Arrays & The for loop

Como hemos podido observar de algunos scripts vistos en clase, Bash entiende varios tipos de variables (enteros, caracteres).También en la clase 4 estudiamos un tipo particular de sintáxis para armar un lazo **for**, *cuya estructura es :*

```
for x in {1..n}; do
grupo de instrucciones
done
```

En este apunte se mostrará un tipo especial de variable, **array**, *junto con otras formas de sintáxis para el* **for** *loop.*

Arrays

Un array es una variable que contiene múltiples valores. Cualquier variable puede ser usada como un array. Un array no tiene restricciones sobre el número de elementos contenidos en el. El primer elemento de un array en indexado con el número 0.
Para declarar un elemento del array indirectamente usamos la sintáxis :

```
$ arr[indice]=valor
```

donde **arr** *es el nombre del array. Un arreglo también puede ser declarado en forma compacta, cuya sintáxis es :*

```
$ arr=(val1 ...  valn)
```

Veamos un ejemplo. Creemos un array cuyos elementos sean día, mes y año.

```
$ fech=(7 11 2011)
```

Para llamar al array de forma total, o por elemento, se hace :

```
$ echo ${fech[*]}
7 11 2011
$ echo ${fech[2]}
2011
```

Para borrar una variable del array, se hace :

```
$ unset fech[1]
```

lo cual en el ejemplo anterior, borrará la variable asignada a mes. Los elementos del array también pueden ser cadenas de texto (string). Por último,
Siempre tener en cuenta que :

- *Son necesarias las llaves*
- *El primer elemento del array es el 0*
- *Si se sustituye el índice entre corchetes por un asterisco, devuelve todos los valores*

The for loop

Este lazo permite realizar operaciones sobre un conjunto de valores, permitiendo ejecutar una lista de comandos para cada valor del conjunto de valores. **For** *tiene varios tipos de sintáxis, abajo se muestran algunos tipos de sintáxis por medio de ejemplos.*

Ejemplo #1
Este ejemplo muestra la tabla del dos, se utiliza el comando **printf** *para conseguir un formato de salida limpio.*

```
for x in $(seq 1 10) ; do
let y=$x*2 ;
printf "%d*%d=%d\n" $x 2 $y
done
```

Ejemplo #2
Esta sintáxis de **for** *es similiar a la que usa el lenguaje C.*

```
wk=(L I N U X)
for((i=0; i<5; i++));
do
echo ${wk[$i]}
done
```