# Regular expressions

Pattern matching syntax

## CONCEPTS

A regex is matched left to right: at each position the engine tries the whole pattern and moves on if it fails.

**Anchor** ^ \b \A
↓ then
**Atom** a . \d [abc] (...)
↓ repeated by
**Quantifier** ? * + {n,m}
↓ joined by
**Alternation** a|b

Flavours differ. This sheet follows PCRE2 (Perl-style), also used by PHP and grep -P; JavaScript, Python and POSIX tools lack some features.

## CHARACTERS

**.**
Any char except newline

**\d  \w  \s**
Digit, word char, space

**[abc]**
Any one of a, b, c

**[^abc]**
Any char except a, b, c

**[a-z0-9]**
Ranges inside a set

**\.**
Escape a metacharacter

**\n  \t  \r**
Newline, tab, carriage return

**\p{L}  \p{Lu}**
Unicode letter, uppercase letter

## QUANTIFIERS

*****
Zero or more

**+**
One or more

**?**
Zero or one (optional)

**{n}**
Exactly n times

**{n,m}**
Between n and m times

## QUANTIFIERS (cont.)

**{n,}**
At least n times

***?  +?  ??**
Lazy: as few as possible

## ANCHORS

**^  $**
Start, end of line

**\b**
Word boundary

**\A  \z**
Start, end of text

## GROUPS

**(...)**
Capture group

**a|b**
Alternation

**(?:...)**
Group without capturing

**(?*<name>*...)**
Named capture group

**\1  \2**
Backreference to group n

**\k*<name>***
Backreference by name

## LOOKAROUND

**(?=...)**
Followed by (lookahead)

**(?!...)**
Not followed by

**(?<=...)**
Preceded by (lookbehind)

**(?<!...)**
Not preceded by

## FLAGS

**(?i)**
Ignore case

**(?m)**
^ and $ match at each line

**(?s)**
Dot matches newline