

Notes from Well House Consultants

These notes are written by Well House Consultants and distributed under their Open Training Notes License. If a copy of this license is not supplied at the end of these notes, please visit

*<http://www.wellho.net/net/whcotnl.html>
for details.*

1.1 Well House Consultants

Well House Consultants provides niche training, primarily but not exclusively in Open Source programming languages. We offer public courses at our training centre and private courses at your offices. We also make some of our training notes available under our "Open Training Notes" license, such as we're doing in this document here.

1.2 Open Training Notes License

With an "Open Training Notes License", for which we make no charge, you're allowed to print, use and distribute these notes provided that you retain the complete and unaltered license agreement with them, including our copyright statement. This means that you can learn from the notes, and have others learn from them too.

You are NOT allowed to charge (directly or indirectly) for the copying or distribution of these notes, nor are you allowed to charge for presentations making any use of them.

1.3 Courses presented by the author

If you would like us to attend a course (Java, Perl, Python, PHP, Tcl/Tk, MySQL or Linux) presented by the author of these notes, please see our public course schedule at

<http://www.wellho.net/course/index.html>

If you have a group of 4 or more trainees who require the same course at the same time, it will cost you less to have us run a private course for you. Please visit our onsite training page at

<http://www.wellho.net/course/otc.html>

which will give you details and costing information

1.4 Contact Details

Well House Consultants may be found online at

<http://www.wellho.net>

graham@wellho.net

technical contact

lisa@wellho.net

administration contact

Our full postal address is

404 The Spa

Melksham

Wiltshire

UK SN12 6QL

Phone +44 (0) 1225 708225

Fax +44 (0) 1225 707126

Lists

A variable might contain a series of words and, if it does, it can be treated in a special way, as a list through which your program can iterate. This module describes how such lists are handled, and goes on to cover list-related commands. Several sample programs show you how to make best use of lists.

<i>How lists are handled in Tcl</i>	4
<i>List-related commands</i>	5
<i>Sample programs using lists</i>	5

A list is a sequence of values separated by white space.

There are some list-related commands which can be used to manipulate lists:

llength returns the number of elements in a list

lrange returns part of a list (given range of indices)

Thus:

```
[graham@chapatti folder]$tcl
tcl>set ld "a b c one two three"
tcl>llength $ld
6
tcl>lrange $ld 2 4
c one two
tcl>exit
[graham@chapatti folder]$
```

Lists are great (and easy!) for storing a number of words, and looping through them:

```
#!/usr/bin/tcl

set days "Sunday Monday Tuesday Wednesday Thursday Friday Saturday"

foreach day $days {
    puts $day
}
```

And runs:

```
[graham@chapatti folder]$tcl days
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
[graham@chapatti folder]$
```

However, things start to get more complex when we want spaces, quotes, etc., to be included in lists. Rather than maintaining a list using the commands we've already learnt, we'll probably use other commands that are provided.

You're likely to find that arrays and regular expressions are useful alternatives to using lists many times. In fact, Tcl lists are surprisingly lightweight structures, and if you're running out of steam with them you should be using something else!

2.1 How lists are handled in Tcl

Lists have the same structure as commands; a series of space-separated values, but with groupings such as { and } to let you include white space within an element.

```
[graham@chapatti folder]$tcl
tcl>set names "{Napoleon Bonaparte} {Duke of Wellington}"
tcl>llength $names
2
tcl>set names
{Napoleon Bonaparte} {Duke of Wellington}
tcl>
```

This list has two elements, or can be handled as a single variable if wished.

Writing lists by hand, getting all the quoting right, is a possible but tricky task if you're unsure where there might be spaces or special characters. Therefore, a `list` command is provided to help you in the process, and it provides automatic quoting.

```
tcl>set morenames [list "Brent Welch" "John Ousterhout" "Paul Raines"]
tcl>set morenames
{Brent Welch} {John Ousterhout} {Paul Raines}
tcl>
```

Note that the initially provided double quotes were removed by the Tcl parser as it interpreted the command, and the `{` and `}` were inserted by the `list` command. It doesn't matter how the initial protection was applied, `list` always switches it to the `{` and `}` syntax.

```
tcl>set further [list {Mark Harrison} "Michael McLennan" Don\ Libes]
tcl>set further
{Mark Harrison} {Michael McLennan} {Don Libes}
tcl>
```

2.2 List-related commands

Creating and modifying lists

As well as creating lists from first principles, you can construct and modify lists using:

- list** provides automatic quoting (see above)
- lappend** adds second and subsequent arguments to list named as first argument
- linsert** inserts elements into a list at a given index position, returns a new list
- lreplace** replaces elements in a list (first argument) with index numbers in a range (second and third arguments) with subsequent arguments
- split** splits a string into list elements and returns the list. If called with two arguments, the second argument is the separator character

Extracting information from a list

Information can be returned from lists using

- lindex** first parameter, the list; second parameter, the index number of the element to be returned
- llength** returns the length of a list
- join** merges the elements of a list, separating each with the second parameter
- lsearch** search a list for a value

Manipulating lists to create other lists

- concat** takes multiple lists as parameters and returns a single joined list
- lrange** returns part of the incoming list, from start index number to end index number
- lsort** sort an incoming list and return a new sorted list

2.3 Sample programs using lists

Reading an access log file

This program reads an access log file and prints a count of the number of times that each visiting host computer has accessed the server.

```
#!/usr/bin/tcl

# Read and analyse an access log file

set fhandle [open seal_log r]

while {[scan [gets $fhandle] %s host] >0} {
    lappend hlist $host
}

set order [lsort $hlist]
lappend order 999

set oldhost 999
set counter 0

foreach host $order {
    if {[string compare $host $oldhost] == 0} {
        incr counter
    } else {
        if {$counter > 0} {
            puts "$oldhost $counter"
        }
        set oldhost $host
        set counter 1
    }
}
}
```

In operation:

```
[graham@chapatti folder]#cr2
aviemore 2
balti 31
barb 2
chapatti 4
curry 2
dhansak 2
dupiaza 1
lecht 2
localhost 4
otter 34
samosa 1
walrus 1
[graham@chapatti folder]$
```

List handling points of note in this program:

lappend hlist \$host

Add the string in **\$host** onto the end of the list in **\$hlist**. Note that **\$hlist** doesn't exist the first time through; that's fine – the list is created.

set order [lsort \$hlist]

Sort the elements of the list; a new list is returned, which we have saved into order.

By default, sorting is in ascii order, but you can specify an option if you wish:

-ascii **-real**
-integer **-dictionary**

and also

```
-increasing           -decreasing  
    lappend order 999
```

We've added an extra element onto the end of our list to force the upcoming loop to go round one extra time. This will cause it to flush out the last real data!

```
foreach host $order {
```

Loop through each element of the list in **\$order** in turn, putting each element in turn into **\$host** and performing the commands in the block.

**Exercise**

Write a program to read the file `stdcodes` from your coursework directory. Each line of this file contains a dialling code and a place name. Read the file line-by-line and make up a list of all the possible dialling codes for Florida. Print out that list.

License

*These notes are distributed under the **Well House Consultants Open Training Notes License**. Basically, if you distribute it and use it for free, we'll let you have it for free. If you charge for its distribution of use, we'll charge.*

3.1 Open Training Notes License

Training notes distributed under the **Well House Consultants Open Training Notes License** (WHCOTNL) may be reproduced for any purpose PROVIDE THAT:

- This License statement is retained, unaltered (save for additions to the change log) and complete.
- No charge is made for the distribution, nor for the use or application thereof. This means that you can use them to run training sessions or as support material for those sessions, but you cannot then make a charge for those training sessions.
- Alterations to the content of the document are clearly marked as being such, and a log of amendments is added below this notice.
- These notes are provided "as is" with no warranty of fitness for purpose. Whilst every attempt has been made to ensure their accuracy, no liability can be accepted for any errors of the consequences thereof.

Copyright is retained by Well House Consultants Ltd, of 404, The Spa, Melksham, Wiltshire, UK, SN12 6QL - phone number +44 (1) 1225 708225. Email contact - Graham Ellis (graham@wellho.net).

Please send any amendments and corrections to these notes to the Copyright holder - under the spirit of the Open Distribution license, we will incorporate suitable changes into future releases for the use of the community.

If you are charged for this material, or for presentation of a course (Other than by Well House Consultants) using this material, please let us know. It is a violation of the license under which this notes are distributed for such a charge to be made, except by the Copyright Holder.

If you would like Well House Consultants to use this material to present a training course for your organisation, or if you wish to attend a public course is one is available, please contact us or see our web site - <http://www.wellho.net> - for further details.

Change log
Original Version, Well House Consultants, 2004

Updated by: _____ on _____

Updated by: _____ on _____

Updated by: _____ on _____

Updated by: _____ on _____

Updated by: _____ on _____

Updated by: _____ on _____

Updated by: _____ on _____

License Ends.