Perl

- **Practical Extraction and Report Language**
Perl

- Practical Extraction and Report Language
- interpreted language
Perl

- **Practical Extraction and Report Language**
- interpreted language
- mainly for scanning ASCII text files
Perl

- Practical Extraction and Report Language
- interpreted language
- mainly for scanning ASCII text files
- has evolved over time
Perl

- Practical Extraction and Report Language
- interpreted language
- mainly for scanning ASCII text files
- has evolved over time
- often called “The Swiss Army Chainsaw”
Perl

- **Practical Extraction and Report Language**
- interpreted language
- mainly for scanning ASCII text files
- has evolved over time
- often called “The Swiss Army Chainsaw”
- Motto: “There’s more than one way to do it”
Perl data types

• scalars
Perl data types

- scalars
  - integers, reals, strings
Perl data types

- scalars
  - integers, reals, strings
- arrays of scalars
Perl data types

- scalars
  
  integers, reals, strings

- arrays of scalars
  
  dynamic — size can be changed
Perl data types

- scalars
  
  integers, reals, strings

- arrays of scalars
  
  dynamic — size can be changed

- associative arrays of scalars
Namespace

- scalar variable names must begin with a $

```
$x = 100;
$y = ```abcdef'';
```
Namespace

- scalar variable names must begin with a $

  \$x = 100;
  \$y = "abcdef";

- array variables must begin with a @

  @a = (1,1,3,5,8);
  @b = @a;
  @c = @a[1,4];
Arrays

indices range from 0 to $\# <arrayname>$
Arrays

indices range from 0 to $\# <arrayname>$

size can be changed
Arrays

indices range from 0 to $\# <arrayname>$

size can be changed

- push, pop
Arrays

indices range from 0 to $\# <\text{arrayname}>$

size can be changed

- push, pop

  push attaches an element to the end of an array
  pop removes an element from the end of an array and returns it
Arrays

indices range from 0 to $\#<\text{arrayname}>$

size can be changed

- push, pop

  push attaches an element to the end of an array
  pop removes an element from the end of an array and returns it

- shift, unshift
Arrays

indices range from 0 to $\# <\text{arrayname}>$

size can be changed

- push, pop

  push attaches an element to the end of an array
  pop removes an element from the end of an array and returns it

- shift, unshift

  shift removes the first value of an array and returns it
  unshift attaches an element to the beginning of an array
Arrays

array elements are accessed as $<arrayname>[<index>]$

That is, if the array is @ar, then

$ar[0]$ is the first element of the array

$ar[\#ar]$ is the last element of the array

Command line arguments are given in the array @ARG
if-then-else

if (<expression>) { ... };

if (<expression>) { ... } else { ... };

if (<expression>) { ... }
    elsif (<expression>) { ... } else { ... };

Loops

```plaintext
while (condition) {
  ...
};
```

```plaintext
foreach variable (arrayname) {
  ...
};
```
string matching

== ~
string matching

if (``aaab'' =~ /^(a*)b$/) {print ``Yes'';};
string matching

```perl
if (``aaab'' =~ /^(a*)b$/) {print ``Yes'';};

if (``aaab'' =~ /^(a*)b$/) {print $1;};
```
**split**

split ( /<pattern>/,  <string> )

- splits a string into a list (array) of substrings

  on a *pattern*
split

\[
\text{split}( \ /\text{pattern}\/, \ \text{string}\ )
\]

- splits a string into a list (array) of substrings on a pattern

- for instance,

\[
\text{split}(\ /\/, \ `\text{Dran,442-3387,LI-95B,CS Dept'\')};
\]
split

\[
\text{split (} /<\text{pattern}>/, \text{ } <\text{string}> \text{)}
\]

- splits a string into a list (array) of substrings
  on a \textit{pattern}

- for instance,

\[
\text{split}(/,/, \text{''Dran,442-3387,LI-95B,CS Dept''});
\]

produces the array

\[
(\text{''Dran'',''} \text{, ''} \text{''442-3387''',''} \text{, ''} \text{''LI-95B''',''} \text{, ''} \text{''CS Dept''}')
\]
A Sample Program

#!/usr/local/bin/perl
$x = $ARGV[0];
@a = split(/,,/, $x);
@b = ();
foreach $i (@a)
    {if (($i % 2) == 1) {push @b, $i;};
#foreach$i (@b) {print $i, " ";};
print "\n";