# INTRODUCCIÓN A PERL

Por Carlos Ciuraneta

carlos@linuxupc.upc.es

# CONTENIDO

- Introducción:"¿Qué es Perl?"
- Tipos y variables
- Sintaxis básica
- Funciones
- Ficheros
- Programas externos
- Expresiones regulares
- Referencias, punteros
- Librerías
- Conceptos avanzados

# CONTENIDO

- **Introducción:"¿Qué es Perl?"**
- Tipos y variables
- Sintaxis básica
- Funciones
- Ficheros
- Programas externos
- Expresiones regulares
- Referencias, punteros
- Librerías
- Conceptos avanzados

# Introducción:"¿Qué es Perl?"

- Lenguaje de scripting
  - Interpretado
  - "Sencillo"
  - Programación rápida para tareas sencillas
- Tratamiento de ficheros
- Programación CGI's

# CONTENIDO

- Introducción:"¿Qué es Perl?"
- **Tipos y variables**
- Sintaxis básica
- Funciones
- Ficheros
- Programas externos
- Expresiones regulares
- Referencias, punteros
- Librerías
- Conceptos avanzados

# Tipos y variables

- Tipos básicos
  - Escalares
  - Arrays
  - Listas
  - Hashes
- Tipos complejos
  - Composición de tipos básicos
  - Array de arrays
  - Array de hashes
  - Hashes de arrays
  - etc.

# Escalares $

- Tipo básico
  - Número (entero, floating)
  - String
  - Referencia
- Se antecede siempre con un $

```
$dummy=5;
$dummy='hola';
$dummy=\$referenciada;
```

# ARRAYS @

- Conjunto con orden de tipos básicos
- Se antecede siempre con una @
  - @vector=(1,2,3,4,5);
  - @vector=('primero','segundo','tercero');
  - @vector=(1,'dos',3,'cuatro','cinco');
- Un elemento es un escalar
  - $vector[0]=3.1416;
- $#vector, scalar @vector

# HASHES %

- Conjunto no ordenado o array asociativo
- Se antecede con %
  - %hash=(‘uno’,1,’dos’,2);
  - %hash=(‘uno’=>1,’dos’=>2,);
- Un elemento es un escalar
  - $hash{‘uno’}=1.3;

# CONTENIDO

- Introducción:"¿Qué es Perl?"
- Tipos y variables
- **Sintaxis básica**
- Funciones
- Ficheros
- Programas externos
- Expresiones regulares
- Referencias, punteros
- Librerías
- Conceptos avanzados

# Sintaxis básica

- Sentencias
  - Asignaciones (=)
  - Llamadas a funciones
  - do {} , eval{}
- Estructuras de control
  - Condicionales (if,unless)
  - Bucles (while,for,foreach)
  - Saltos (goto)

# Condicionales

```
if (!open (FOO,$foo)) {
   die "can't open $foo: $!";
}
else
{
   print "it works well \n";
}

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

# Bucles

```
while (expr){
   $line=<STDIN>;
}

unless(expr) …

for($i=0;$i<30;$i++) {
   print "posicion: $i\n";
}

foreach $i (@elements){
   print "elemento $i\n";
}
```

# Saltos

```
LABEL: $lexpr=rexpr;
       …

goto LABEL;

SWITCH: {
   $abc=1, last SWITCH if /^abc/;
   $def=1, last SWITCH if /^def/;
   $nothing=1;
}
```

# CONTENIDO

- Introducción:"¿Qué es Perl?"
- Tipos y variables
- Sintaxis básica
- **Funciones**
- Ficheros
- Programas externos
- Expresiones regulares
- Referencias, punteros
- Librerías
- Conceptos avanzados

# Funciones

- Declaración
  - sub nombre () {}
- Llamada
  - &dummy(params);
  - dummy(params);
  - &dummy;  … se pasa el valor de @_

# Funciones- Declaración

```
sub funcioncilla;

sub funcioncilla {
   my $param=shift;

   print "Parametro: $param\n";
   return;
}
```

# CONTENIDO

- Introducción:"¿Qué es Perl?"
- Tipos y variables
- Sintaxis básica
- Funciones
- **Ficheros**
- Programas externos
- Expresiones regulares
- Referencias, punteros
- Librerías
- Conceptos avanzados

# Ficheros

- Entrada/salida básica
  - `open FD,"ficherillo.txt"`
  - `<>`
  - `print`
- Uso de ficheros

```
open FD,'mi_fichero.txt';
while (<FD>)
{
   print;
}
close FD;
```

# CONTENIDO

- Introducción:"¿Qué es Perl?"
- Tipos y variables
- Sintaxis básica
- Funciones
- Ficheros
- **Programas externos**
- Expresiones regulares
- Referencias, punteros
- Librerías
- Conceptos avanzados

# Programas externos

- Ejecución
  - `$dummy=`ls -la`;`
  - `$dummy=system('ls -la');`
- Recogida de resultados
  - `$dummy=`ls -la`;`
  - `@dummy=`ls -la`;`
  - `open FD,'ls -la |';`

# CONTENIDO

- Introducción:"¿Qué es Perl?"
- Tipos y variables
- Sintaxis básica
- Funciones
- Ficheros
- Programas externos
- **Expresiones regulares**
- Referencias, punteros
- Librerías
- Conceptos avanzados

# Expresiones regulares

- `m/^.*[a]+\s+/`
- `$dummy=~m/expresion.*/`
- `($uno,$dos)=$dummy=~m/(expresion1),(expresion2)/`
- `$dummy=~m/(expresion1),(expresion2)/;`
  `$uno=$1;`
  `$dos=$2;`

# CONTENIDO

- Introducción:"¿Qué es Perl?"
- Tipos y variables
- Sintaxis básica
- Funciones
- Ficheros
- Programas externos
- Expresiones regulares
- **Referencias, punteros**
- Librerías
- Conceptos avanzados

# Referencias, punteros

- Hard references \, ->
  - `$r_uno=\$uno;`
    `print $$r_uno;`
  - `$r_vector=\@vector;`
  - `print $r_vector->[0]`
- Soft references
  - `$dummy=5;`
    `$aux='dummy';`
    `print $$aux;`

# Creación de estructuras

- Vectores:
  - `$vector=[4,5,6,7];`
    `$vector->[4]=8;`
- Hashes:
  - `$hash={};`
    `$hash->{'clave'}='valor';`

# CONTENIDO

- Introducción:"¿Qué es Perl?"
- Tipos y variables
- Sintaxis básica
- Funciones
- Ficheros
- Programas externos
- Expresiones regulares
- Referencias, punteros
- **Librerías**
- Conceptos avanzados

# Librerías

- use
- require

# CONTENIDO

- Introducción:"¿Qué es Perl?"
- Tipos y variables
- Sintaxis básica
- Funciones
- Ficheros
- Programas externos
- Expresiones regulares
- Referencias, punteros
- Librerías
- **Conceptos avanzados**

# Conceptos avanzados

- Creación de librerías
- CPAN http://www.cpan.org
- Programación OO
- Threads