

GUIA BASICA DE PERL

1. ¿Qué es Perl?

Perl es un lenguaje de programación interpretado, conocido por su flexibilidad y potencia en la manipulación de texto. Es ampliamente utilizado en desarrollo web, administración de sistemas, análisis de datos, y más.

2. Instalación de Perl

- Linux/Mac: Perl suele venir preinstalado. Puedes verificarlo con `perl -v`.
- Windows: Descarga e instala Strawberry Perl desde strawberryperl.com.

3. Primer script en Perl

Crea un archivo llamado `primer_script.pl` y escribe lo siguiente:

```
#!/usr/bin/perl  
print "¡Hola, mundo!\n";
```

Para ejecutar el script:

```
perl primer_script.pl
```

4. Estructura Básica de Perl

Comentarios: Se usa `#` para comentarios de una sola línea.

```
#Comentario en Perl
```

Declaración de variables:

Escalares: Para valores simples (números o cadenas).

```
my $nombre = "Juan";  
my $edad = 25;
```

Arrays: Listas ordenadas.

```
my @frutas = ("manzana", "banana",  
"cereza");
```

Hashes: Pares clave-valor.

```
my %paises = ("SV" => "El Salvador", "MX"  
=> "México");
```

5. Impresión de datos

```
print "Mi nombre es $nombre y tengo $edad años.\n";  
print "Primera fruta: $frutas[0]\n"; # Acceso a arrays  
print "País: $paises{'SV'}\n";      # Acceso a hashes
```

6. Condicionales y Bucles

Condicionales:

```
if ($edad > 18) {  
    print "Eres mayor de edad.\n";  
} else {  
    print "Eres menor de edad.\n";  
}
```

Bucles:

```
# Bucle for
for (my $i = 0; $i < 5; $i++) {
    print "Iteración $i\n";
}

# Bucle while
my $contador = 0;
while ($contador < 5) {
    print "Contador: $contador\n";
    $contador++;
}

# Bucle foreach
foreach my $fruta (@frutas) {
    print "Fruta: $fruta\n";
}
```

7. Subrutinas

Las subrutinas permiten definir bloques de código reutilizables.

```
sub saludar {  
    my ($nombre) = @_;  
    print "¡Hola, $nombre!\n";  
}  
saludar("Carlos");
```

8. Entrada y salida de datos

Para capturar datos del usuario:

```
print "¿Cómo te llamas? ";  
my $nombre = <STDIN>;  
chomp($nombre); # Remueve el salto de línea  
print "¡Hola, $nombre!\n";
```

9. Manejo de archivos

Lectura de archivos:

```
open(my $fh, '<', 'archivo.txt') or die "No se pudo abrir el
archivo: $!";
while (my $linea = <$fh>) {
    print $linea;
}
close($fh);
```

Escritura en archivos:

```
open(my $fh, '>', 'salida.txt') or die "No se pudo crear el
archivo: $!";
print $fh "Esta es una línea de texto.\n";
close($fh);
```

10. Módulos y bibliotecas

Perl tiene muchos módulos disponibles. Puedes instalarlos con `cpan` o `cpanm`.

```
cpan install JSON
```


Y usarlos en tus scripts:

```
use JSON;  
my $data = '{"nombre": "Ana", "edad": 30}';  
my $parsed_data = decode_json($data);  
print "Nombre: $parsed_data->{'nombre'}\n";
```

Y usarlos en tus scripts:

```
use JSON;  
my $data = '{"nombre": "Ana", "edad": 30}';  
my $parsed_data = decode_json($data);  
print "Nombre: $parsed_data->{'nombre'}\n";
```

11. Consejos y buenas prácticas

- Usa `use strict;` y `use warnings;` al inicio de tus scripts para ayudar a detectar errores.

```
use strict;  
use warnings;
```

- Siempre usa `my` para declarar variables locales.

12. Recursos adicionales

- Documentación oficial de Perl
- [Perl Monks](#): Una comunidad activa de programadores de Perl.