

Lenguajes de Programación

Tema: Lenguajes tipo Script

Objetivo: Conocer y aplicar el concepto de arreglo y arreglo asociativo en Perl

Referencia: Schwartz, Randal L. “**Learning Perl**” O’Reilly & Associates
1994

Arreglos

¿Qué es un arreglo?

Es una lista de datos escalares. Los arreglos pueden contener cualquier número de elementos.

Ejemplo:

```
(1,2,3,4,5)
("Paris", "Madrid", "Roma", "Venecia")
("cl3p", "Clavo 3 pulgadas", 10000, 103.4)
($a, 17)
(1..5)
(1.2..5.2)
```

¿ Cómo se escribe una variable arreglo en Perl?

Cuando se hace referencia a todos los datos de la lista , se identifican con el signo @ al inicio del nombre de la variable. Por ejemplo:

```
@numeros, @capitales, @producto
```

Cuando se hace referencia a un elemento del arreglo se utiliza el operador \$nombre[índice] , por ejemplo:

```
print "La primera capital es $capitales[0]";
```

¿Cómo se crea una variable arreglo?

- Asignándole una lista de datos escalares

```
@capitales=("Madrid", "París", "Roma", "Viena");
```

- Asignando valores a una localidad del arreglo (operador []):

```
$hobbies[0] = "correr";  
$hobbies[1] = "leer";  
$hobbies[2] = "cine";
```

- Añadiendo una lista a otra

```
@listaUno=(0,2,4);  
@listaDos=(@listaUno,1,3,5); # @listaDos = (0,2,4,1,3,5)
```

Otros operadores:

- Operador **push**(@nombreArreglo, \$nuevoValor): inserta un elemento al final de la lista.

```
@listaUno=(0,2,4);  
push (@listaUno,6,8); # @listaUno=(0,2,4,6,8)
```

- Operador **pop**(@nombreArreglo): regresa y elimina el último valor de la lista.

```
@listaUno=(0,2,4,6);  
$x = pop(@listaUno); # $x=6 y @listaUno=(0,2,4)
```

- Operador **shift**(@nombreArreglo, \$nuevoValor): inserta un elemento al inicio de la lista.

```
@listaUno=(0,2,4);  
shift(@listaUno,6,8); # @listaUno=(6,8,0,2,4)
```

- Operador **unshift**(@nombreArreglo): regresa y elimina el primer valor de la lista.

```
@listaUno=(0,2,4,6);  
$x = unshift(@listaUno); # $x=0 y  
# @listaUno=(2,4,6)
```

- Si una variable arreglo es asignada a una variable escalar, la variable escalar toma el valor de la longitud del arreglo.

```
@listaUno=(0,2,4,6);  
$x = @listaUno;          # $x toma el valor 4
```

Se puede forzar el valor de la longitud dentro de una expresión de la siguiente forma...

```
print  ``.@listaUno    # despliega el valor 4.
```

- **reverse (@variableArreglo)** invierte el orden del arreglo.

```
@listaUno=(0,2,4);  
@listaDos = reverse(@listaUno);  
           # @listaDos = (4,2,0)
```

- **sort (@variableArreglo)** ordena alfabéticamente el arreglo.

```
@listaUno=("yo","el","tu");  
@listaDos = sort(@listaUno);  
           # @listaDos=("el","tu","yo")
```

```
@listaUno=(1,23,12,34,3,2);  
@listaDos = sort(@listaUno);  
           # @listaDos=(1,12,2,23,3,34)
```

- **chop(@variableArreglo)** elimina el último carácter de cada elemento del arreglo.

```
@listaUno=("yo\n","el\n","tu");  
@listaDos = chop(@listaUno);  
           # @listaDos=("yo","el","t")
```

- **<STDIN>** aplicado sobre una variable arreglo, lee cada línea de entrada como un elemento del arreglo
- **foreach \$i (@nombreArreglo):** Para recorrer cada uno de los elementos contenidos en el arreglo.

```
foreach $i ("Hugo","Paco","Luis") {  
    print "$i\n";  
}
```

Arreglos Asociativos (hash)

¿Qué es una asociación?

Es una asociación entre dos valores, al primero de ellos se le llama **llave** (**key**) y al segundo su **valor**.

Ejemplo: (**"250007"**, **"James Bond"**)
 (**"España"**, **"Madrid"**)

¿Qué es un arreglo asociativo?

Es una colección de asociaciones

¿Cómo se representa una variable arreglo asociativo en Perl?

Cuando se hace referencia a todo el conjunto, se identifican con el signo **%** al inicio del nombre de la variable. Por ejemplo:

%capitales, %hobbies, %alumnos

Cuando se hace referencia a un elemento del arreglo se utiliza el operador **\$nombre{ llave }**, por ejemplo:

```
print "La capital de España es $capitales{"España"}"
```

¿Cómo se crea una variable arreglo asociativo?

- Asignando valores a una llave del arreglo (operador { }):

```
$hobbies{ "felipito" } = "comics";  
$hobbies{ "manolito" } = "dinero";  
$hobbies{ "mafalda" } = "TV";
```

- Asignándole un arreglo con los pares de elementos:

```
@lista=("felipito", "comics", "manolito",  
          "dinero","mafalda", "TV");  
%hobbies = @lista;
```

Otros operadores:

- Operador **keys(%nombreArreglo)**: regresa la lista de las llaves definidas en el arreglo.

```
@llaves = keys(%hobbies);  
print @llaves;  
# imprime la lista (felipito,manolito,mafalda)
```

- En un contexto escalar, el operador **keys(%nombreArreglo)** regresa El número de elementos del arreglo.

```
$long = keys (%hobbies);  
print $long          #imprime el valor 3
```

- Operador **values(%nombreArreglo)**: regresa la lista de los valores de las llaves asociadas en el arreglo.

```
@valores = values(%hobbies);  
print @valores;  
# imprime la lista ("comics","dinero","TV")
```

- Operador **each(%nombreArreglo)**: Para recorrer cada uno de los elementos contenidos en el arreglo se utiliza la siguiente estructura de código:

```
while (($llave, $valor) = each( %nombreArreglo )) {  
    ...  
    # código donde se utilizan las variables  
    # $llave y $valor  
    ...  
}
```

Por ejemplo:

```
while (($llave, $valor) = each( %hobbies)) {  
    print "El hobbie de $llave es $valor";  
}
```

Operador **delete \$nombreArreglo { llave }**: elimina una asociación del arreglo, por ejemplo:

```
delete $hobbies{ manolito };  
    # elimina la relación ("manolito","dinero")
```