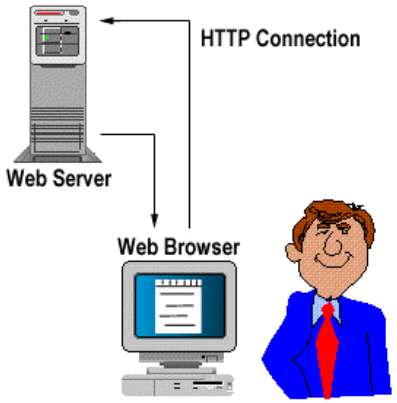
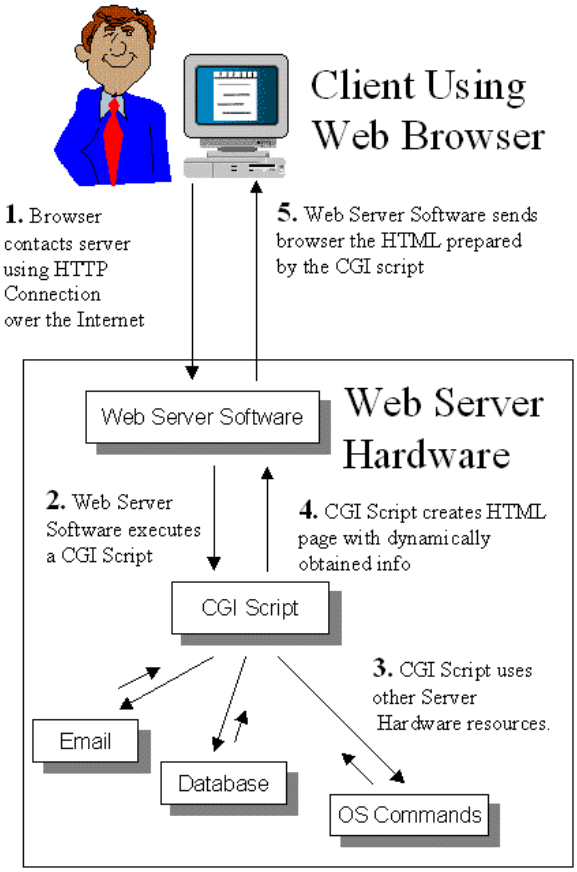
Lenguajes de programación
Tema: PERL

Cliente – Servidor

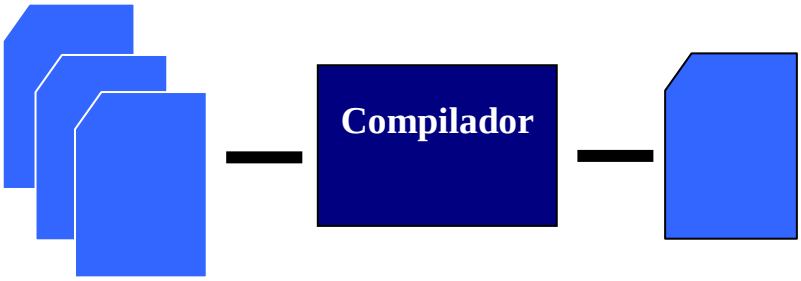


Cliente – Servidor



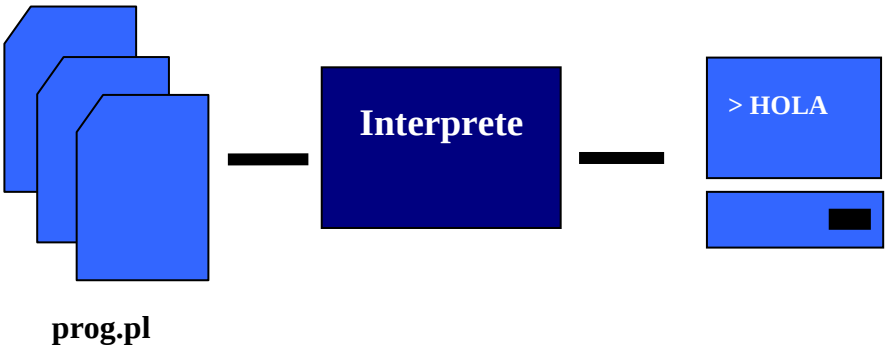
¿Qué es un compilador?

Es un programa que recibe un programa en un lenguaje de programación y genera el programa en el lenguaje de la máquina



¿Qué es un interprete?

Es un programa que recibe un programa en un lenguaje de programación y ejecuta las instrucciones del programa.



Lenguajes Script

+ Lenguajes con sintaxis sencilla (fácil de aprender)

+ Orientados a una aplicación en particular

+ Ejemplos: Javascript, VScript, Perl, Pyton, Visual Basic, Lingo, AutoLisp, etc...

Historia

+ "Practical Extraction and Report Language"

+ Creador, arquitecto en jefe, implementador y administrador: Larry Wall

+ Direcciones: http://www.perl.org y http://www.perl.com

Perl es un lenguaje sencillo

+ Es un diseño sencillo

+ Es interpretado

+ Es altamente portable (Unix en todos sus sabores, NT, windows)

Es un lenguaje especializado para el manejo de String

+ Tiene implementado una gran cantidad de operaciones eficientes para el manejo de cadenas de caracteres

+ Aplicaciones

+ Programación de Sistemas Opreativos

+ Aplicaciones de CGI

+ Lenguaje script de otras aplicaciones (HTML, servidores de WEB)

¿Cómo es un programa de Perl?

```
print "Hola Todo Mundo<H2><BR>"
```

<u>Datos Escalares en Perl</u>

- **Un dato escalar es el tipo más simple de dato que un lenguaje puede manipular.**

- **En Perl los datos escalares son números o strings.**

- **Los valores escalares pueden...**

  + Aplicarse a operadores (como la suma o la concatenación) para obtener otro valor escalar

  + Pueden almacenarse en variables escalares

  + Pueden ser leídos y almacenados en archivos y dispositivos

- **<u>Literales:</u> es la manera como un valor es representado en el texto de un programa:**

  Literales enteras: **`162, -1562, 0, 12345`**

  Literales reales (float): **`1.25, -16.45E-25`**

  Literales octales: **`0377, -02`**

  Literales hexadecimales: **`-0xffff, 0x1a`**

- **Literales string:**

  string con comilla sencilla
  string con comilla doble

- **Literales string con comilla sencilla:**

  string con comilla sencilla: **`'hola que tal \n'`**
  string con comilla doble: **`"hola que tal \n"`**

Operadores de números

- **+,-,*,/,%,++,--,****

- **Operadores de strings**
  - Concatenación: **"Casa"."blanca" => "Casablanca"**
  - Repetición: **"tec" x 3 => "tectectec"**

- **Comparación**

Números	**String**
==	eq
!=	ne
<	lt
>	gt
<=	le
>=	ge

- **Conversión entre strings y números**

  - Si un string es utilizado en un operador numérico, Perl automáticamente lo convierte a número:

    "3.2" * 3 => 9.6

  - Si un número es utilizado en un operador string, Perl Automáticamente lo convierte en string

    "al" . 567890 => "al567890"

- **Variables Escalares**

  - Las variables escalares sólo pueden guardar valores escalares: string y números

  - El nombre de una variable escalar inicia con el signo de pesos:

    $x, $nombre, $dirección-ip

  - Perl hace diferencia entre las letras mayúsculas y las minúsculas

$x es diferente a **$X**

- Operadores de asignación

=, +=, *=, /=, **=, %=, .=

- **Función chop( )**

La función chop elimina el último carácter de un string

```
chop("plumas")     =>   "pluma"
chop ("hola\n")    =>    "hola"
```

- **Diferencia entre la comilla sencilla y la doble**

Cuando un string tiene doble comilla se dice que permite la sustitución de variables, esto significa que el string es revisado buscando posibles variables escalares. Cuando esta se encuentra, se reemplaza por el valor de la variable, por ejemplo...

```
$nombre = "Pavaroti"
$b = "mi cantante favorito es $nombre"
$c = 'mi cantante favorito es $nombre'
print $b      # imprime mi cantante favorito es Pavaroti
print "\n"
print $c      # imprime mi cantante favorito es $nombre
```

- **Para leer del teclado valores escalares...**

Se utiliza <STDIN>, por ejemplo

```
print "cual es tu nombre: "
$n = <STDIN>
chop($n)                    # borra el ultimo carácter (\n)
print "tu nombre es $n"
```