The if-clause

Synopsis

```
if <LIST>; then
  <LIST>
fi
```

```
if <LIST>; then
  <LIST>
else
  <LIST>
fi
```

```
if <LIST>; then
  <LIST>
elif <LIST>; then
  <LIST>
else
  <LIST>
fi
```

Description

The `if` -clause can control the script's flow (what's executed) by looking at the exit codes of other commands.

All commandsets `<LIST>` are interpreted as command lists, thus they can contain the whole palette from simple commands over pipelines to compound commands (and their combination) as condition.

Operation

The **`if <LIST>`** commands are executed. If the exit code was 0 (TRUE) then the **`then <LIST>`** commands are executed, otherwise the **`elif <LIST>`** commands and their **`then <LIST>`** statements are executed in turn, if all down to the last one fails, the **`else <LIST>`** commands are executed, if one of the `elif` succeeds, its `then` thread is executed, and the `if` -clause finishes.

Basically, the `elif` clauses are just additional conditions to test (like a chain of conditions) if the very first condition failed. If one of the conditions fails, the `else` commands are executed, otherwise the commands of the condition that succeeded.

Examples

Check if a specific user exists in /etc/passwd 😃

```
if grep ^myuser: /etc/passwd >/dev/null 2>&1; then
  echo "Yes, it seems I'm real"
else
  echo "Uh - am I a ghost?"
fi
```

Mount with check

```
if ! mount /mnt/backup >/dev/null 2>&1; then
  echo "FATAL: backup mount failed" >&2
  exit 1
fi
```

Multiple commands as condition

It's perfectly valid to do:

```
if echo "I'm testing!"; [ -e /some/file ]; then
  ...
fi
```

The exit code that dictates the condition's value is the exit code of the very last command executed in the condition-list (here: The `[ -e /some/file ]`)

A complete pipe as condition

A complete pipe can also be used as condition. It's very similar to the example above (multiple commands):

```
if echo "Hello world!" | grep -i hello >/dev/null 2>&1; then
  echo "You just said 'hello', yeah?"
fi
```

Portability considerations

See also

- Internal: the classic test command

Discussion