Regular Expression

Search patterns

Cheat sheet

Character classes		Quantifiers & Alternation	
.	any character except newline	a* a+ a?	0 or more, 1 or more, 0 or 1
\w \d \s	word, digit, whitespace	a{5} a{2,}	exactly five, two or more
\W \D \S	not word, digit, whitespace	a{1,3}	between one & three
[abc]	any of a, b, or c	a+? a{2,}?	match as few as possible
[^abc]	not a, b, or c	ab|cd	match ab or cd
[a-g]	character between a & g		
Anchors		**Groups & Lookaround**	
^abc$	start / end of the string	(abc)	capture group
\b	word boundary	\1	backreference to group #1
Escaped characters		(?:abc)	non-capturing group
\. * \\	\ is used to escape special chars. * matches *	(?=abc)	positive lookahead
\t \n \r	tab, linefeed, carriage return	(?!abc)	negative lookahead

Common regex (1)

1.Common email

/^([a-zA-Z0-9._%-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,6})*$/

2. Whole Numbers: /^\d+$/

Decimal Numbers: /^\d*\.\d+$/

Whole + Decimal Numbers: /^\d*(\.\d+)?$/

Negative, Positive Whole + Decimal Numbers

/^-?\d*(\.\d+)?$/

Whole + Decimal + Fractions

/[-]?[0-9]+[,.]?[0-9]*([\/][0-9]+[,.]?[0-9]*)*/

Common regex (2)

3. Alphanumeric without space – /^[a-zA-Z0-9]*$/

Alphanumeric with space – /^[a-zA-Z0-9]*$/

4.Password Strength

Complex: Should have 1 lowercase letter, 1 uppercase letter, 1 number, 1 special character and be at least 8 characters long

/(?=(.*[0-9]))(?=.*[\!@#$%^&*()\\[\]{}\-_+=~`|:;"'<>,./?])(?=.*[a-z])(?=(.*[A-Z]))(?=(.*)).{8,}/

Moderate: Should have 1 lowercase letter, 1 uppercase letter, 1 number, and be at least 8 characters long

/(?=(.*[0-9]))((?=.*[A-Za-z0-9])(?=.*[A-Z])(?=.*[a-z]))^.{8,}$/

Common regex (3)

5. Username

Alphanumeric string that may include _ and – having a length of 3 to 16 characters –

```
/^[a-z0-9_-]{3,16}$/
```

6. URL

Include http(s) Protocol

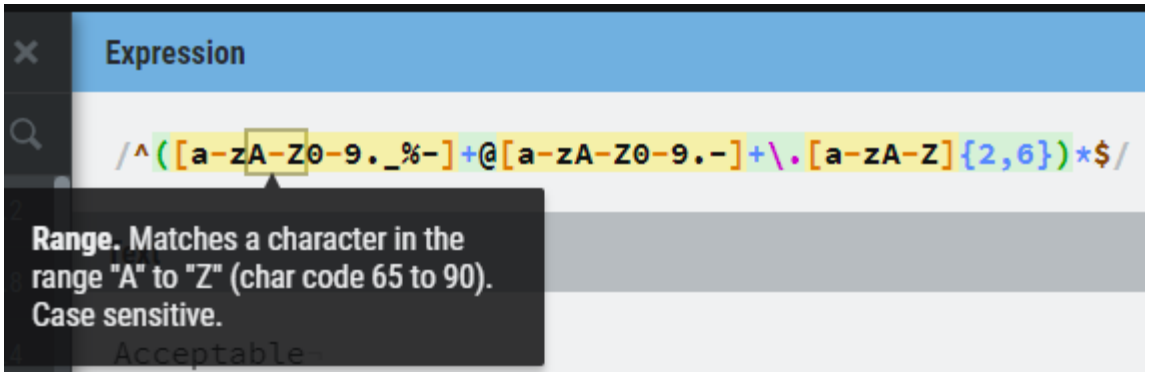
```
/https?:\/\/(www\.)?[-a-zA-Z0-9@:%._\+~#=]{2,256}\.[a-z]{2,6}\b([-a-zA-Z0-9@:%_\+.~#()?&//=]*)/
```

Protocol Optional

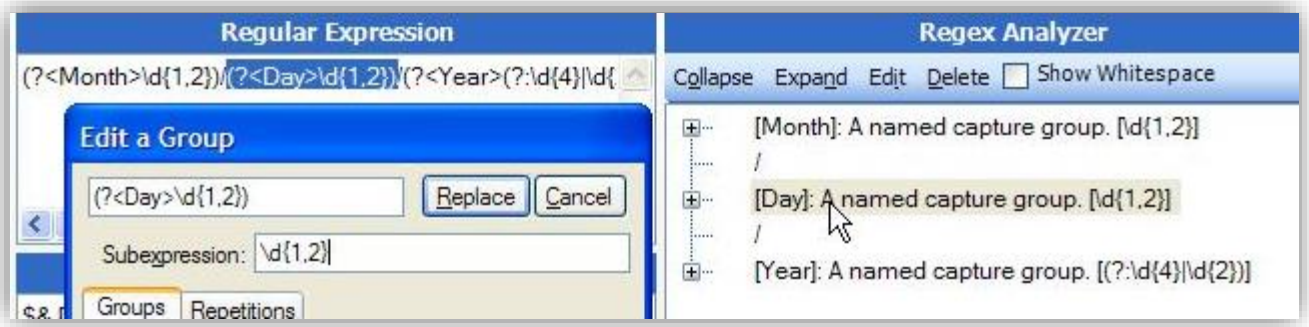
```
/(https?:\/\/)?(www\.)?[-a-zA-Z0-9@:%._\+~#=]{2,256}\.[a-z]{2,6}\b([-a-zA-Z0-9@:%_\+.~#?&//=]*)/
```

Tools for beginner

- regexr.com



- Expresso
  - http://www.ultrapico.com/ExpressoDownload.htm



Regex class in C#

```
Regex regex = new Regex(emailPattern,
    RegexOptions.Compiled | RegexOptions.IgnoreCase);

var valid = regex.IsMatch("tdquang7@gmail.com");
Debug.WriteLine(valid ? "Email is valid" : "Email is not valid");
```

Find repeated words

```
// Define a regular expression for repeated words.
Regex rx = new Regex(@"\b(?<word>\w+)\s+(\k<word>)\b",
  RegexOptions.Compiled | RegexOptions.IgnoreCase);

// Define a test string.
string text = "The the quick brown fox  fox jumps over the lazy dog dog.";

// Find matches.
MatchCollection matches = rx.Matches(text);

// Report the number of matches found.
Console.WriteLine("{0} matches found in:\n   {1}",
                  matches.Count,
                  text);

// Report on each match.
foreach (Match match in matches)
{
    GroupCollection groups = match.Groups;
    Console.WriteLine("'{0}' repeated at positions {1} and {2}",
                      groups["word"].Value,
                      groups[0].Index,
                      groups[1].Index);
}
```