Capítulo 2

Perl, Estilos y Ejemplos

Estilos

- Cada programador tiene su propio estilo
- Existen lineamientos generales
  - Programas más fáciles de leer, entender y mantener
- Más importantes
  - Siempre ejecutar los programas con -w (use warnings)
  - Siempre usar “use strict”
    - Si no se usa, indicar las razones en el código
- Más importante estéticamente
  - Siempre ubicar “}” alineado con palabra que inició el bloque

Estilos

- Otras recomendaciones
  - Indentación de 4 columnas
  - Agregar “{“ en la misma línea que comienza el bloque
  - Agregar un espacio antes de “{“ en un bloque de varias líneas
  - Bloque de una línea debería colocarse en una línea
    - Incluyendo las llaves
  - No dejar espacios antes de un ;
  - Omitir ; en bloques de una línea
  - Usar espacios rodeando la mayoría de los operadores
  - Usar espacios rodeando expresiones complejas
    - Para las que estén entre paréntesis

Estilos

- Otras recomendaciones:
  - Lineas en blanco entre bloques de código que hacen distintas cosas
  - No usar espacios entre nombre de funciones y sus paréntesis
  - Un espacio después de cada coma (,)
  - Considerar que si Perl permite hacer algo, no significa que deberían hacerlo

```
open(FOO,$foo) || die "Can't open $foo: $!";

# Cual es mejor?

die "Can't open $foo: $!" unless open(FOO,$foo);
```

Estilos

- Recomendaciones
  - Preocuparse de que la acción principal no quede escondida

```
print "Starting analysis\n" if $verbose;
```

```
$verbose && print "Starting analysis\n";
```

  - Es recomendable usar los paréntesis

```
return print reverse sort num values %array;
```

```
return print(reverse(sort num (values(%array))));
```

Estilos

- Recomendaciones
  - Use el operador “last” para salir de un ciclo bajo cierta condición

```
LINE:
    for (;;) {
        statements;
        last LINE if $foo;
        next LINE if /^#/;
        statements;
    }
```

  - Por portabilidad, se recomienda usar “eval” para las funcionalidades que no están en todos los sistemas
    - Se puede averiguar la versión de Perl consultando $]

Estilos

- Recomendaciones
  - Use identificadores nemotécnicos
  - Use "_" como separador de identificadores largos
    - $var_names_like_this es mejor que $VarNamesLikeThis
  - Uso correcto de mayúsculas y minúsculas para indicar el alcance de las variables

```
$ALL_CAPS_HERE                    constants only (beware
                                  clashes with perl vars!)

$Some_Caps_Here package-wide      global/static

$no_caps_here                     function scope my() or
                                  local() variables
```

Estilos

- Recomendaciones
  - Usar “and” y “or” en vez de “&&” y “||”
  - Usar “here documents” en lugar de muchos “print()”;
  - Alineación vertical

```
$IDX = $ST_MTIME;
$IDX = $ST_ATIME          if $opt_u;
$IDX = $ST_CTIME          if $opt_c;
$IDX = $ST_SIZE           if $opt_s;

mkdir $tmpdir, 0700 or die "can't mkdir $tmpdir: $!";
chdir($tmpdir)      or die "can't chdir $tmpdir: $!";
mkdir 'tmp',   0777 or die "can't mkdir $tmpdir/tmp: $!";
```

Estilos

- Recomendaciones
  - Siempre revisar códigos de retorno de llamadas de sistema

```
opendir(D, $dir)     or die "can't opendir $dir: $!";
```

    - Los buenos mensajes de error deberían irse a STDERR, incluyendo la causa del problema