

Perl Lesson 8

L615

Spring 2009

1. Sorting in Perl: you can use the `sort` command to sort an array of items

```
@mylist = ("gamma", "beta", "alpha", "alf");
@sortedlist = sort @mylist;
print "@sortedlist\n";
```

2. Sort subroutines

- Perl has its own algorithms to sort arrays
- But we have to tell it what to use when comparing two items—i.e., are we sorting by string value, numerical order, some other criterion?
- We can define a subroutine and then invoke it during a sort call to tell Perl how to compare two variables `$a` and `$b`

```
sub by_number {
    # Perl knows that $a and $b are the items to be compared
    if ($a < $b) {
        -1
    }
    elsif ($a > $b) {
        1
    }
    else {
        0
    }
}
```

```
@sortedlist = sort by_number @numbers;
```

- There's actually an easier way to write this

```
sub by_number { $a <=> $b }
sub reverse_order { $b <=> $a }
```

```
@sortedlist = sort by_number @numbers;
```

- Likewise, for strings

```
# This is the default:
sub alphabetically { $a cmp $b }

sub case_insensitive { "\L$a" cmp "\L$b" }
```

3. Sorting a hash by values

- With the sort routines in place, we can trivially sort a hash by its keys (by getting the keys and sorting them)
- We can also order the keys based on their values

```
sub by_value { $hash_table{$b} <=> $hash_table{$a} }

%hash_table = ("benny" => 5, "bjorn" => 6,
               "agnetha" => 9, "anifrid" => 4);
@ordered_list = sort by_value keys %hash_table;
```

4. Sorting by multiple keys

- Our sorting subroutines can sort in multiple ways

```
sub by_value_and_key {
    $hash_table{$b} <=> $hash_table{$a} # descending order
    or
    $a cmp $b # alphabetically
}

%hash_table = ("benny" => 6, "bjorn" => 6,
               "agnetha" => 9, "anifrid" => 4);
@ordered_list = sort by_value_and_key keys %hash_table;
```