

Definire funzioni

E' possibile definire una funzione in uno script

```
function function_name {  
command...  
command...  
}
```

Sintassi alternativa

```
function_name () {  
command...  
command...  
}
```

e altre varianti sintattiche ragionevoli ...

Invocare funzioni

- Si può invocare una funzione semplicemente scrivendo il suo nome nello script DOPO che è stata definita

```
#!/bin/bash
```

```
funky ()
```

```
{
```

```
    echo "This is a funky function."
```

```
    echo "Now exiting funky function."
```

```
} # Function declaration must precede call.
```

```
funky
```

Argomenti delle funzioni

- Non si dichiarano gli argomenti formali
- E' possibile invocare la funzione con un numero arbitrario di argomenti attuali che, nel corpo della funzione saranno riferibili come \$1, \$2, ecc: attenzione all'omonimia
- I parametri sono passati per valore
- Sta al programmatore gestire numero e correttezza dei parametri

```
#!/bin/bash
```

```
fun  ()
```

```
{
```

```
  echo $1 # SI STAMPA IL PRIMO PARAMETRO DELLA FUNZIONE
```

```
}
```

```
echo $1  # SI STAMPA IL PRIMO PARAMETRO DA COMMAND LINE
```

```
fun UNASCRIPTA
```

Valore di ritorno

- Ogni funzione restituisce un valore, chiamato exit status.
- L'exit status può essere stabilito esplicitamente tramite l'istruzione return, altrimenti è l'exit status dell'ultimo comando nella funzione (0 se OK, diverso da zero altrimenti, come noto).
- L'exit status può essere riferito come \$?

```
#!/bin/bash
fun  ()
{
  echo $1
  return 1
}
fun UNASCRITTA
echo $?
```

Scope delle variabili

- Eventuali variabili definite nel corpo di una funzione sono considerate globali, la loro inizializzazione avviene dopo la prima invocazione della funzione

```
#!/bin/bash
fun  ()
{
  nome=PIPP0
}
echo $nome # QUI NON HA VALORE
fun
echo $nome # QUI HA VALORE
```

Variabili locali

- E' possibile definire variabili locali facendole precedere dalla parola chiave local
- Al solito, tali variabili hanno vita separata per ogni invocazione della funzione

```
#!/bin/bash
fun  ()
{
    local nome=PIPPO
    echo $nome # QUI HA VALORE
}
echo $nome # QUI NON HA VALORE
fun
echo $nome # NEMMENO QUI HA VALORE
```