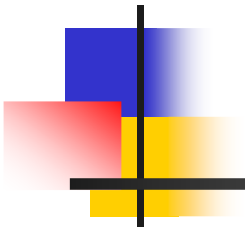
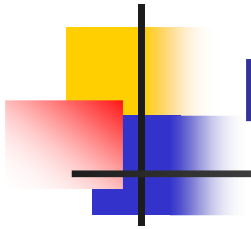


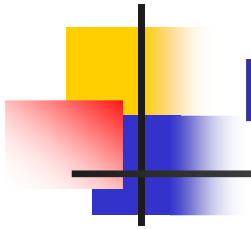
Regular expressions

J.Holvikivi



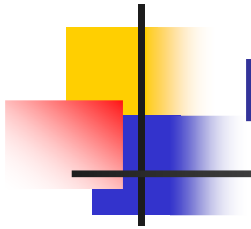
Regular expressions theory

- Commonly used in programming and schema languages to describe **sequences of characters** or **elements**
- Σ: an alphabet (typically Unicode characters or element names)
- a regular expression over Σ is built on the following rules:
  - each atom in Σ is by itself a regular expression
  - if α and β are regular expressions, then the following are also regular expressions: α?, α*, α+, α β, α | β and (α)



Regular expressions rules

- the operators ?, *, + have higher precedence than concatenation, which has higher precedence than |
- σЄΣ matches the string σ
- α? matches zero or one α
- α* matches zero or more α's
- α+ matches one or more α's
- α β matches any concatenation of an α and a β
- α | β matches the union of α and β



Examples

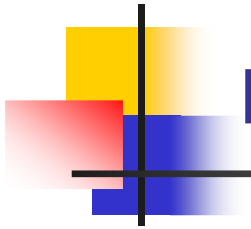
- A regular expression describing **integers**:

  0|-?(1|2|3|4|5|6|7|8|9)(0|1|2|3|4|5|6|7|8|9)*

- A regular expression describing the valid contents of table elements in XHTML:

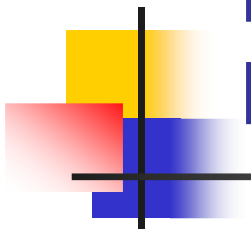
caption? (col* | colgroup*) thead? tfoot? (tbody+ | tr+)

Character	Description	Example
Any character except [\^$.|?*+()	All characters except the listed special characters match a single instance of themselves.	a matches a
\ (backslash) followed by any of [\^$.|?*+(){}	A backslash escapes special characters to suppress their special meaning.	\+ matches +
\Q...\E	Matches the characters between \Q and \E literally, suppressing the meaning of special characters.	\Q+-*/\E matches +-*/
\n, \r and \t	Match an LF character, CR character and a tab character respectively.	
[(opening square bracket)	Starts a character class.	
\d, \w and \s	Shorthand character classes matching digits, word characters, and whitespace.	[\d\s] matches a character that is a digit or whitespace
- (hyphen) except immediately after the opening [	Specifies a range of characters.	[a-zA-Z0-9] matches any letter or digit



Examples: regular expressions in Javascript

- split method:
  - split (/ /) will match spaces
  - transform ="translate (11,22) rotate(90,100,100)"
  - split (/\)/)[0] will return translate(11,22
  - split (/\)/)[1] will return rotate(90,100,100
- reg = /([0-9]+)(\.?) ([0-9]*)/



Regular expressions: Matching an IP address

- complexity vs. exactness:
  \b\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}\b
  will match any IP address just fine, but will also match 999.999.999.999 as if it were a valid IP address.
- To restrict all 4 numbers in the IP address to 0..255:
  \b(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\b
- source: http://www.regular-expressions.info/reference.html