Regular Expressions

Based on Slides from Andreas Spillner

perldoc.perl.org/perlre.html
Regular Expressions

- Compact notation to describe pattern for searching in a text
- Very powerful version of „Search“ or „Search and Replace“
- A regular expression describes a regular language – a set of strings over some alphabet.
- Built into Perl
#!/usr/bin/perl
use strict;
use warnings;

my @f = ("muse", "aal", "falten", ",", "Abt", "aaa");

foreach my $str (@f){
    if ($str =~ m/a/){
        print "$str matches expression.\n";
    }
}
#!/usr/bin/perl
use strict;
use warnings;

my @f = ("muse", "aal", "falten", ",", "Abt", "aaa");

foreach my $str (@f){
    if ($str =~ m/^a/){
        print "$str matches expression.\n";
    }
}

aal matches expression.
aaa matches expression.
#!/usr/bin/perl
use strict;
use warnings;

my @f = ("muse", "aal", "falten", ",", "aasgeier", "aaa");

foreach my $str (@f){
    if ($str =~ m/\w{2}/){
        print "$str matches expression.\n";
    }
}
#!/usr/bin/perl
use strict;
use warnings;

my @f = ("aba", "abba", "abbba", "abbbba");

foreach my $str (@f){
    if ($str =~ m/ab{2,3}a/){
        print "$str matches expression.
    }
}

abba matches expression.
abbba matches expression.
#!/usr/bin/perl
use strict;
use warnings;

my @f = ("muse", "aal", "falten", ",", "Abt", "aaa");

foreach my $str (@f){
    if ($str =~ m/\a/i){ # match case insensitive
        print "$str matches expression.\n";
    }
}

aal matches expression.
falten matches expression.
Abt matches expression.
aasgeier matches expression.
aaa matches expression.
#!/usr/bin/perl
use strict;
use warnings;

my @f = ("muse", "aal", "falten", ",", "Abt", "aaa");

foreach my $str (@f){
    if ($str =~ m/a$/){
        print "$str matches expression.\n";
    }
}

aaa matches expression.
#!/usr/bin/perl
use strict;
use warnings;

my @f = ("muse", "aal", "falten", ",", "Abt", "aaa");

foreach my $str (@f){
    if ($str =~ m/^a/){
        print "$str matches expression.\n";
    }
}

aal matches expression.
falten matches expression.
aaa matches expression.
#!/usr/bin/perl
use strict;
use warnings;

my @f = ("muse", "aal", "falten", ",", "Abt", "aaa");

foreach my $str (@f){
    if ($str =~ m/(a+)/i){
        print "$str matches expression: $1\n";
    }
}

aal matches expression: aa
falten matches expression: a
Abt matches expression: A
aaa matches expression: aaa
#!/usr/bin/perl
use strict;
use warnings;

my @f = ("chr1:400-500", "chr2:1000-2000", "chrX:1-10000");

foreach my $str (@f){
    if ($str =~ m/(.*):(.*)-(.*)/){
        print "In sequence $1 from position $2 to $3.\n";
    }
}

In sequence chr1 from position 400 to 500.
In sequence chr2 from position 1000 to 2000.
In sequence chrX from position 1 to 10000.
#!/usr/bin/perl
use strict;
use warnings;

my @f = ("chrM:400-500", "chrY:1000-2000", "chrX:1-10000");

foreach my $str (@f){
    if ($str =~ m/(chr[YX])/){
        print "$str matches with $1
";
    }
}

ChrY:1000-2000 matches with chrY
ChrX:1-10000 matches with chrX
#!/usr/bin/perl
use strict;
use warnings;

my @f = ("chr1:400-500", "chr2:1000-2000", "chrX:1-10000");

foreach my $str (@f) {
    if ($str =~ m/([a-zA-Z]+)/) {
        print "$str matches with $1\n";
    }
}

chr1:400-500 matches with chr
chr2:1000-2000 matches with chr
chrX:1-10000 matches with chrX
#!/usr/bin/perl
use strict;
use warnings;

my @f = ("chr1:400-500", "chr2:1000-2000", "chrX:1-10000");

foreach my $str (@f)
{
    if ($str =~ m/^chr[^X]/)
    {
        print "$str matches.
"
    }
}
#!/usr/bin/perl
use strict;
use warnings;

my @f = ("chr1:400-500", "chr2:1000-2000", "chrX:1-10000");

foreach my $str (@f){
    if ($str =~ m/(chr\d)/){
        print "$str matches with $1\n";
    }
}

chr1:400-500 matches with chr1
chr2:1000-2000 matches with chr2
#!/usr/bin/perl
use strict;
use warnings;

my @f = ("Mr X", "Mr Bell", "Mr W");

foreach my $str (@f){
    if ($str =~ m/Mr (X|..)/){
        print "$str\n";
    }
}

Mr X
Mr Bell
#!/usr/bin/perl
use strict;
use warnings;

my @f = ("bbbbbbbbb", "ttgbbbbB", "ttgbbbbC", "aaa");

foreach my $str (@f){
    if ($str =~ m/^.{3}(a+|b+|c+)/i){
        print "$str matches expression.\n"
    }
}

What is the output?
#!/usr/bin/perl
use strict;
use warnings;

my @f = ("bbbbbbb", "ttgbbbbB", "ttgbbbbC", "aaa");

foreach my $str (@f){
    if ($str =~ m/^.{3}(a+|b+|c+)/i){
        print "$str matches expression.\n"
    }
}
#!/usr/bin/perl
use strict;
use warnings;

my @f = ("maschine", "machen", "masch");

foreach my $str (@f) {
    if ($str =~ m/s?ch/){
        print "before=$`  match=$&  after=$' \n";
    }
}

before=ma  match=sch  after=ine
before=ma  match=ch  after=en
before=ma  match=sch  after=
#!/usr/bin/perl
use strict;
use warnings;

my @f = ("papa", "geologe", "abba");

foreach my $str (@f){
    if ($str =~ m/^(..).*\1$/){
        print "$str matches.
";
    }
}

papa matches.
gologe matches.
#!/usr/bin/perl
use strict;
use warnings;

my $a;
my $b;
$a = $b = "Three genes on chr1, chr3 and chrX.\n";
$a =~ s/chr/chromosome /;
$b =~ s/chr/chromosome /g;
print $a, $b;

Three genes on chromosome 1, chr3 and chrX.
Three genes on chromosome 1, chromosome 3 and chromosome X.
# Escapes and special characters

<table>
<thead>
<tr>
<th>Escape</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\t</td>
<td>tabulator</td>
</tr>
<tr>
<td>\n</td>
<td>line break</td>
</tr>
<tr>
<td>.</td>
<td>any character</td>
</tr>
<tr>
<td>\s</td>
<td>whitespace</td>
</tr>
<tr>
<td>\S</td>
<td>non whitespace</td>
</tr>
<tr>
<td>\w</td>
<td>word character ([a-zA-Z_])</td>
</tr>
<tr>
<td>\W</td>
<td>not a word character</td>
</tr>
<tr>
<td>\d</td>
<td>digit ([0-9])</td>
</tr>
<tr>
<td>$</td>
<td>a dollar sign</td>
</tr>
<tr>
<td>^</td>
<td>a carret</td>
</tr>
</tbody>
</table>