SC2013 – ShellCheck Wiki

See this page on GitHub (https://github.com/koalaman/shellcheck/wiki/SC2013)

To read lines rather than words, pipe/redirect to a while read loop.

Problematic code:

```
for line in $(cat file | grep -v '^ *#')
do
  echo "Line: $line"
done
```

Correct code:

```
grep -v '^ *#' < file | while IFS= read -r line
do
  echo "Line: $line"
done
```

or without a subshell (bash, zsh, ksh):

```
while IFS= read -r line
do
  echo "Line: $line"
done < <(grep -v '^ *#' < file)
```

or without a subshell, with a pipe (more portable, but write a file on the filesystem):

```
mkfifo mypipe
grep -v '^ *#' < file > mypipe &
while IFS= read -r line
do
  echo "Line: $line"
done < mypipe
rm mypipe
```

NOTE: `grep -v '^ *#'` is a placeholder example and not needed. To just loop through a file:

```
while IFS= read -r line
do
  echo "Line: $line"
done < file
# or: done <<< "$variable"
```

Rationale:

For loops by default (subject to `$IFS`) read word by word. Additionally, glob expansion will occur.

Given this text file:

```
foo *
bar
```

The for loop will print:

```
Line: foo
Line: aardwark.jpg
Line: bullfrog.jpg
...
```

The while loop will print:

```
Line: foo *
Line: bar
```

Exceptions

If you do want to read word by word, you can set `$IFS` appropriately and disable globbing with `set -f`, and then ignore (ignore) this warning. Alternatively, you can pipe through `tr ' ' '\n'` to turn words into lines, and then use `while read`. In Bash/Ksh, you can also use a `while read -a` loop to get an array of words per line.
