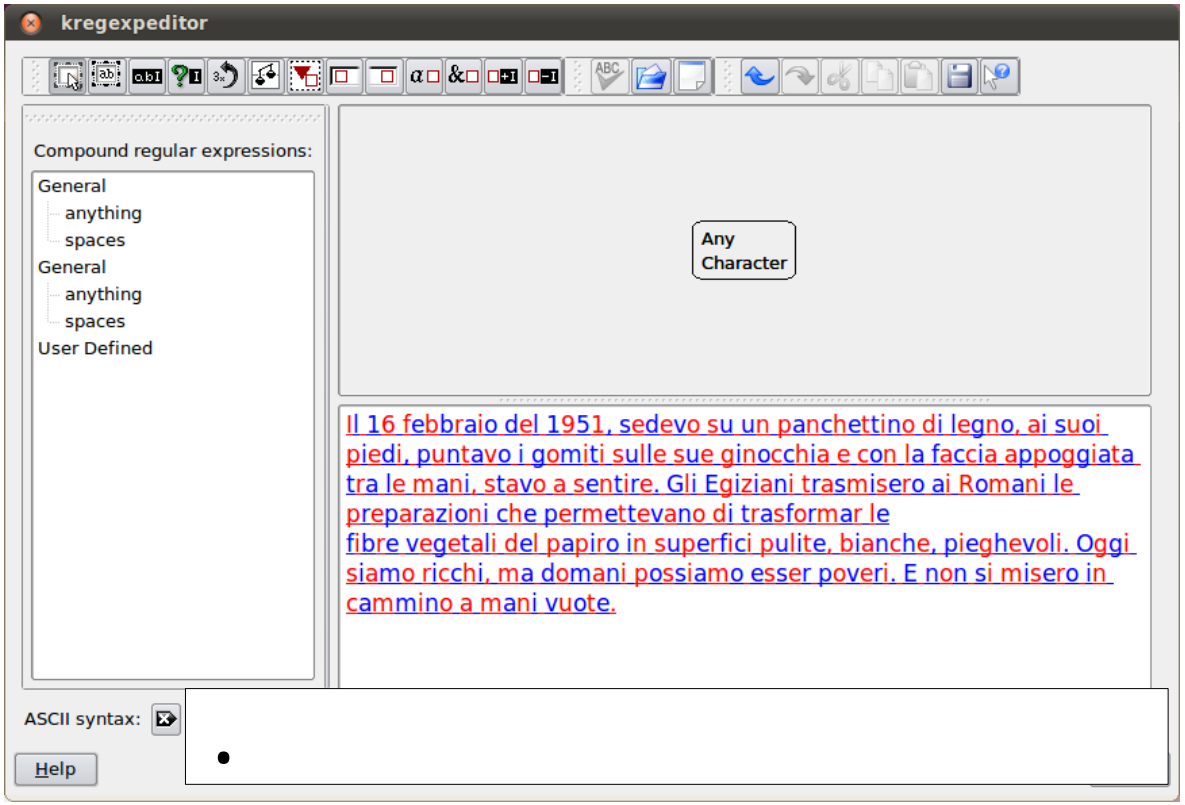
Inizio riga '^'

Cerco tutti i caratteri che sono all'inizio di una riga



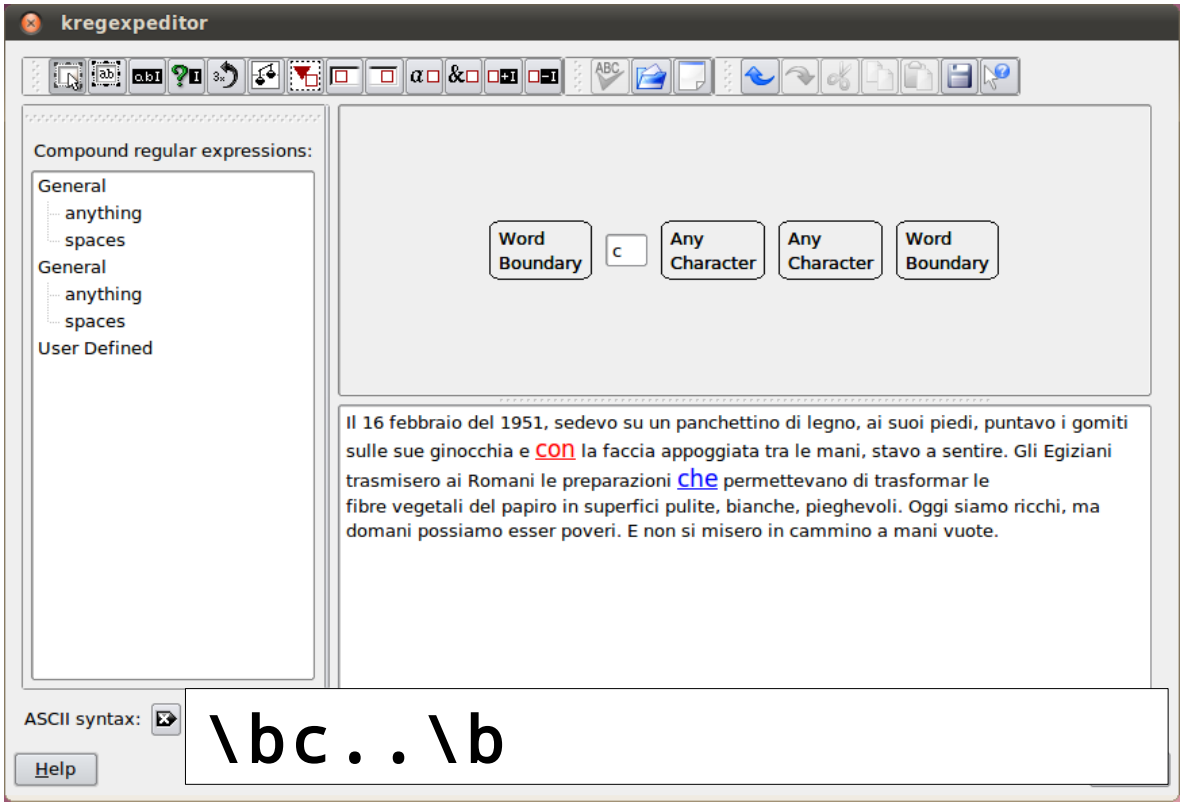
Wildcard '.'

- Il carattere speciale/jolly '.' (punto) equivale a qualsiasi carattere (tranne il ritorno a capo '\n')



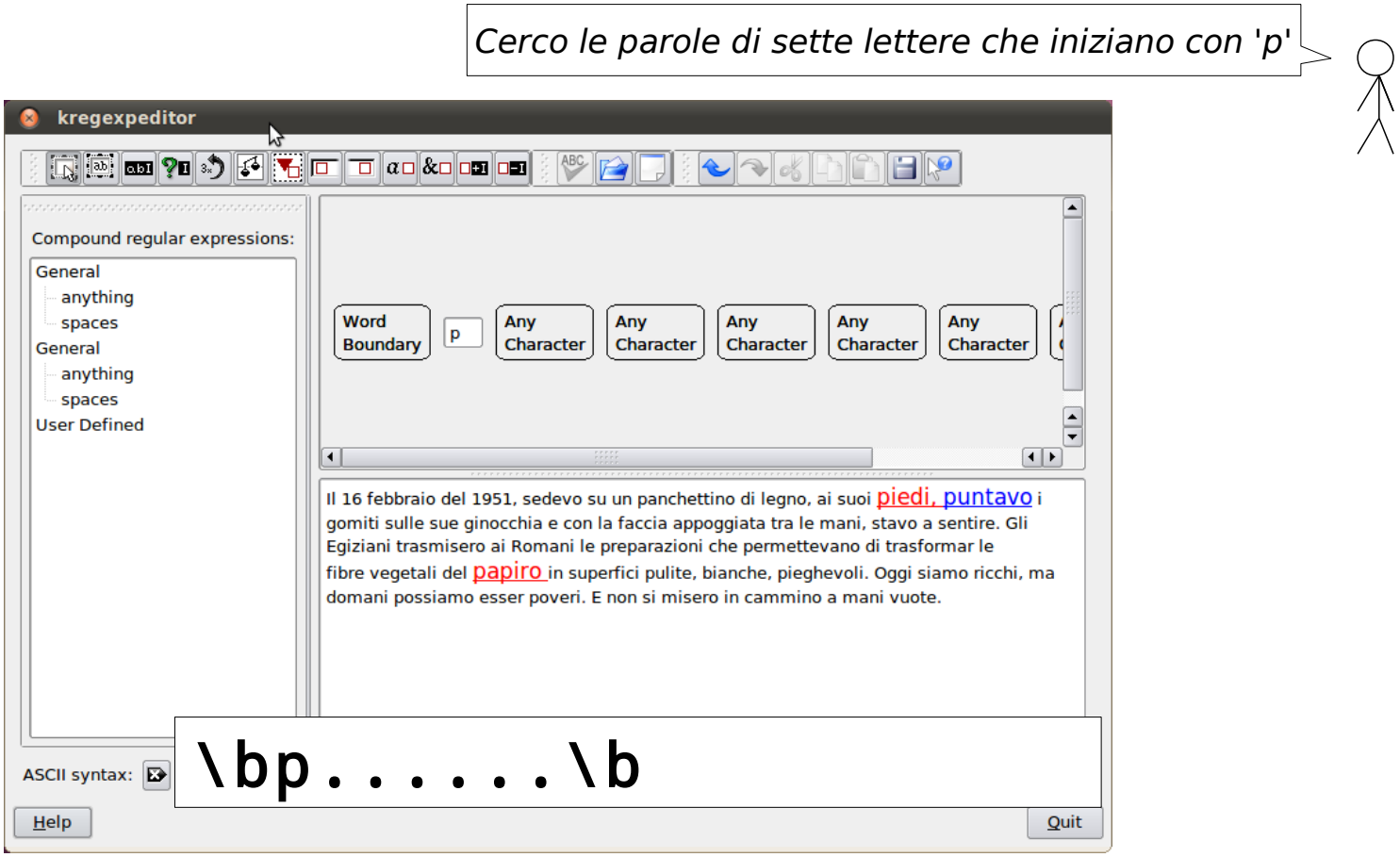
Wildcard '.'

Cerco le parole di tre lettere che iniziano con 'c'



Wildcard '.'

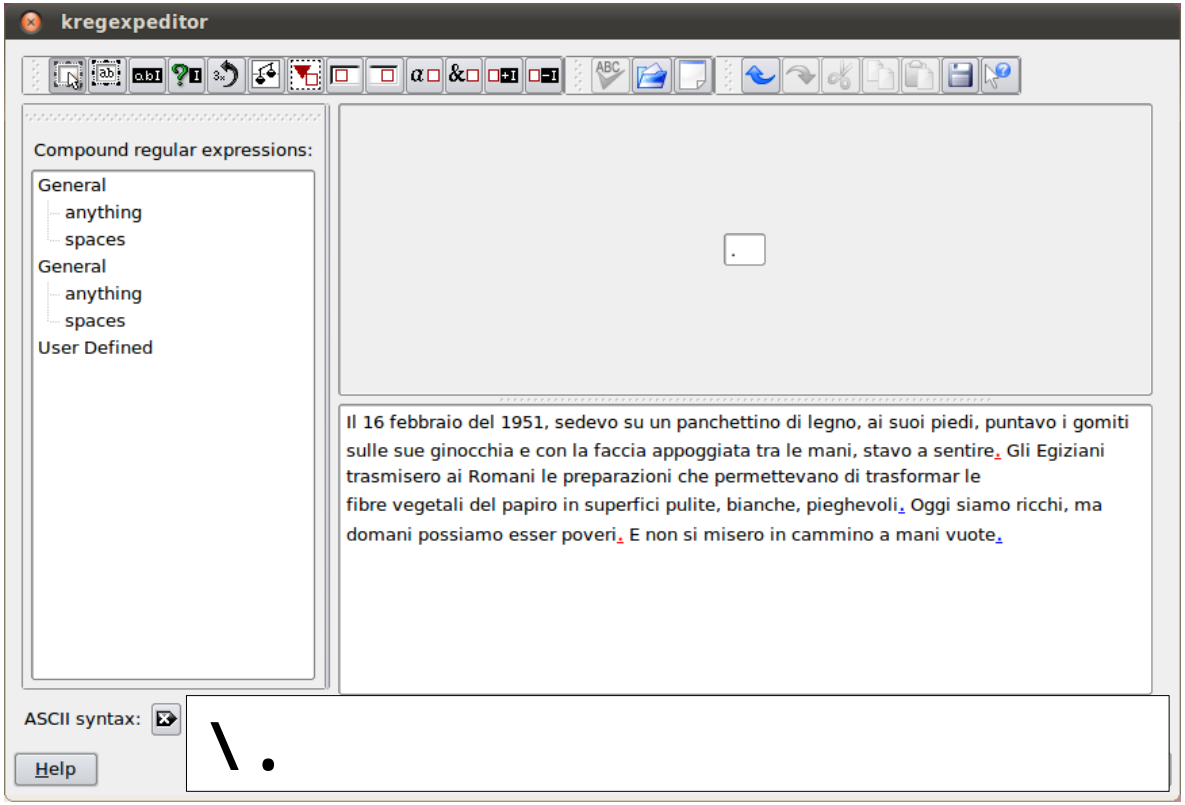
- In realtà l'esempio precedente non è molto robusto, es.



Wildcard '.'

Cerco il carattere '.'

Devo usare backslash perché '.' è un carattere speciale



Insiemi di caratteri [ ]

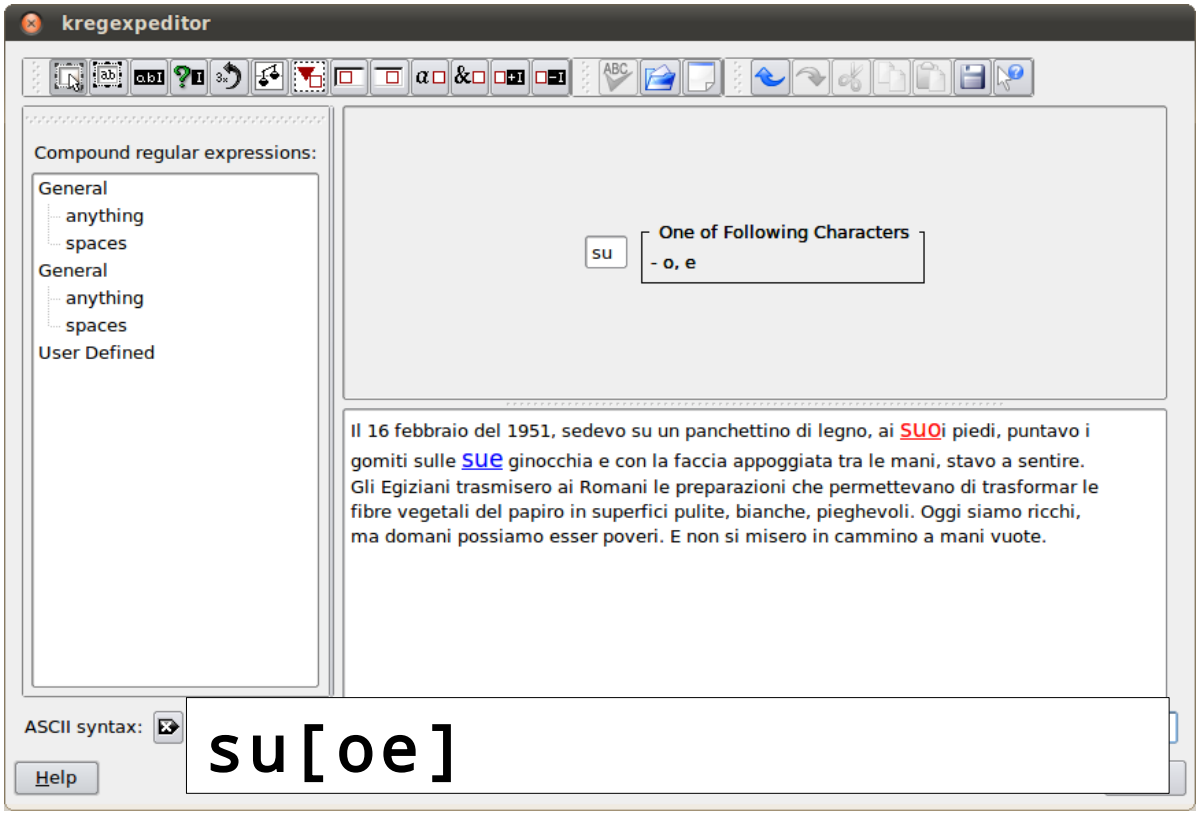
- Con **[ ]** possiamo specificare un carattere a scelta tra un insieme definito

```
su[oe]
```

Cerca la corrispondenza con 'suo' o 'sue'

Insiemi di caratteri [ ]

- Con **[ ]** possiamo specificare un carattere a scelta tra un insieme definito



Insiemi di caratteri POSIX

- Alcuni insiemi di caratteri sono già definiti dallo standard POSIX
- Esempi:
  - **[:alnum:]** caratteri alfanumerici (equivale a [a-zA-Z0-9])
  - **[:alpha:]** caratteri alfabetici (equivale a [a-zA-Z])
  - **[:digit:]** cifre (equivale a [0-9])
  - **[:blank:]** spazi e tabulazioni
  - **[:space:]** tutti gli spazi e interruzioni di riga

Sequenza di caratteri [-]

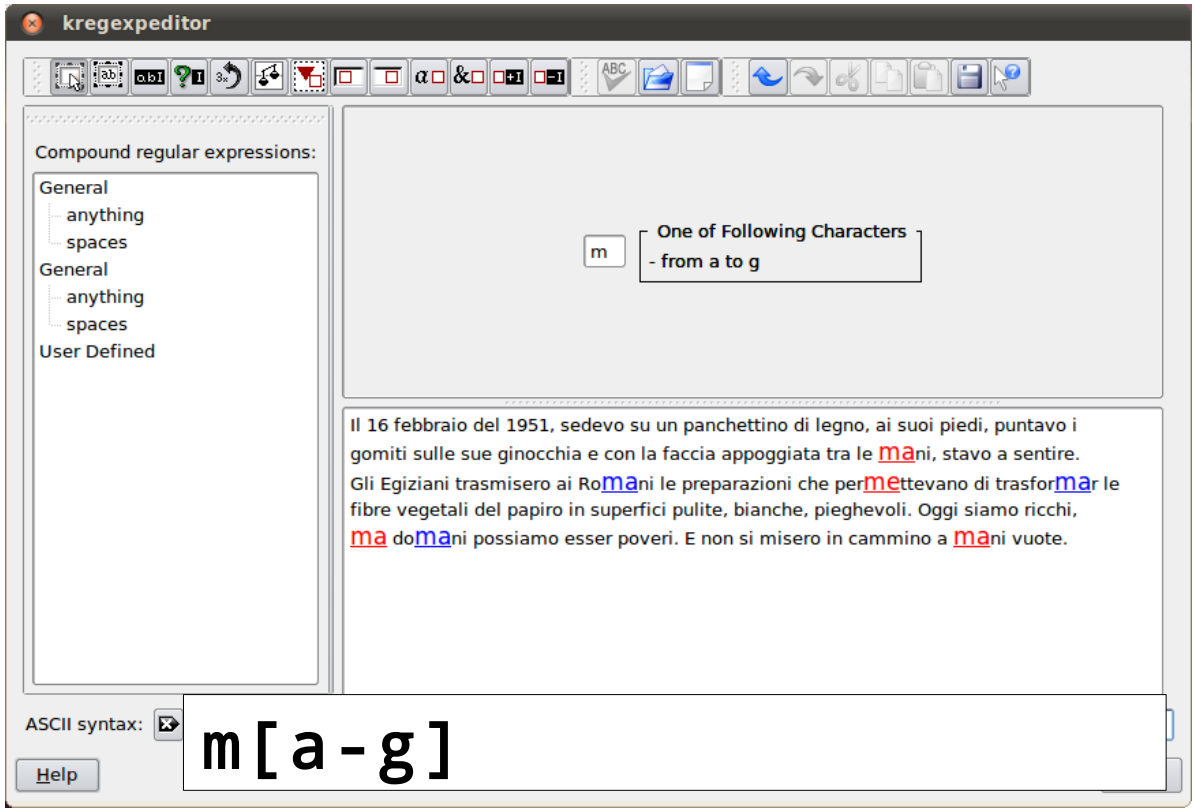
- Con **[-]** possiamo specificare un carattere a scelta tra una sequenza definita dal carattere iniziale e il carattere finale (secondo l'ordine ASCII, http://en.wikipedia.org/wiki/ASCII)

```
m[a-g]
```

Cerca la corrispondenza con 'm' seguita da un carattere da 'a' a 'g'

Sequenza di caratteri [-]

- Con **[-]** possiamo specificare un carattere a scelta tra una sequenza definita dal carattere iniziale e il carattere finale (secondo l'ordine ASCII, http://en.wikipedia.org/wiki/ASCII)



Nota: posso specificare anche più intervalli, p.es. [e-gs-z] (da 'e' a 'g' oppure da 's' a 'z')

Tutti i caratteri tranne... [^]

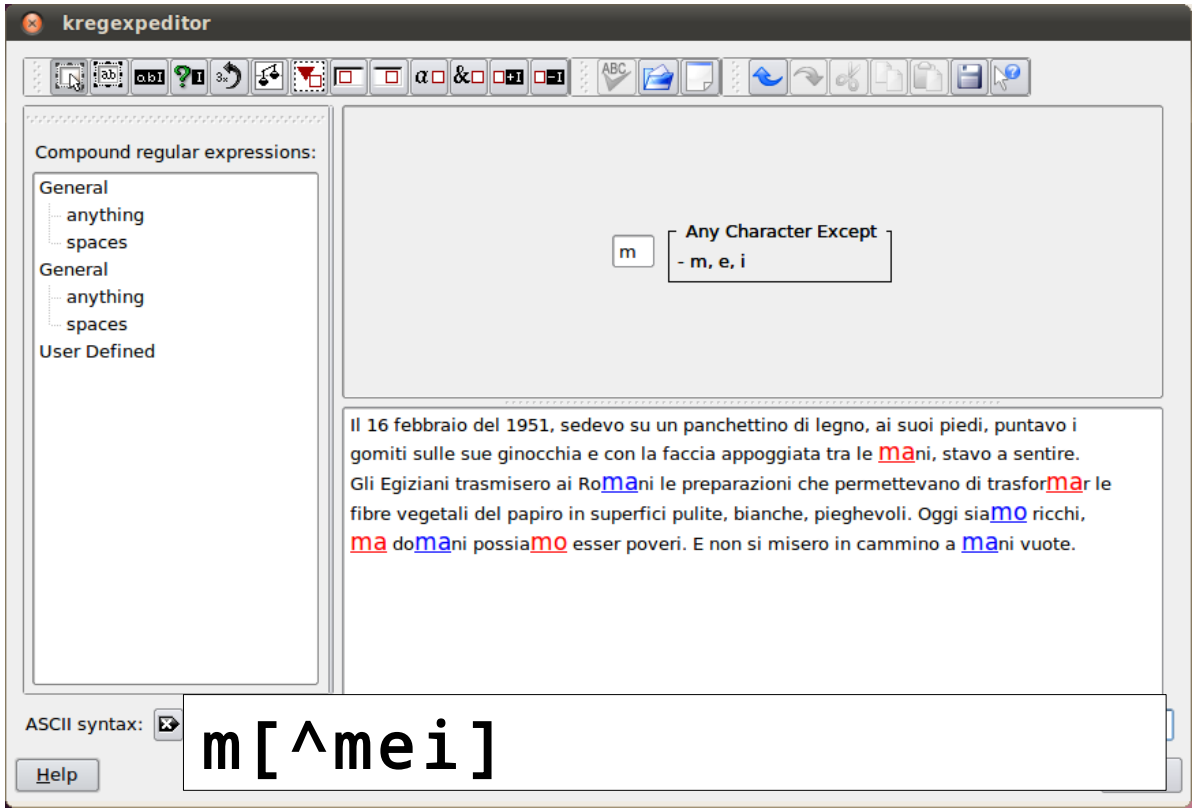
- Con **[^]** possiamo cercare la corrispondenza di tutti i caratteri escludendone alcuni

```
m[^mei]
```

Cerca la corrispondenza con 'm' seguita da un un qualsiasi carattere tranne 'm', 'e', oppure 'i'

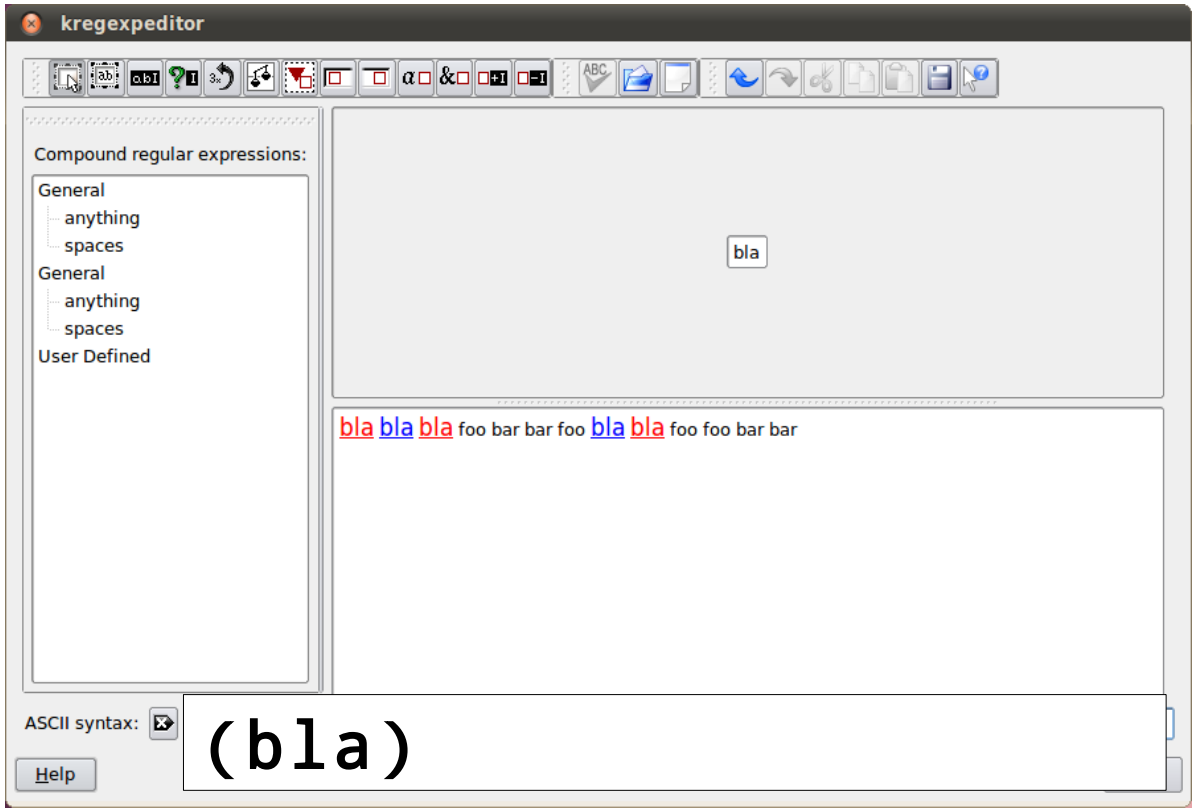
Tutti i caratteri tranne... [^]

- Con **[^]** possiamo cercare la corrispondenza di tutti i caratteri escludendone alcuni



Gruppi

- Con le parentesi **( )** possiamo raggruppare più termini
  - Vedremo più avanti come recuperare i gruppi dopo aver trovato le corrispondenze



Alternative

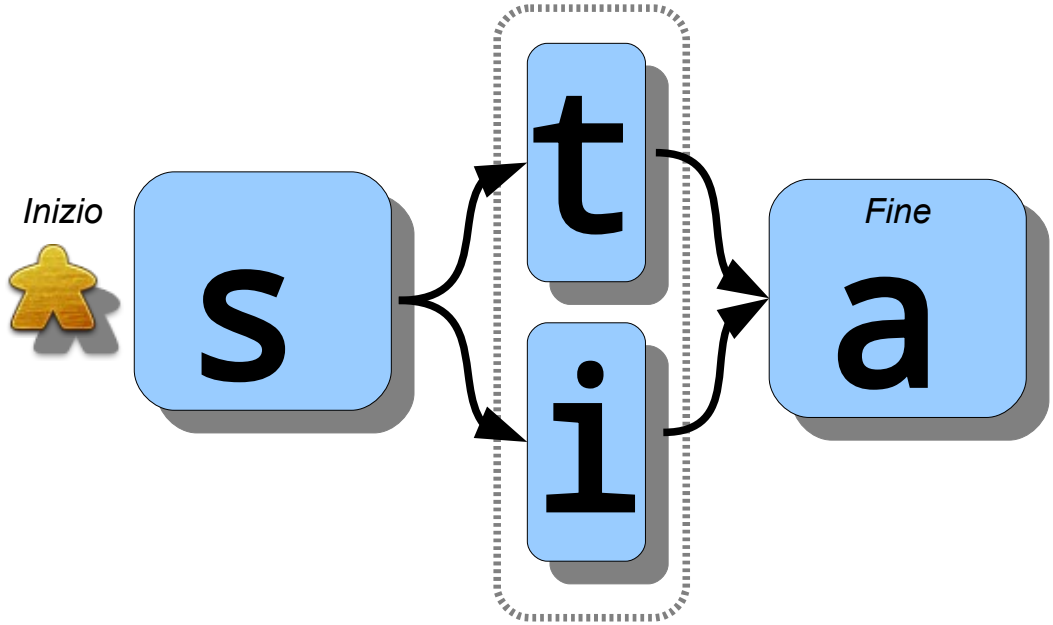
- Il carattere | ci permette di definire delle alternative

```
s(t|i)a
```

Cerca la corrispondenza con 's' seguita da 't' oppure 'i', e poi da 'a'

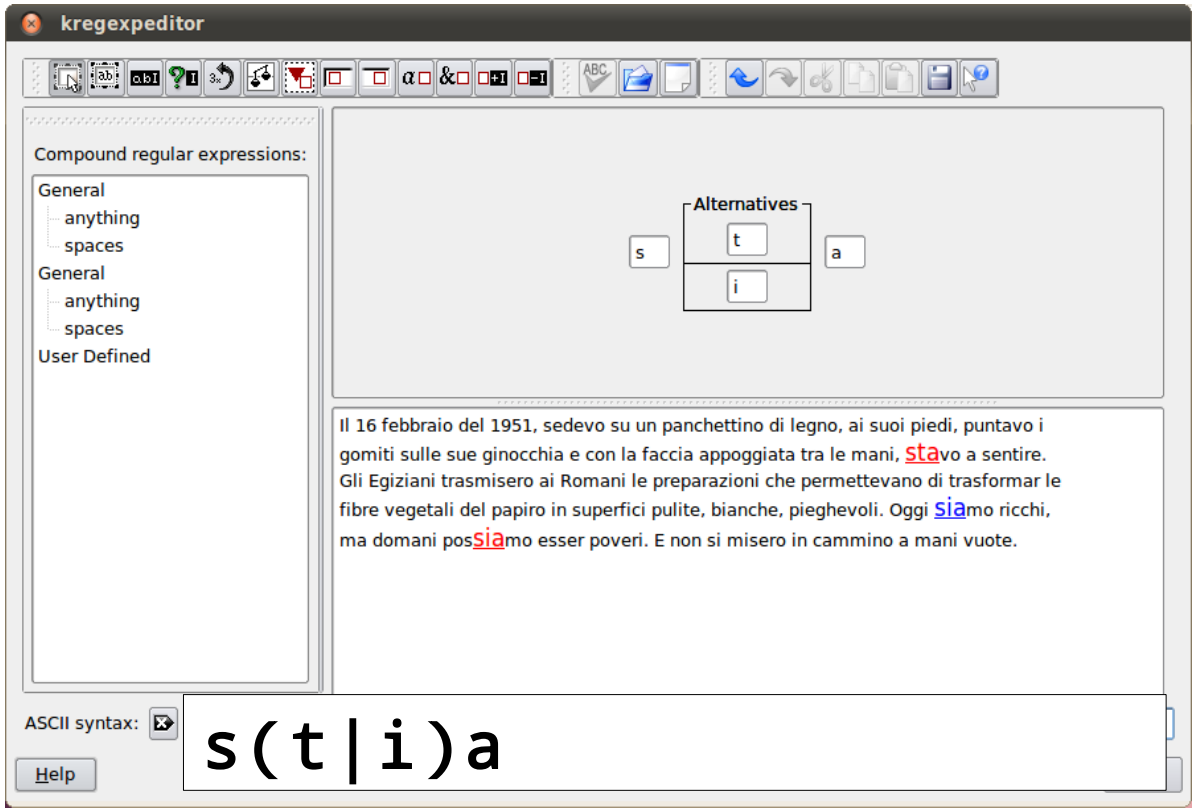
Alternative

- Il carattere | ci permette di definire delle alternative



Alternative

- Il carattere | ci permette di definire delle alternative



Alternative

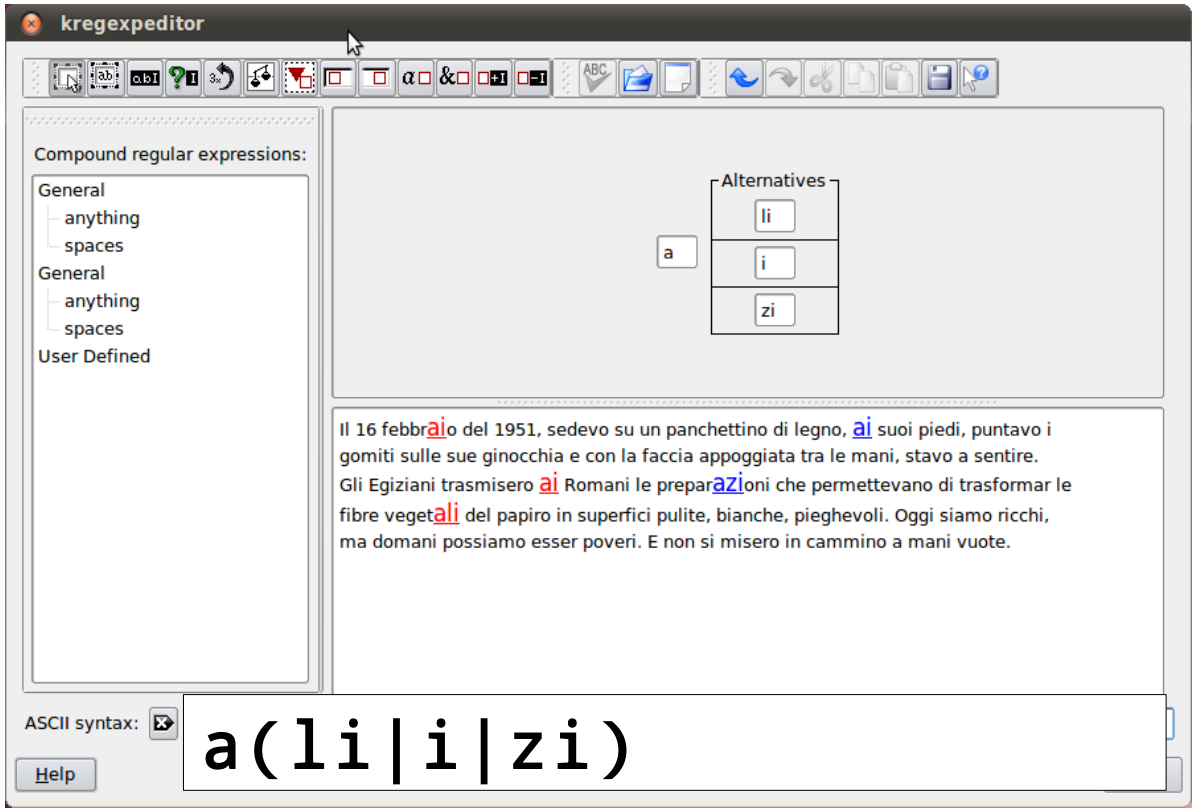
- Possiamo avere più alternative possibili

```
a(li|i|zi)
```

Cerca la corrispondenza con 'a' seguita da 'l' e 'i' oppure 'i' oppure 'z' e 'i'

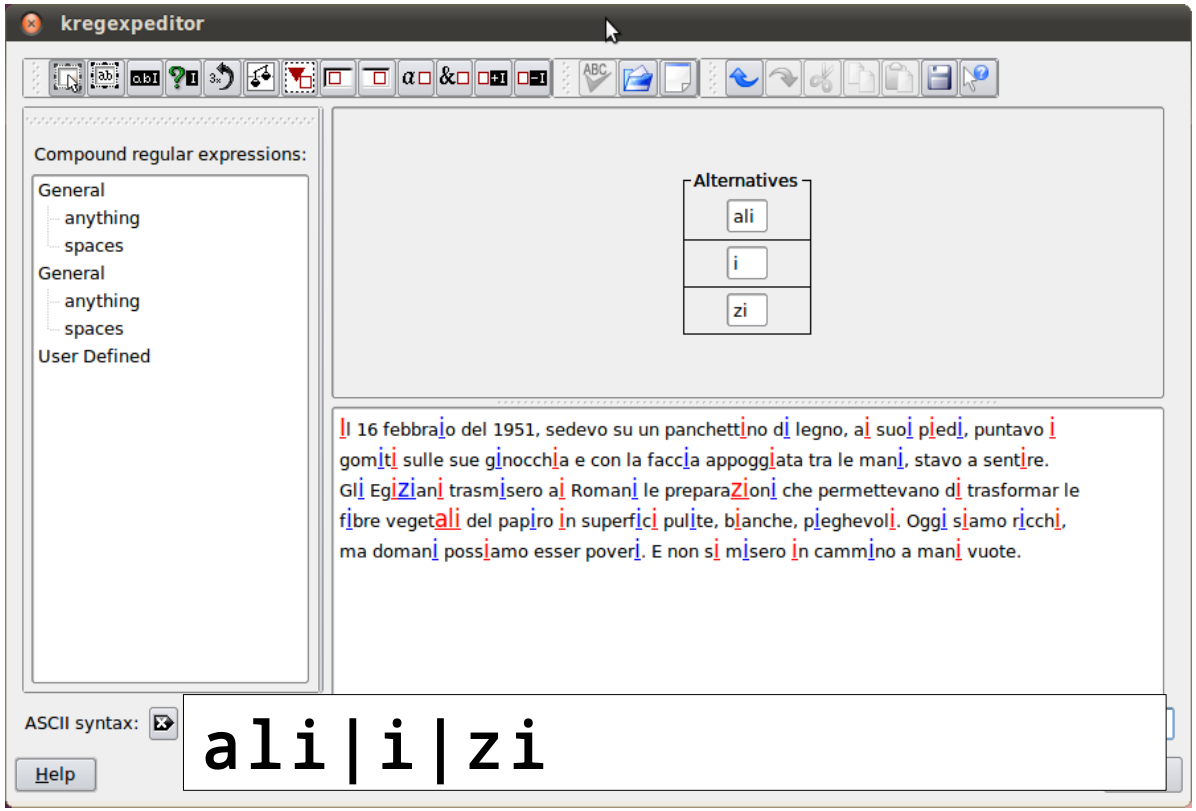
Alternative

- Possiamo avere più alternative possibili



Alternative

- Attenzione! Con le alternative bisogna utilizzare correttamente i gruppi!



Zero o più (*)

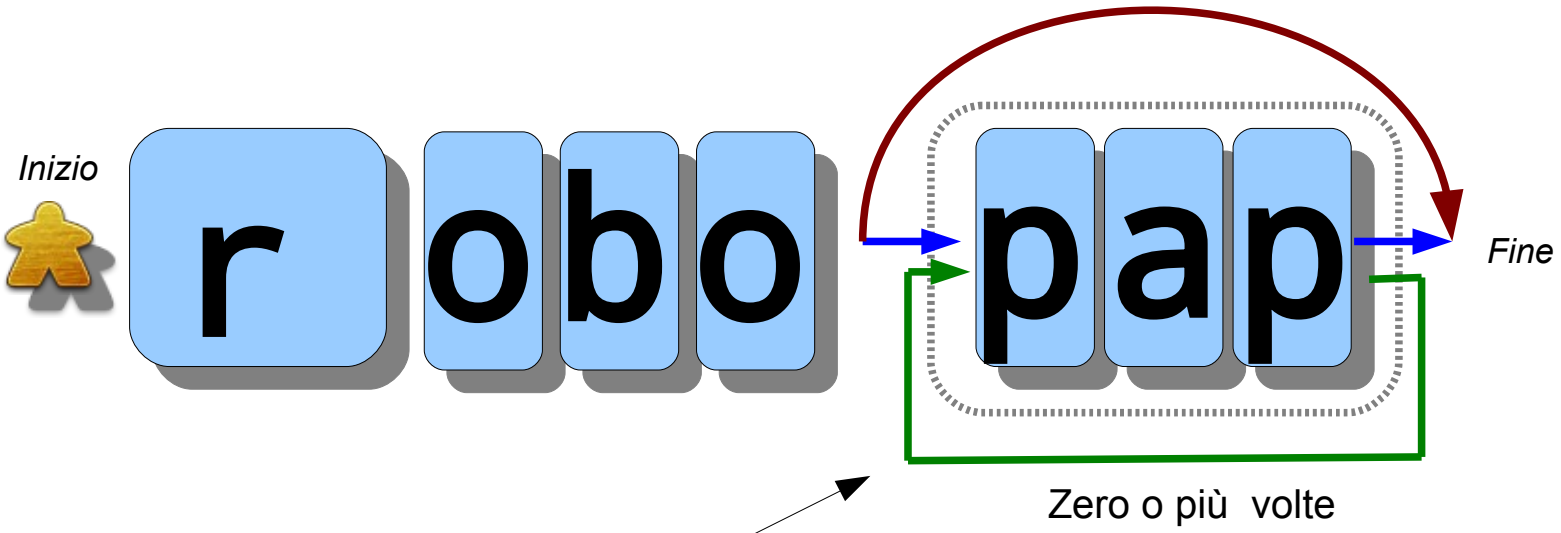
- Aggiungendo il quantificatore '*' possiamo specificare zero o più ripetizioni del termine (o gruppo) che precede

```
robo(pap)*
```

Cerca la corrispondenza le sequenze che contengono 'robo' seguito da zero o più ripetizioni di 'pap'

Zero o più (*)

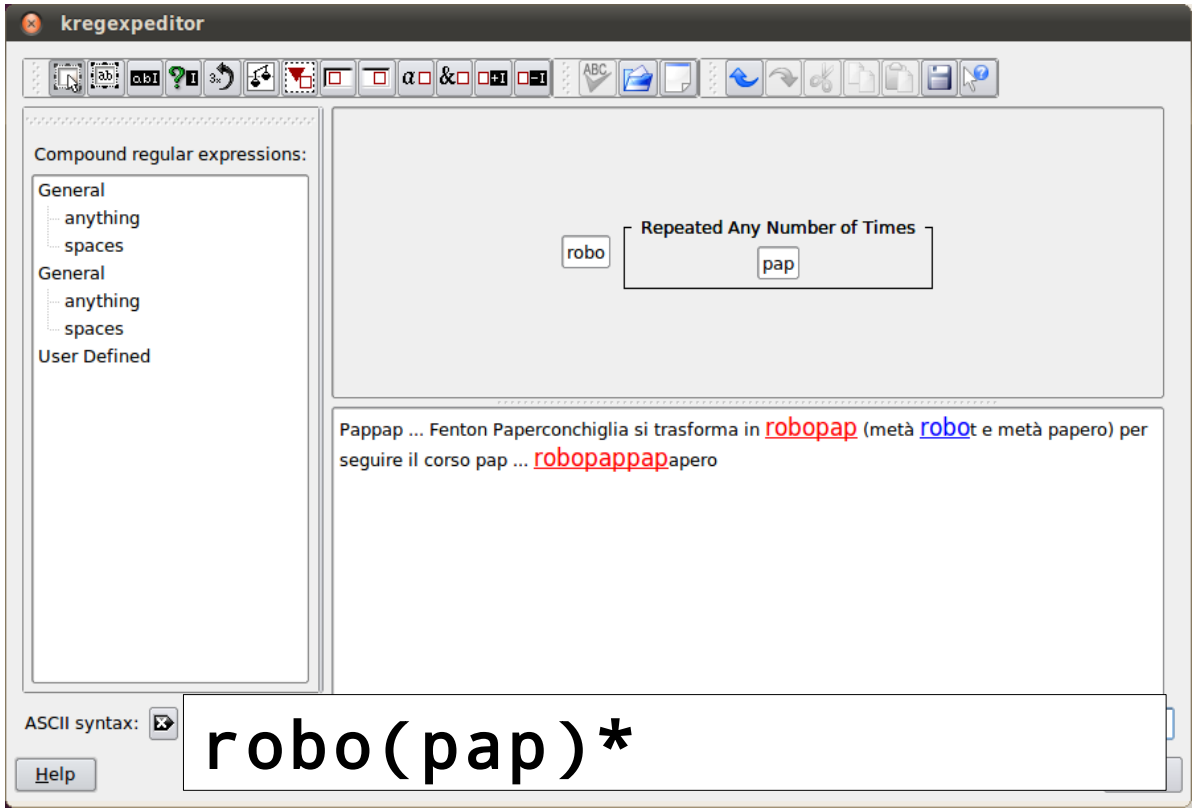
- Aggiungendo il quantificatore '*' possiamo specificare zero o più ripetizioni del termine (o gruppo) che precede



Se non riesco a completare una "ripetizioni", il match si ferma all'ultima volta che siamo arrivati alla Fine

Zero o più (*)

- Aggiungendo il quantificatore '*' possiamo specificare zero o più ripetizioni del termine (o gruppo) che precede



Uno o più (+)

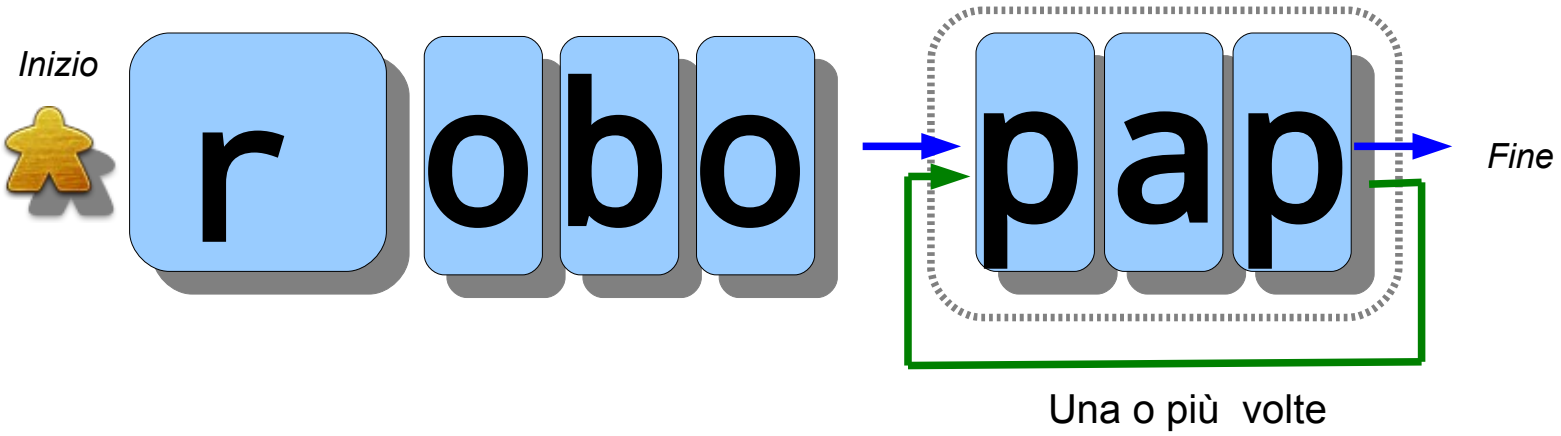
- Aggiungendo il quantificatore '**+**' possiamo specificare una o più ripetizioni del termine (o gruppo) che precede

```
robo(pap)+
```

Cerca la corrispondenza le sequenze che contengono 'robo' seguito da una o più ripetizioni di 'pap'

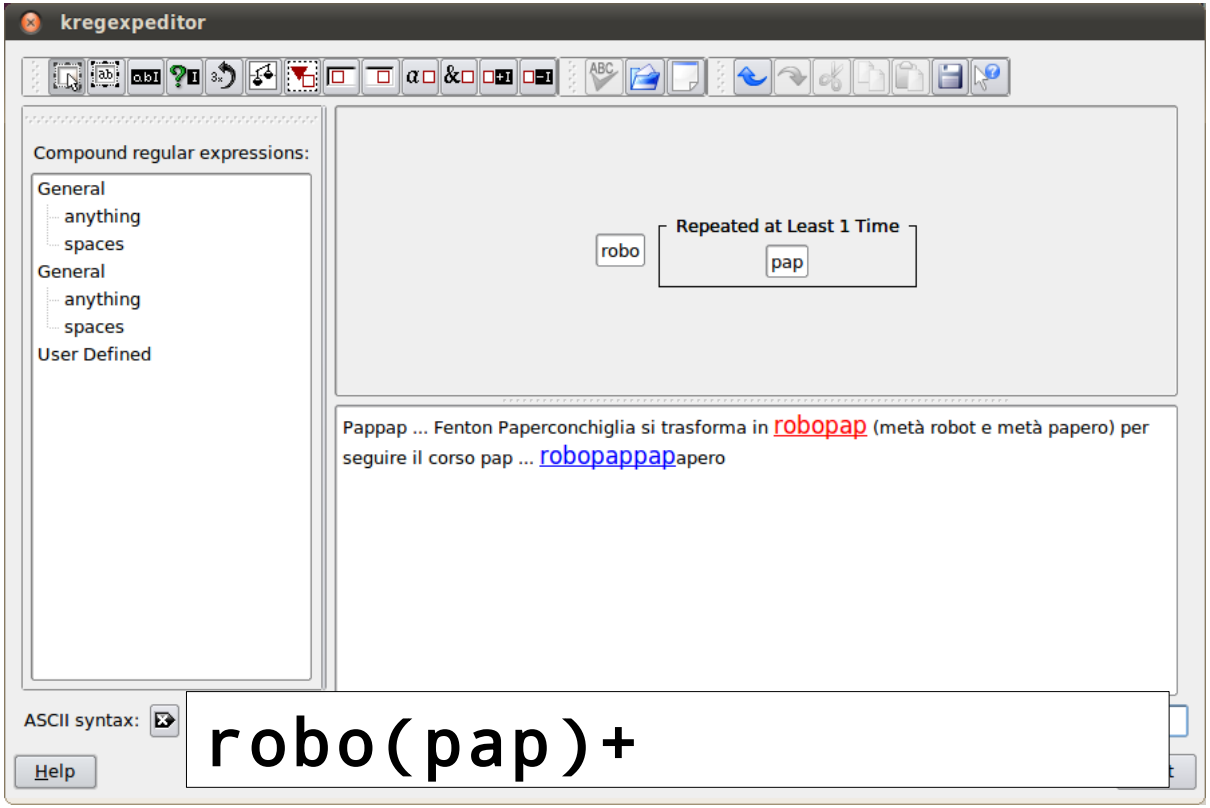
Uno o più (+)

- Aggiungendo il quantificatore '**+**' possiamo specificare una o più ripetizioni del termine (o gruppo) che precede



Uno o più (+)

- Aggiungendo il quantificatore '**+**' possiamo specificare una o più ripetizioni del termine (o gruppo) che precede



Zero o uno (?)

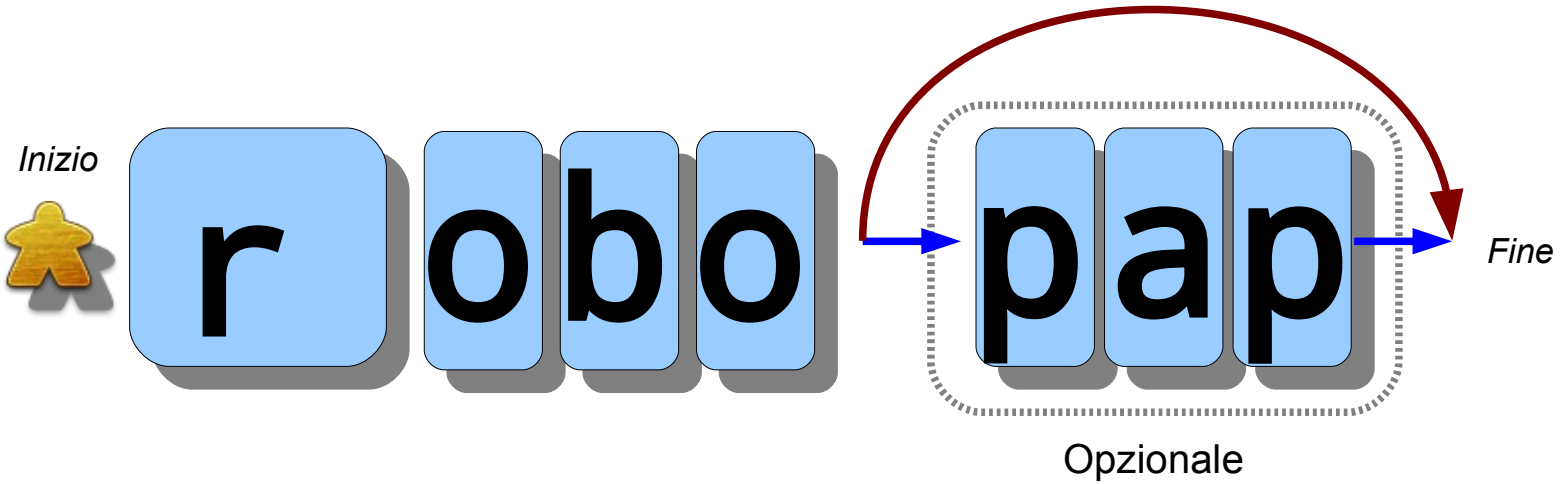
- Aggiungendo il quantificatore '**?**' possiamo specificare che il termine (o gruppo) che precede è opzionale

```
robo(pap)?
```

Cerca la corrispondenza le sequenze che contengono 'robo' seguito opzionalmente da 'pap'

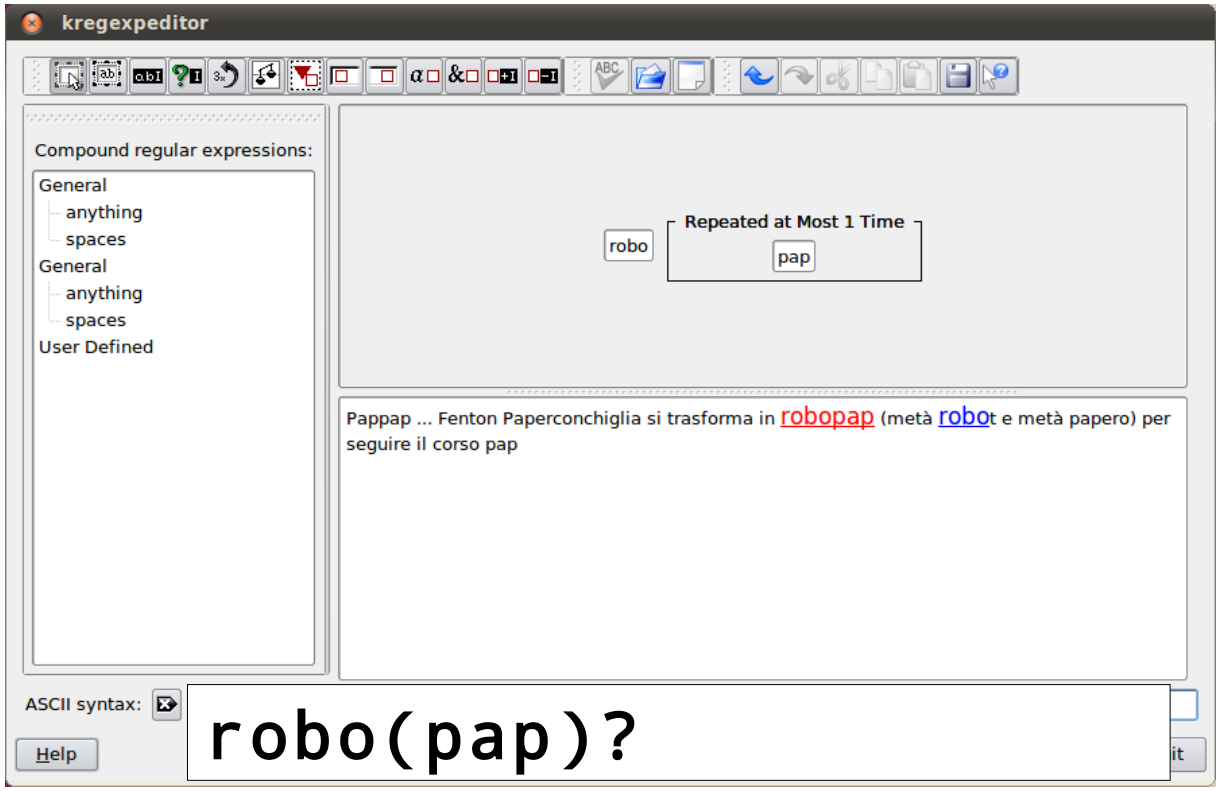
Zero o uno (?)

- Aggiungendo il quantificatore '**?**' possiamo specificare che il termine (o gruppo) che precede è opzionale



Zero o uno (?)

- Aggiungendo il quantificatore '**?**' possiamo specificare che il termine (o gruppo) che precede è opzionale



Numero di ripetizioni '{n}', '{n,m}', {n,}

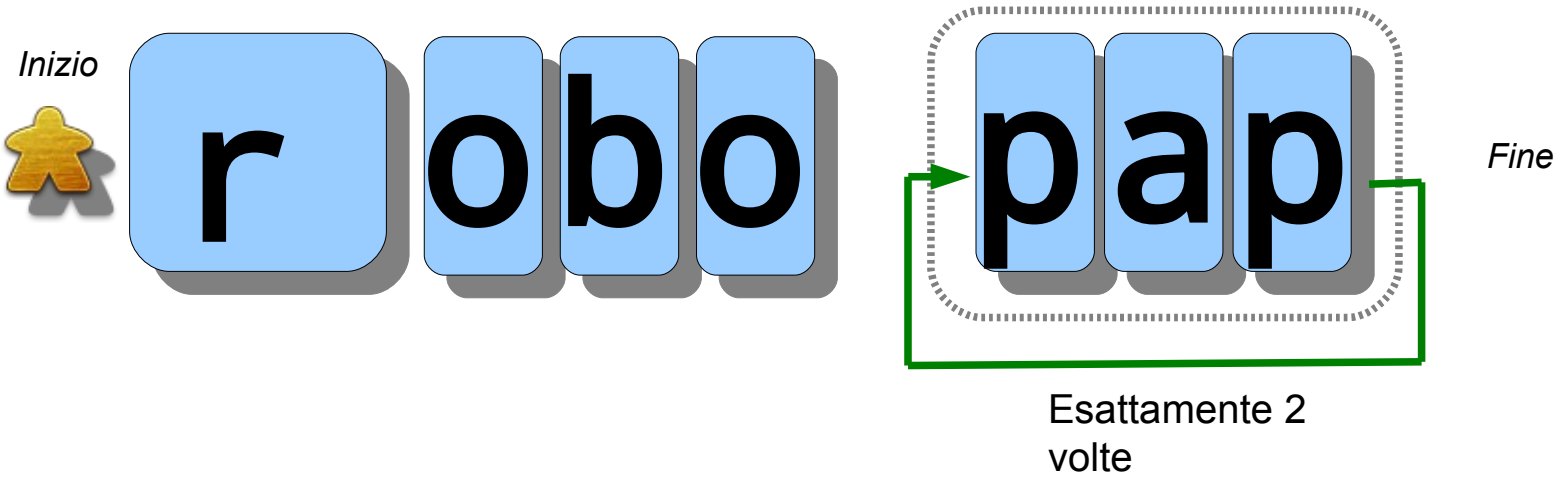
- Posso anche specificare un numero esatto di ripetizioni '**{n}**' , un intervallo '**{n,m}**', o ripetizioni di almeno un numero specificato di volte '**{n,}**' o fino a un massimo di volte '**{,n}**'

```
robo(pap){2}
```

Cerca la corrispondenza le sequenze che contengono 'robo' seguito esattamente due ripetizioni di 'pap'

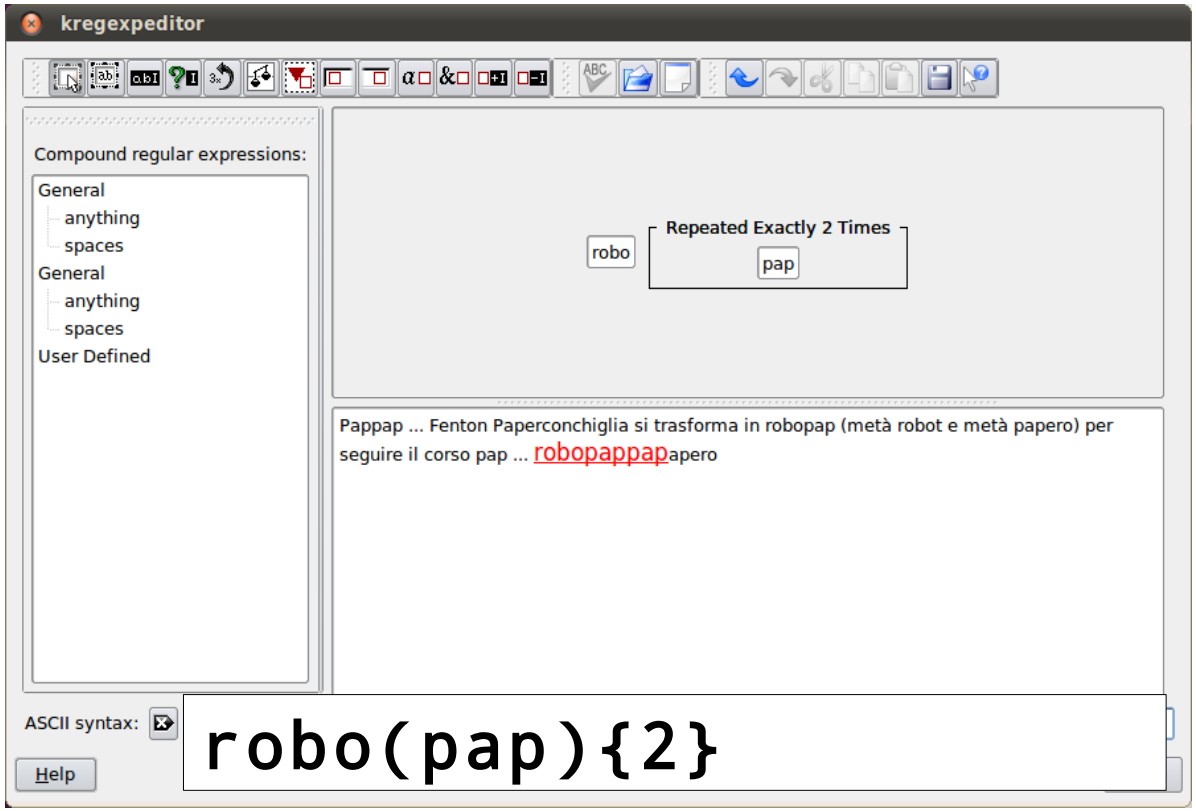
Numero di ripetizioni '{n}', '{n,m}', {n,}

- Posso anche specificare un numero esatto di ripetizioni '**{n}**' , un intervallo '**{n,m}**', o ripetizioni di almeno un numero specificato di volte '**{n,}**' o fino a un massimo di volte '**{,n}**'



Numero di ripetizioni '{n}', '{n,m}', {n,}

- Posso anche specificare un numero esatto di ripetizioni '**{n}**' , un intervallo '**{n,m}**', o ripetizioni di almeno un numero specificato di volte '**{n,}**' o fino a un massimo di volte '**{,n}**'



Numero di ripetizioni '{n}', '{n,m}', {n,}

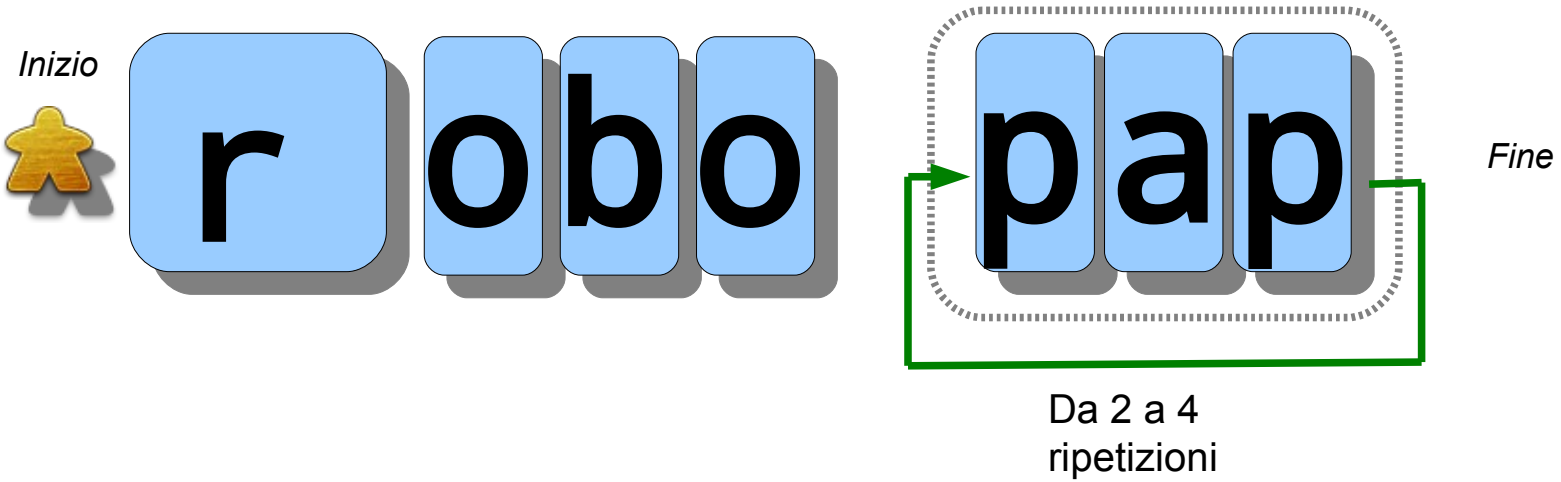
- Posso anche specificare un numero esatto di ripetizioni '**{n}**' , un intervallo '**{n,m}**', o ripetizioni di almeno un numero specificato di volte '**{n,}**' o fino a un massimo di volte '**{,n}**'

```
robo(pap){2,4}
```

Cerca la corrispondenza le sequenze che contengono 'robo' seguito da 2 a 4 ripetizioni di 'pap'

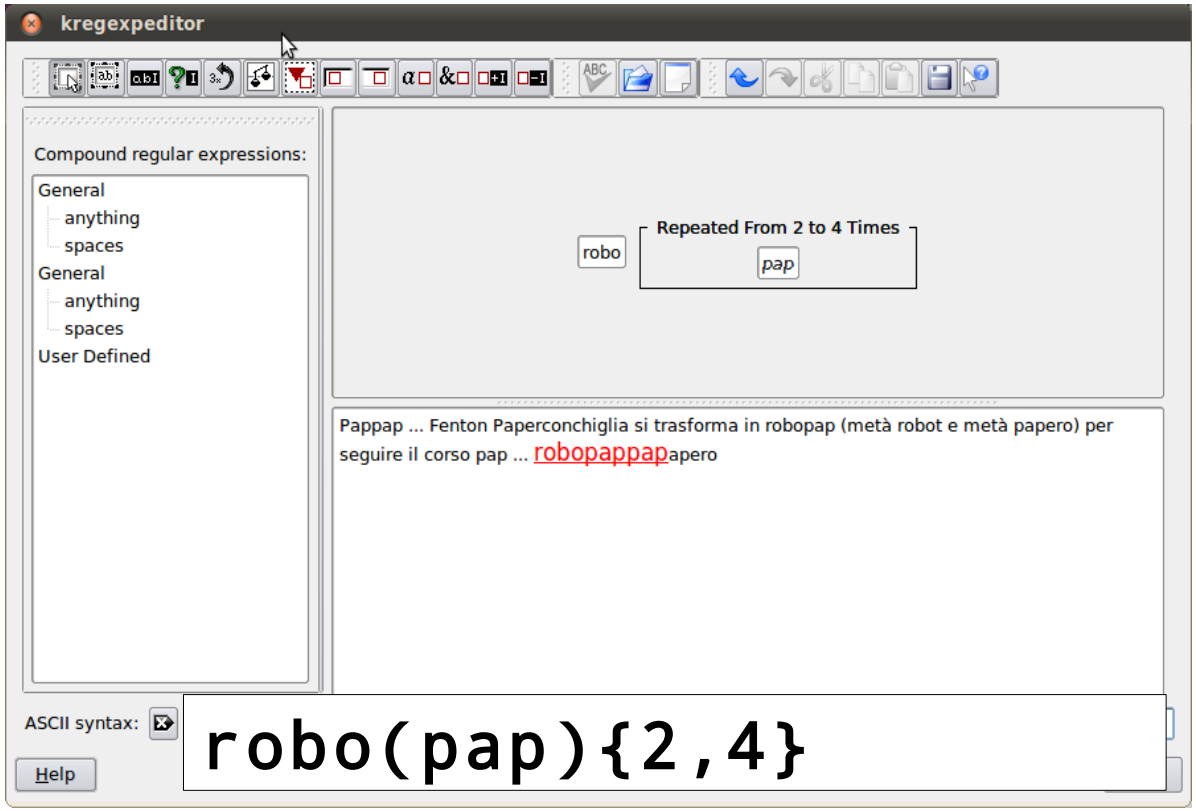
Numero di ripetizioni '{n}', '{n,m}', {n,}

- Posso anche specificare un numero esatto di ripetizioni '**{n}**' , un intervallo '**{n,m}**', o ripetizioni di almeno un numero specificato di volte '**{n,}**' o fino a un massimo di volte '**{,n}**'



Numero di ripetizioni '{n}', '{n,m}', {n,}

- Posso anche specificare un numero esatto di ripetizioni '**{n}**' , un intervallo '**{n,m}**', o ripetizioni di almeno un numero specificato di volte '**{n,}**' o fino a un massimo di volte '**{,n}**'



Numero di ripetizioni '{n}', '{n,m}', {n,}

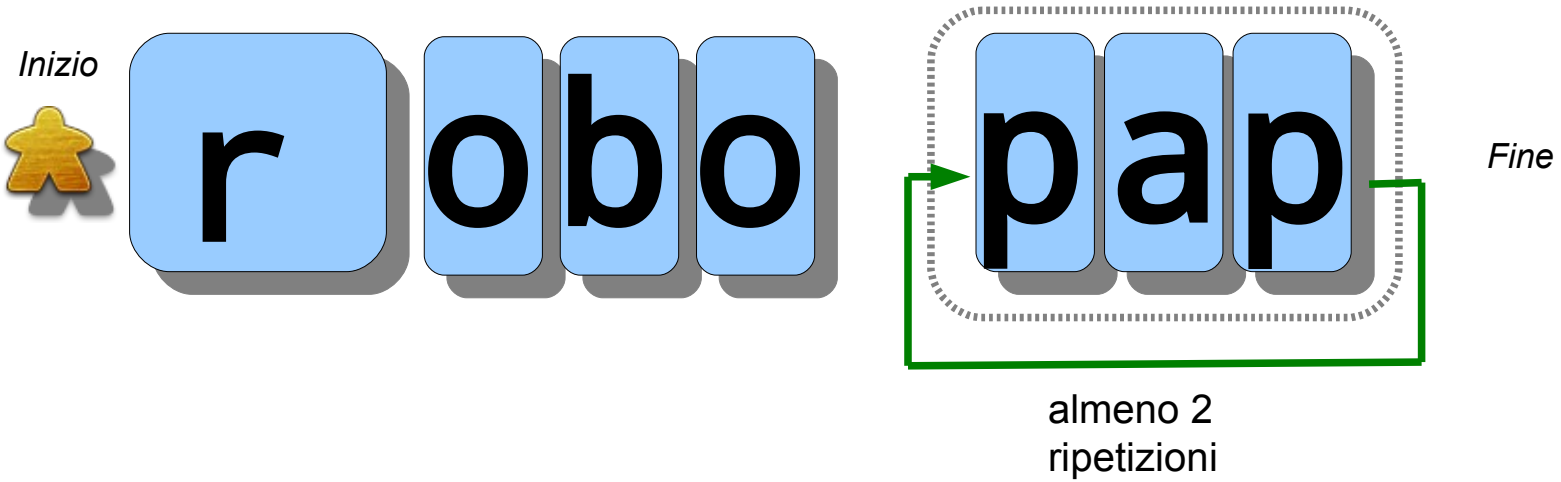
- Posso anche specificare un numero esatto di ripetizioni '**{n}**' , un intervallo '**{n,m}**', o ripetizioni di almeno un numero specificato di volte '**{n,}**' o fino a un massimo di volte '**{,n}**'

```
robo(pap){2,}
```

Cerca la corrispondenza le sequenze che contengono 'robo' seguito da almeno 2 ripetizioni di 'pap'

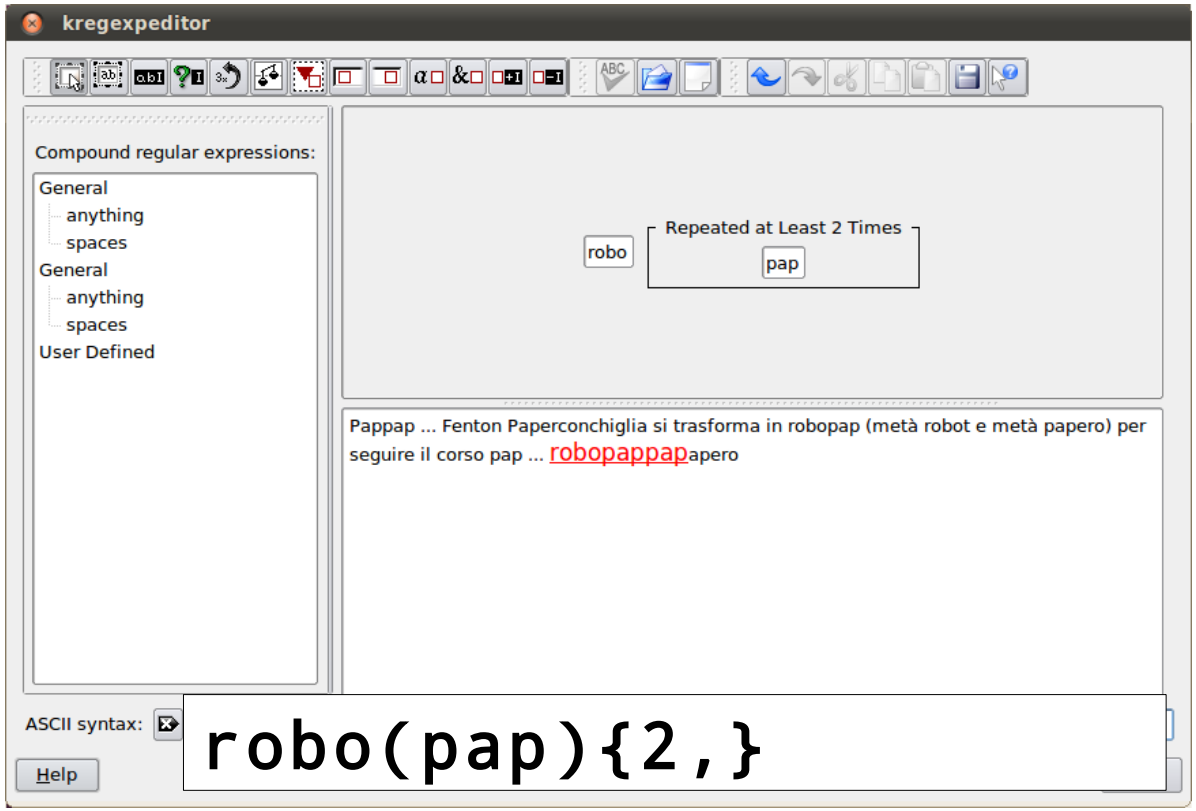
Numero di ripetizioni '{n}', '{n,m}', {n,}

- Posso anche specificare un numero esatto di ripetizioni '**{n}**' , un intervallo '**{n,m}**', o ripetizioni di almeno un numero specificato di volte '**{n,}**' o fino a un massimo di volte '**{,n}**'



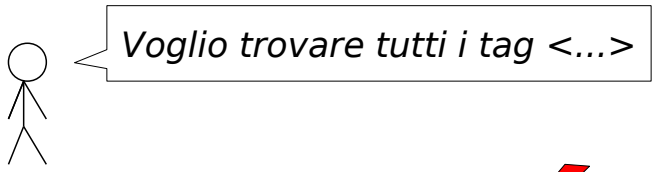
Numero di ripetizioni '{n}', '{n,m}', {n,}

- Posso anche specificare un numero esatto di ripetizioni '**{n}**' , un intervallo '**{n,m}**', o ripetizioni di almeno un numero specificato di volte '**{n,}**' o fino a un massimo di volte '**{,n}**'



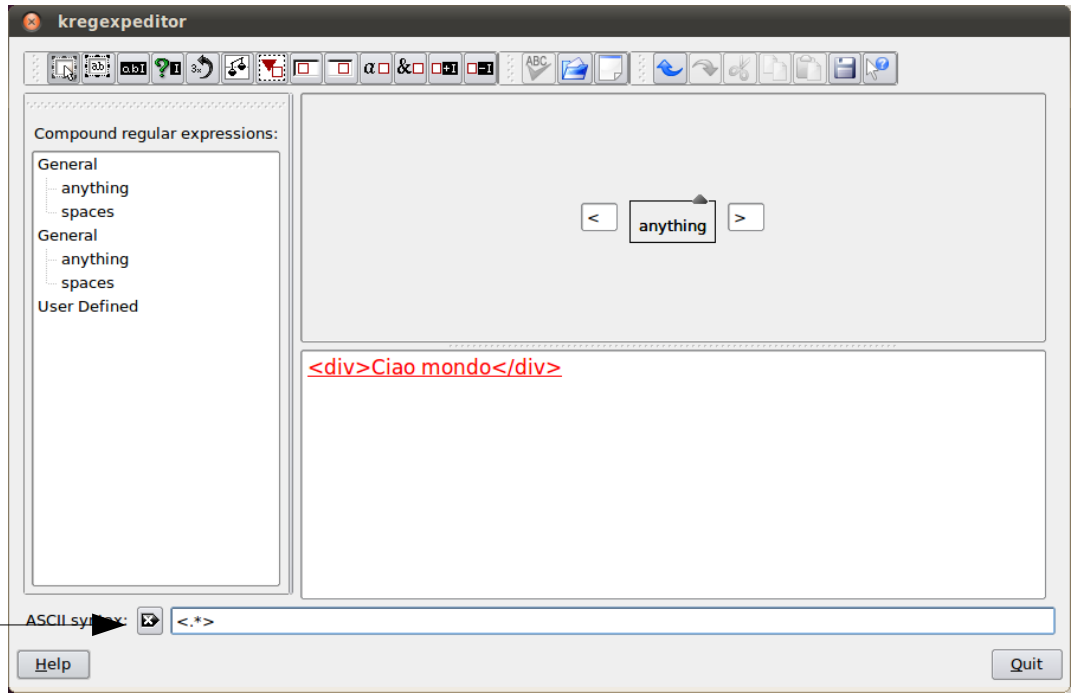
Greediness

- Quando usiamo i quantificatori il motore di ricerca cerca di ripetere il maggior numero di volte possibile
  - Modalità **greedy** (ingorda)
- Questo può portare a degli errori, per esempio:



Il termine '.' ripetuto si "mangia" tutto quello che trova

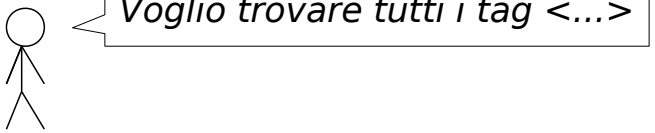
< . * >

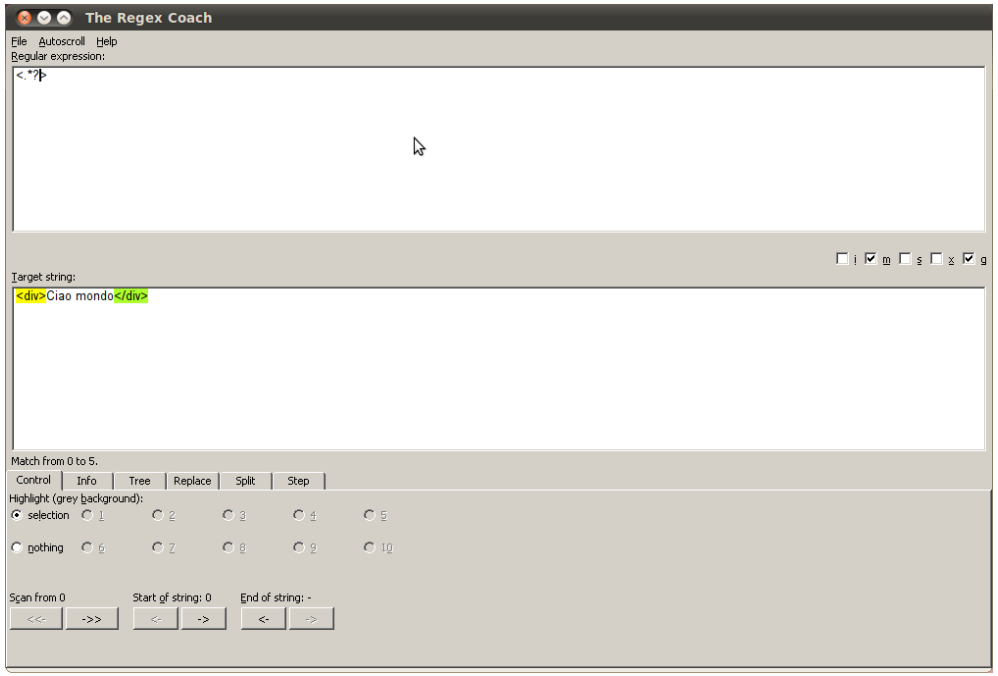


Laziness e back-tracking

- Se il motore delle regex lo supporta, è possibile limitare le ripetizioni al numero minimo possibile, aggiungendo '**?**' dopo il quantificatore
  - Questo meccanismo rallenta sensibilimente il processo di matching, perché il motore deve fare *back-tracking*: ritornare indietro e provare un'altro cammino

<.*?>

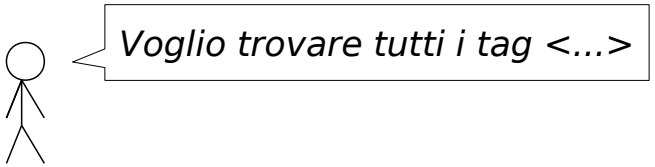


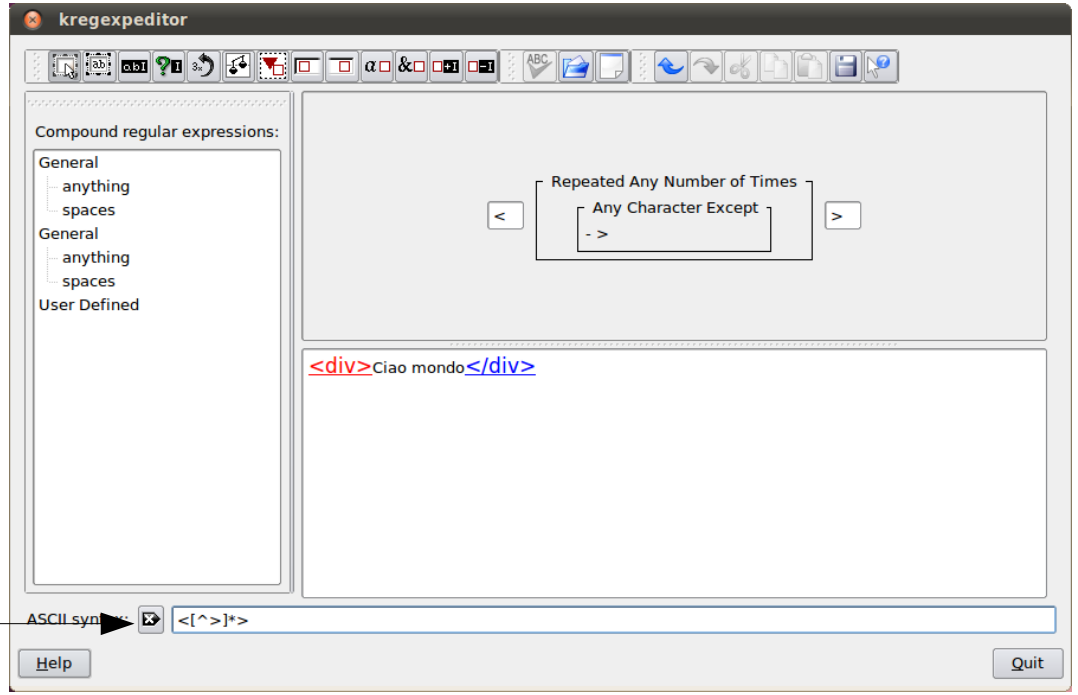


Attenzione! Questa funzione non è supportata da kregexpeditor

Alternativa a laziness...

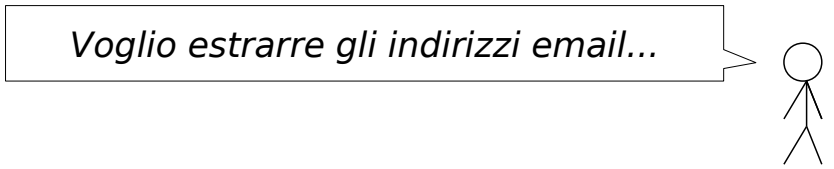
- Per evitare i problemi di performance, o per ovviare alla mancanza della “laziness” nel motore utilizzato a volte è possibile utilizzare degli insiemi di negazione





<[^>]*>

Un esempio pratico

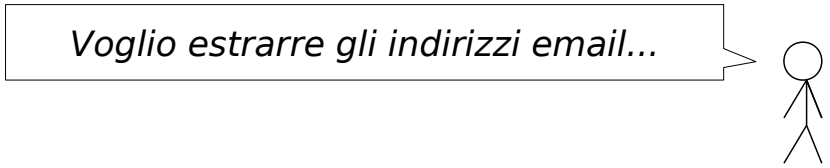


Transport of email across the Internet uses the Simple Mail Transfer Protocol (SMTP), which is defined in Internet standards RFC 5321 and RFC 5322, while mailboxes are most often accessed with the Post Office Protocol (POP) and the Internet Message Access Protocol (IMAP).

Email addresses, such as jsmith@example.org, have two parts. The part before the @ sign is the local-part of the address, often the username of the recipient (jsmith), and the part after the @ sign is a domain name to which the email message will be sent (example.org).

For more information contact neo@matrix.eu or jeff.jeffty@jeff.co.uk

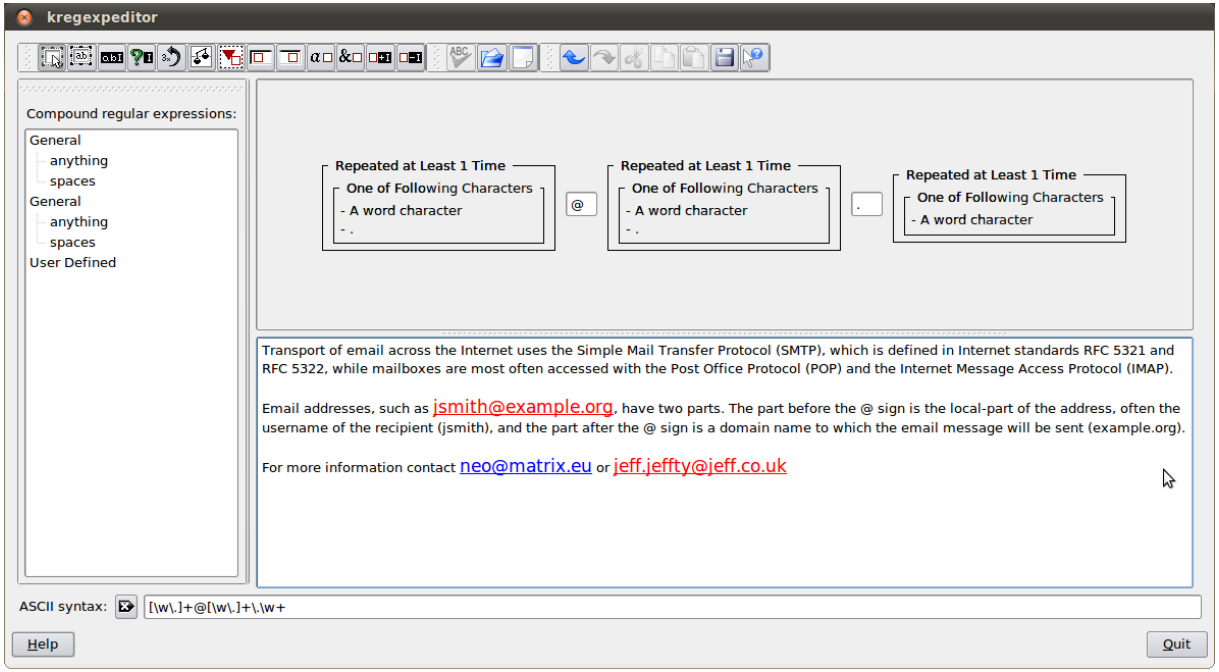
Un esempio pratico




```
[\w\.]+@[\w\.]+\.\w+
```

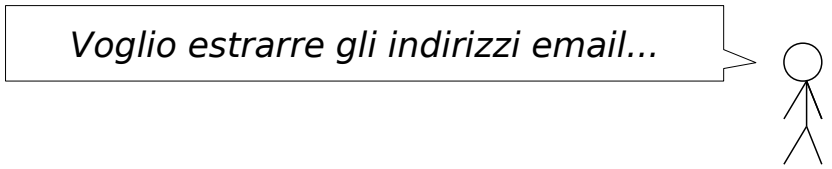
Un esempio pratico


```
[\w\.]+@[\w\.]
```



Questa soluzione è corretta... ma obbliga il motore a fare back-tracking... sapete individuare dove?

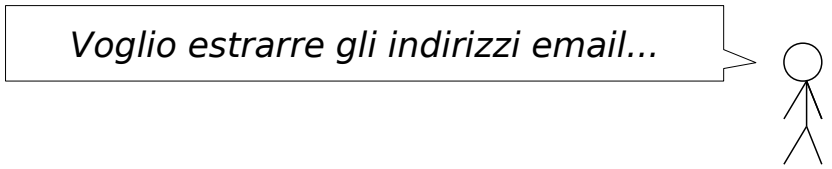
Un esempio pratico




```
[\w\.]+@[\w\.]+\.\w+
```

Se dopo la ripetizione di [\w\.]+ non viene trovato il carattere '.', il motore torna indietro di un carattere nell'input e riprova. Se ancora non funziona, si torna indietro ancora di un carattere, e così via,...

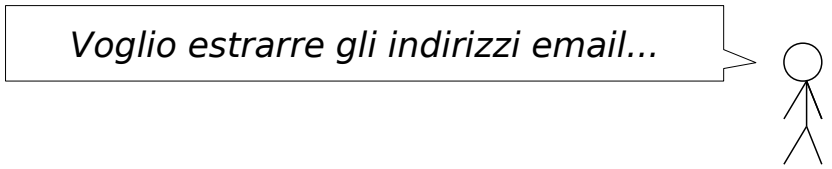
Un esempio pratico



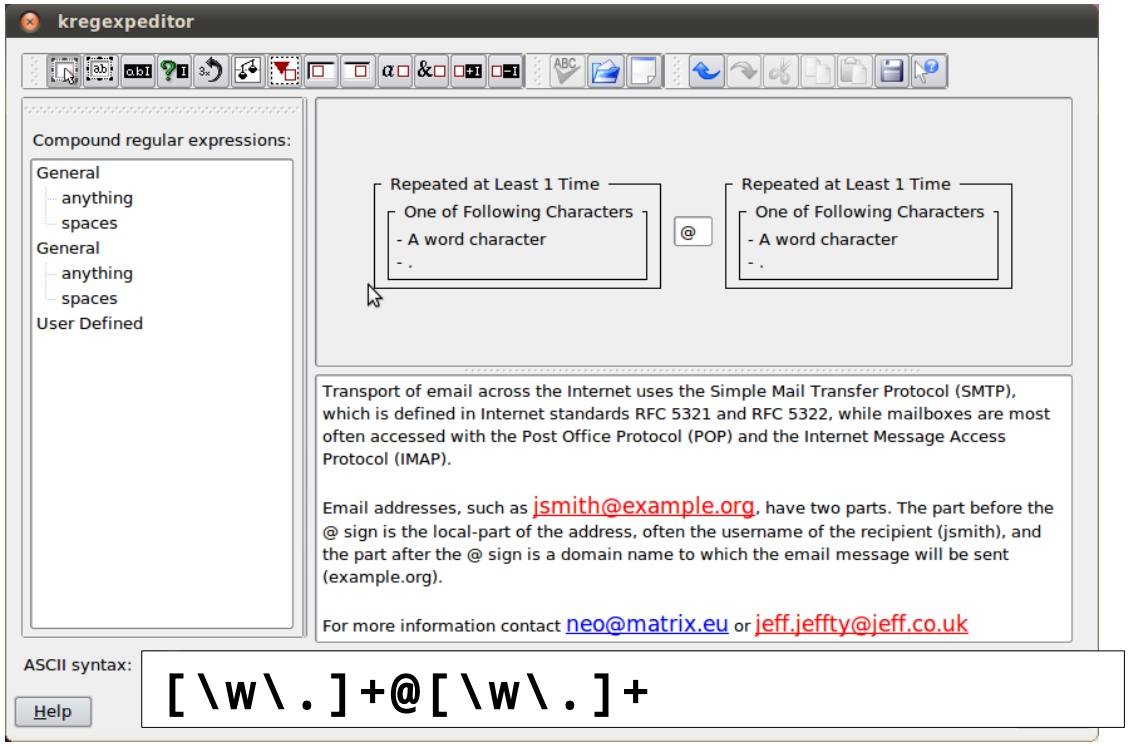
Soluzione senza back-tracking

```
[\w\.]+@[\w\.]+
```

Un esempio pratico

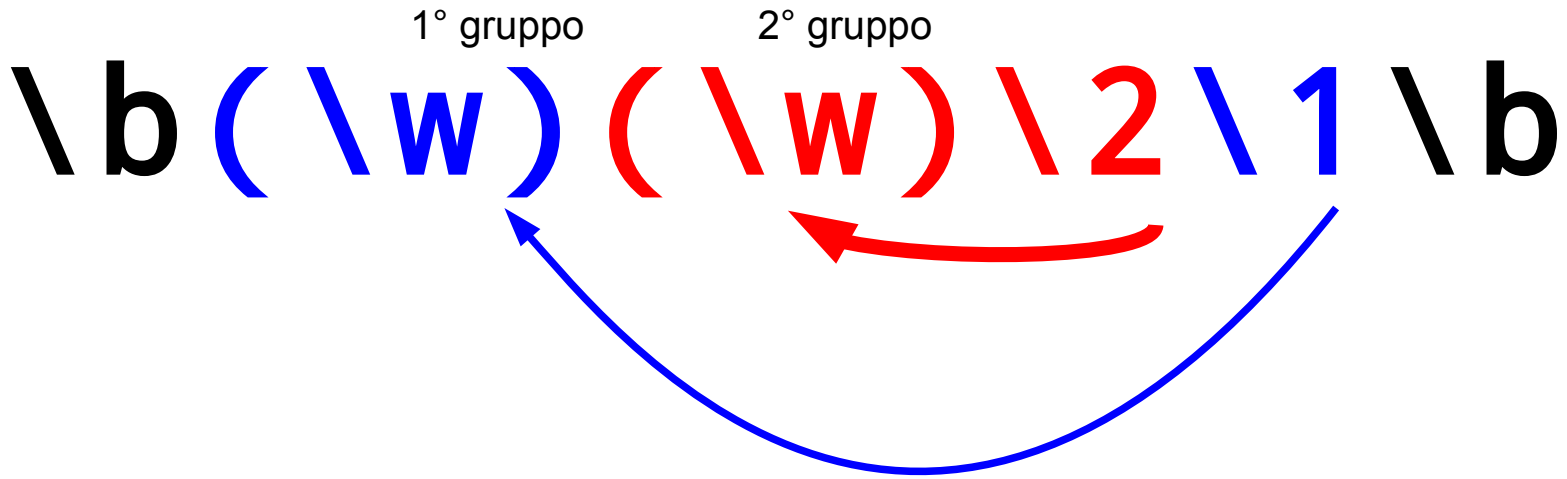


Soluzione senza back-tracking



Back-reference (referenze all'indietro) \n

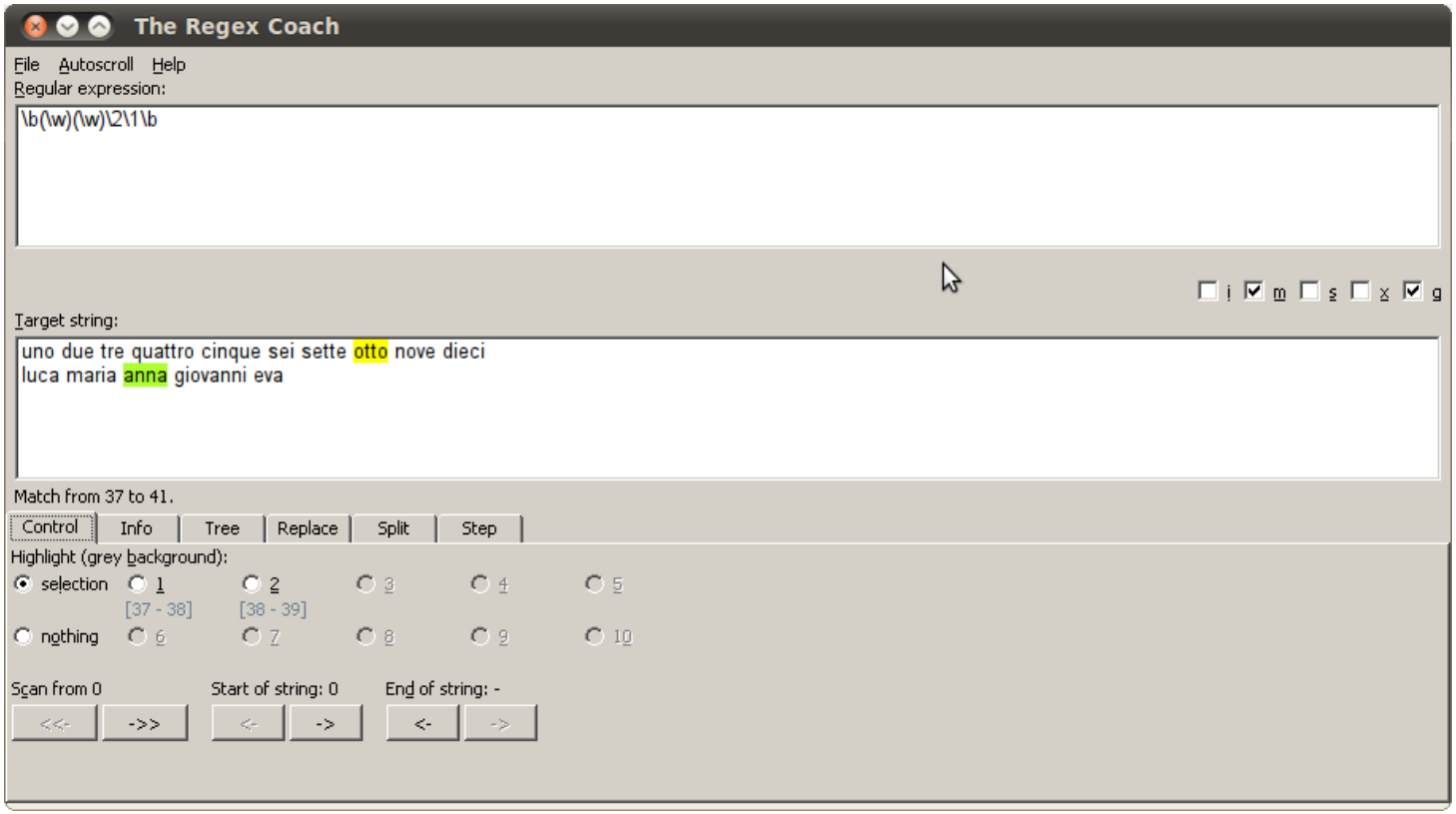
- Quando si utilizzano i gruppi ( ) è possibile referenziarli successivamente nell'espressione regolare, specificando '\' (backslash) poi il loro indice (nell'ordine di definizione)



Questa espressione regolare trova i palindromi di 4 lettere

Back-reference (referenze all'indietro) \n

- Quando si utilizzano i gruppi ( ) è possibile referenziarli successivamente nell'espressione regolare, specificando '\' (backslash) poi il loro indice (nell'ordine di definizione)



Precedenze

- Quando l'espressione viene valutata, valgono le seguenti precedenze:
  - Alternative “|” (più forte)
  - Concatenazione di caratteri
  - Quantificatori (ripetizioni) (più debole)

- Raggruppando i termini in gruppi posso ridefinire le precedenze

grep

- grep può filtrare le righe in input in base a un'espressione regolare
  - Se viene trovato almeno una corrispondenza, la linea viene stampata
  - Noi utilizzeremo grep con l'opzione -E (extended regular expression) che supporta i quantificatori ?, +, {n}, {n,m}, e {n,}
- Attenzione, le parentesi **{ }** **( )** , nonché **|**, **?**, e **+** devono essere preceduti da **** (backslash), se non viene usata l'opzione -E

[X] bash

```
utente@host:~$ grep -E "\bmani\b" testo.txt
gomiti sulle sue ginocchia e con la faccia appoggiata tra le mani, stavo a sentire.
ma domani possiamo esser poveri. E non si misero in cammino a mani vuote.

utente@host:~$ grep -E "\bl[aei]\b" testo.txt
gomiti sulle sue ginocchia e con la faccia appoggiata tra le mani, stavo a sentire.
Gli Egiziani trasmisero ai Romani le preparazioni che permettevano di trasformar le
```

Bash regex =~ / BASH_REMATCH

- Bash include un motore di espressioni regolari
  - Per verificare se una corrispondenza si trova all'interno di una stringa si utilizza **=~** e le doppie parentesi quadre **[[ ]]**

[X] bash

```
utente@host:~$ [[ "The dog is under the aBple" =~ [^d]*dog([^t]*)(.*) ]]
utente@host:~$ echo $?
0
```

  - Quando ho un match posso recuperare le stringhe corrispondenti ai vari gruppi accedendo all'array BASH_REMATCH

[X] bash

```
utente@host:~$ if [[ "The dog is under the aBple" =~ [^d]*dog([^t]*)(.*) ]]; then echo ${BASH_REMATCH[2]}; fi
the aBple
```

  - Posso anche simulare grep:

[X] bash

```
utente@host:~$ cat testo.txt | while read i; do if [[ $i =~ (^..) ]]; then echo $i; fi; done
```

Cheatsheet

- http://www.addedbytes.com/cheat-sheets/regular-expressions-cheat-sheet/