

Ridirezione dell'input passato a riga di comando

Here Documents << (ridireziona più righe) 1/2

La word che segue il metacarattere << viene utilizzata come delimitatore finale dell'input da passare al comando a sinistra del metacarattere.

Viene ridirezionato nell'input tutto quello che compare tra la word esplicitata dopo il metacarattere << e fino a dove quella word compare ancora all'inizio di una riga.

NB: NON vengono espansi eventuali metacaratteri presenti nella word.

Utilità del ridirezionamento: poter passare input come se provenisse da un file ma senza dover scrivere un file.

Esempio, il seguente script:

```
while read A B C ; do echo $B ; done <<FINE
uno due tre quattro
alfa beta gamma
gatto cane
FINE
echo ciao
```

produce in output

due
beta
cane
ciao

Ridirezione dell'input passato a riga di comando

Here Documents << (ridirezione più righe) 2/2

Notare che, dentro alle righe passate come input, posso inserire delle variabili che vengono espanse. Non occorre che le variabili siano di ambiente perché vengono processate dalla bash corrente prima di creare la subshell che esegue il gruppo di comandi

Esempio: il seguente script

```
VAR=contenuto  
while read A B C ; do echo $B ; done <<FINE  
uno due tre quattro  
alfa ${VAR} gamma  
gatto cane  
FINE  
echo ciao
```

produce in output:

```
due  
contenuto  
cane  
ciao
```

Ridirezione dell'input passato a riga di comando

Here Strings <<< (ridireziona una sola parola)

Viene ridirezionato nell'input la PRIMA parola che compare subito dopo il metacarattere <<<

Se dopo il <<< c'è una variabile questa viene espansa.

Utilità del ridirezionamento: poter passare input contenuto in una variabile

Esempio: una sola parola

```
read A B C <<< alfa
```

```
echo 1 $A 2 $B 3 $C
```

produce in output

```
1 alfa 2 3
```

Esempio con espansione variabili:

```
VAR=paperino
```

```
read A B C <<< ${VAR}
```

```
echo 1 $A 2 $B 3 $C
```

produce in output

```
1 paperino 2 3
```

Esempio: più parole con quoting

```
read A B C <<< "alfa beta gamma"
```

```
echo 1 $A 2 $B 3 $C
```

produce in output

```
1 alfa 2 beta 3 gamma
```

Esempio più parole senza quoting:

```
read A B C <<< alfa beta gamma
```

```
echo 1 $A 2 $B 3 $C
```

produce in output

```
1 alfa 2 3
```