

Espressioni regolari

1 Costruzione di una regex

- `/./`: qualsiasi carattere (le `/` sono solo delimitatori, non fanno parte della regex)
- `[range]`: qualsiasi carattere tra le parentesi
- `[^range]`: qualsiasi carattere che non è tra le parentesi
- `^[range]`: i caratteri del range devono essere all'inizio della stringa
- `[range]$`: i caratteri del range devono essere alla fine della stringa
- `[range1][range2]`: i caratteri del range1 devono essere seguiti dai caratteri del range2
- `[range1]|[range2]`: i caratteri del range1 o del range2
- `(regex)`: gruppo di regex
- `(?:regex)`: gruppo di regex non catturante, non influisce sul riconoscimento dei gruppi

2 Quantificatori

- `x+`: x presente una o più volte
- `x*`: x presente zero o più volte
- `x?`: x presente zero o una volta
- `x{n}`: x presente esattamente n volte
- `x{n,}`: x presente almeno n volte
- `x{n,m}`: x presente da n a m volte

3 Caratteri speciali

- `\t`: carattere tab
- `\n`: carattere newline
- `\s`: qualsiasi carattere di spazio (tab, newline, spazio, ecc.)
- `\S`: qualsiasi carattere che non sia di spazio (equivalente a `[^\s]`)
- `\d`: qualsiasi cifra (equivalente a `[0-9]`)
- `\D`: qualsiasi carattere che non sia una cifra (equivalente a `[^0-9]`)
- `\w`: qualsiasi carattere alfanumerico (equivalente a `[a-zA-Z0-9_]`)
- `\W`: qualsiasi carattere che non sia alfanumerico (equivalente a `[^\w]`)

4 Esempi

4.1 Targhe automobilistiche italiane

$[A-Z]\{2\}\backslash d\{3\}[A-Z]\{2\}$

Spiegazione: vuole due lettere, tre cifre e due lettere

4.2 Numeri in base 16

$0[xX][\backslash dA-Fa-f]^+$

Spiegazione: lo 0 deve essere all'inizio, seguito da x o X, seguito da una o più cifre o lettere da A a F

4.3 Esercizio da esame

Stabilire a quale gruppo appartengono le stringhe

$(\backslash \{ [^ \backslash]^* \backslash \}) | (\backslash d \backslash d / \backslash d \backslash d) | (\backslash w + (\backslash . \backslash w +)^*)$

Spiegazione:

- Il primo gruppo accetta stringhe che iniziano con { e finiscono con }, in mezzo possono esserci da 0 a infiniti caratteri che non siano }
- Il secondo gruppo accetta due cifre, uno / e altre due cifre
- Il terzo gruppo accetta una o più caratteri alfanumerici, seguite da zero o più gruppi di un punto e una o più caratteri alfanumerici
- "_a.": non appartiene a nessun gruppo, quello più vicino è il terzo ma dopo il punto ci vuole almeno un carattere alfanumerico
- "_a._": appartiene al terzo gruppo
- "12/02": appartiene al secondo gruppo
- "12/3": non appartiene a nessun gruppo, quello più vicino è il secondo ma dopo \ci vogliono due cifre
- "{}": appartiene al primo gruppo
- "{,}": appartiene al primo gruppo