KUAS
KYOTO UNIVERSITY OF ADVANCED SCIENCE
京都発世界人財

京都先端科学大学

Information Literacy 09

quoting, expansion, conditionals

Ian Piumarta

Faculty of Engineering, KUAS

reminder: fonts tell you what is literal and what is not

things in `typewriter font` are literal

- enter them exactly as written

things in *italic font* are placeholders in templates

- you do not type them as written
- **replace** them with what they **describe**

template:

name=*value*
`echo $`*name*

example:

```
answer=42
echo $answer
```

(replacing *name* with `answer` and *value* with 42 in the template)

self-preparation review: variables are named values

variable names: letter, followed by letters or digits

- '_' is a letter

setting a variable: *name*=*value*

review: quotation ‘disables’ special characters

quotation:

- `'single quotes'` protect all characters
- `"double quotes"` protect spaces, wildcards

review: values can be modified during expansion

variable expansion: `${name}` or `"${name}"`

- expands to the value of the variable *name*

trimming prefixes and suffixes

- remove prefixes with `${`*`name#prefix`*`}`
- remove suffixes with `${`*`name%suffix`*`}`

review: parameters are numbered, not named

script parameters:

- `$1` is first argument, `$2` is second, etc.
- `$@` is a list of all the arguments
- `$#` is the number of arguments

review: substitutions evaluate expressions, run programs

arithmetic substitution: `$((expression))`

command substitution: `$(command)`

review: `while` loop is controlled by a condition

```
while CONDITION
do
    body: one or more commands
    to run until CONDITION fails
done
```

```
while CONDITION ; do COMMANDS ; done
```

review: `if` conditionally executes commands

```
if CONDITION
then
   one or more COMMANDS
   to run if CONDITION succeeds
fi
```

```
if CONDITION
then
   one or more COMMANDS
   to run if CONDITION succeeds
else
   one or more COMMANDS
   to run if CONDITION fails
fi
```

```
if CONDITION ; then COMMANDS ; fi
```

```
if CONDITION ; then COMMANDS1 ; else COMMANDS2 ; fi
```

review: `test` performs tests and comparisons

`test` is useful as a *CONDITION* for `if` and `while`

test	*succeeds if...*
`test -f` *`filename`*	*filename* is a regular file
`test -d` *`filename`*	*filename* is a directory
`test` *`n1`* `-lt` *`n2`*	$n1 < n2$
`test` *`n1`* `-eq` *`n2`*	$n1 = n2$
`test` *`n1`* `-ne` *`n2`*	$n1 \neq n2$

next week we change topic completely

10 The Internet – what it is, how it works
11 Data mobility – moving data and computation around
12 The World Wide Web – what it is, how it works
13 Content creation – making web pages
14 Web applications and cloud services – risks, benefits
15 Safety and security – protecting yourself and your data