

## Le funzioni

La **funzione** è una lista di comandi che costituisce un sottoprogramma dello shell script e che svolge un determinato compito. La funzione offre la possibilità di organizzare un insieme di comandi che può essere eseguito più volte.

La funzione è identificata da un nome, con il quale può essere richiamata dallo script, così come si fa con i comandi interni di Linux.

La struttura generale di una **function** è:

```
function nome_funzione () {  
    comandi  
}
```

I comandi che formano il corpo della funzione sono raggruppati all'interno di una coppia di parentesi graffe.

La specificazione della parola *function* è facoltativa.

Ci possono essere variabili che sono utilizzate solo all'interno della funzione, cioè variabili locali: esse devono essere dichiarate all'interno della *function* e precedute dalla parola chiave **local**.

```
local variabile=valore
```

Possiamo sfruttare le caratteristiche delle funzioni per eseguire uno stesso gruppo di comandi più volte, ma con valori diversi per un parametro. In questo caso occorre introdurre all'interno della *function* i **parametri posizionali** (le variabili che iniziano con il carattere \$), proprio come con gli script. In questo caso la chiamata della *function* deve contenere anche i valori da passare ai parametri posizionali.

Il comando **return** consente la terminazione anticipata della *function*.

Le nuove funzioni sono rese disponibili alla shell attraverso il comando **export**, come accade per le variabili. Se le funzioni devono diventare permanenti, è possibile anche inserire la loro dichiarazione all'interno dei file di inizializzazione che vengono letti al momento della connessione.

L'utente può controllare le funzioni attualmente disponibili attraverso il comando **set**.

## Progetto 1

**Costruire uno script che calcola lo spazio occupato da due categorie di file: i file di testo aventi estensione .txt nel nome e i file immagine aventi estensione .png.**

Lo script riutilizza in parte il codice usato in precedenza, come esempio per illustrare la struttura *for*, per calcolare lo spazio occupato da un insieme di file. La *function* contiene una variabile locale e un parametro posizionale.

```
#!/bin/bash  
#funzione per calcolare lo spazio occupato  
function occupa () {  
    local somma=0  
    for spazio in `ls -l *.$1 | cut -c29-32`  
    do  
        let somma+=spazio  
    done  
    echo "spazio totale = $somma"  
}
```

```
#main
echo 'occupazione file testi'
occupa txt
echo 'occupazione file immagini'
occupa png
```

La posizione dei caratteri da estrarre con il comando *cut* può variare a seconda della directory corrente.

La shell consente la **chiamata ricorsiva** delle *function*, cioè è possibile chiamare l'esecuzione di una *function* dall'interno della funzione stessa.

## Progetto 2

Scrivere lo script per calcolare il fattoriale di 3 usando una funzione ricorsiva.

```
#!/bin/bash
#fattoriale
function fact() {
    if [ $1 = 0 ]
    then
        echo 1
    else
        echo $(( $1 * $( fact $(( $1 - 1 )) ) ))
    fi
}
if [ $# -ne 1 ]
then
    echo 'manca il numero'
else
    fact $1
fi
```

In questo progetto, per una maggiore comprensione, è stata usata la notazione con il carattere *\$* seguito dalle parentesi tonde per rappresentare l'espansione dei calcoli aritmetici e la *command substitution*, come mostrato alla fine del Paragrafo 5.

Si osservi con attenzione che lo script usa due volte la variabile posizionale *\$1*:

- nell'ultima riga dello script la variabile *\$1* indica il numero fornito dall'utente insieme al comando;
- all'interno della *function* la variabile *\$1* rappresenta il parametro passato con la chiamata della funzione.