



# Curso de Programación en PHP

## Nivel I

Universidad Autónoma de Entre Ríos  
Facultad de Ciencia y Tecnología - Oro Verde - v2.1.2



### Capítulo 1: Conceptos básicos

**Conceptos básicos**

**Sintaxis del lenguaje**

**Variables y constantes**

**Expresiones y operadores del lenguaje**

**Estructuras de control**



### Clase 1: Sintaxis, variables, expresiones y estructuras

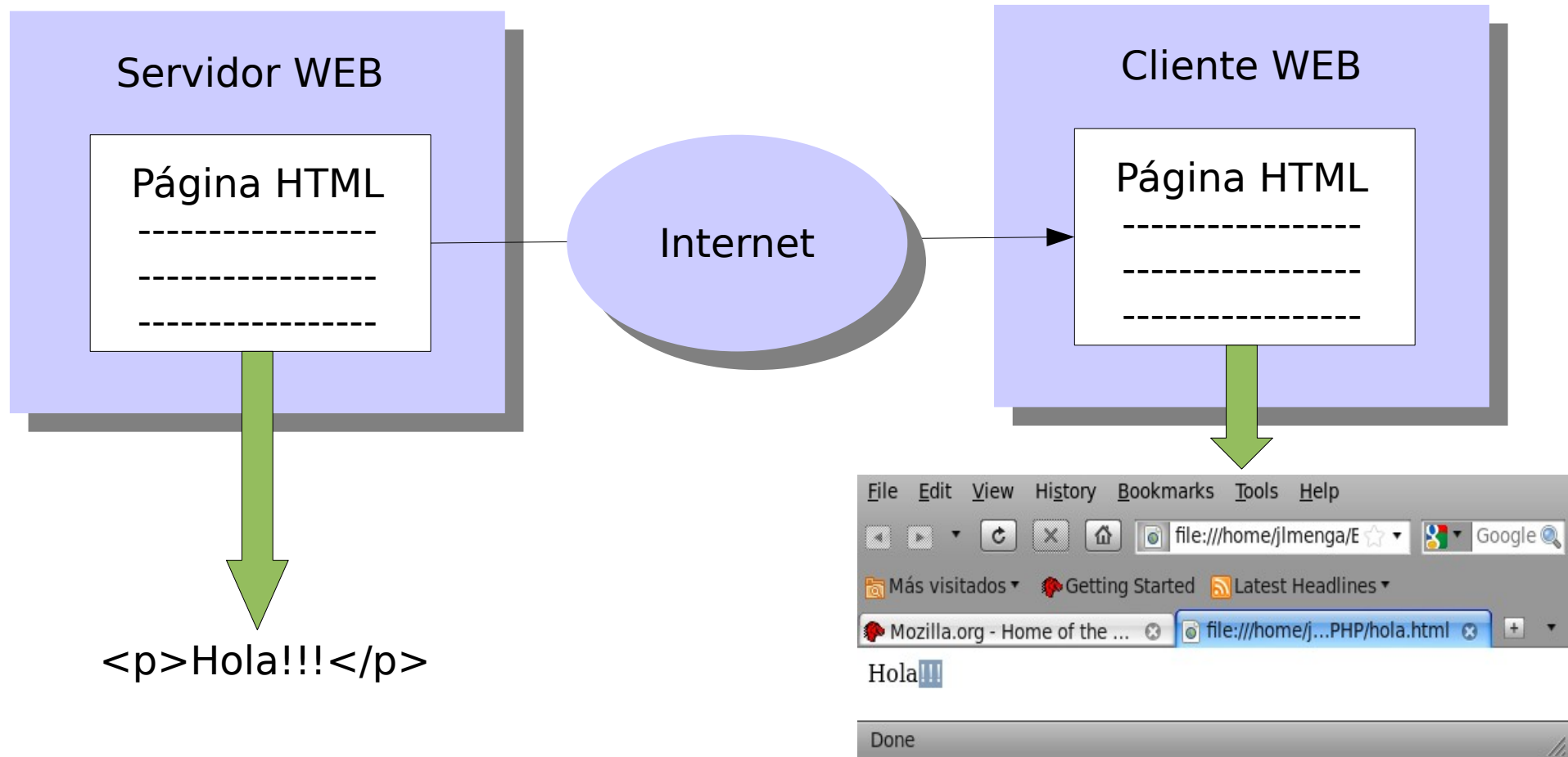
## Lenguajes de Script

- PHP es un lenguaje de script que se ejecuta del lado del servidor.
- Existen otros lenguajes similares a PHP como ASP, Python, Ruby, .net.
- El código PHP se encuentra incrustado en los archivos HTML que se encuentran en el servidor, el cual interpreta y ejecuta el código antes enviar resultados al cliente.
- El cliente no visualiza el código PHP sino los resultados del antes mencionado en formato HTML.



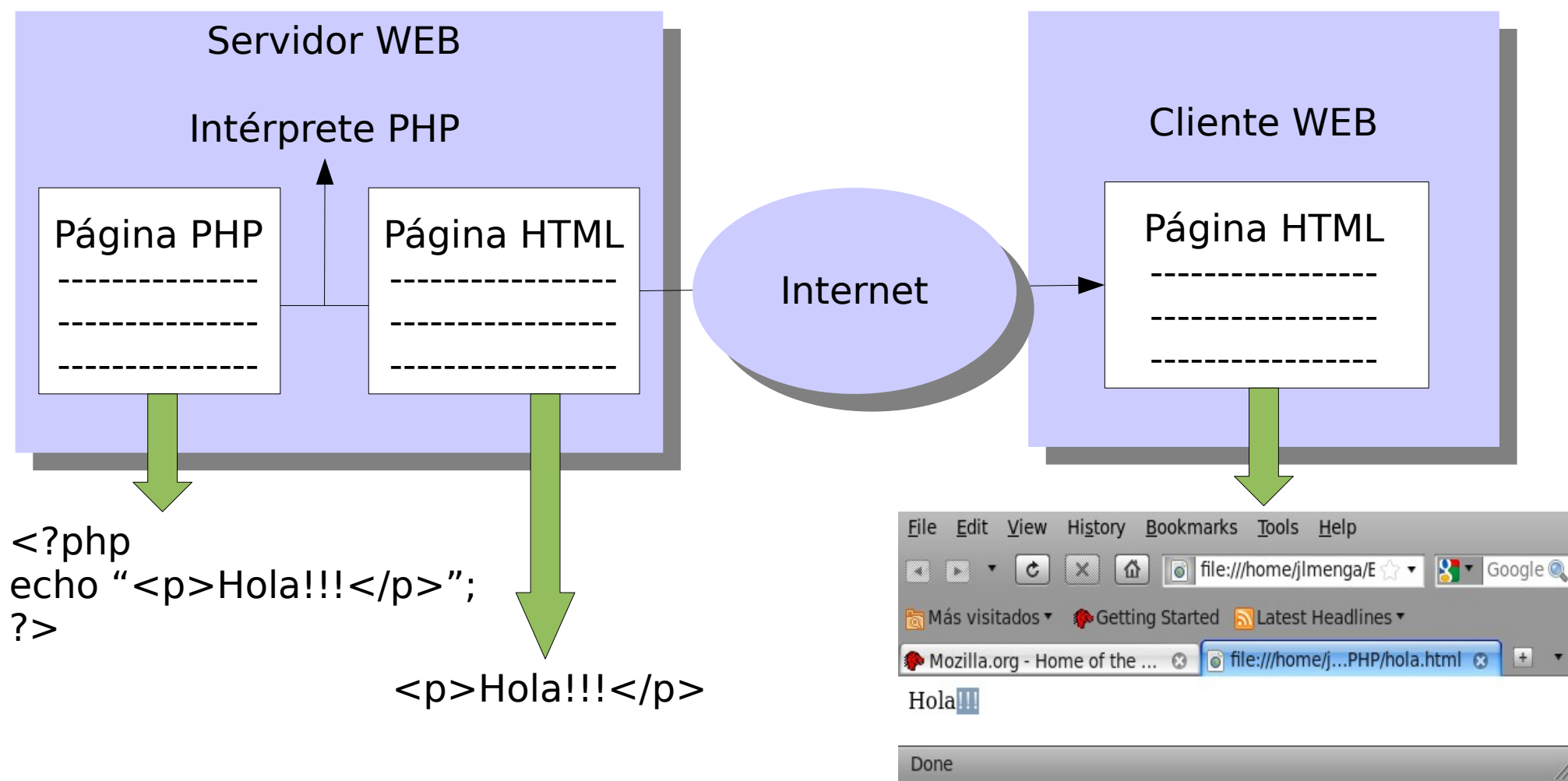
### Clase 1: Sintaxis, variables, expresiones y estructuras

## Funcionamiento de PHP



### Clase 1: Sintaxis, variables, expresiones y estructuras

## Funcionamiento de PHP (continuación)





### Clase 1: Sintaxis, variables, expresiones y estructuras

## Historia de PHP

- PHP nació en 1994 de la mano de Rasmus Lerdorf.
- PHP significa PHP HyperText PreProcessor es un acrónimo recursivo.
- Actualmente desarrollada y mantenida por The PHP Group bajo la licencia PHP License 3.01.
- La versión 1 de PHP nace en 1995 y actualmente tenemos la versión 8 del lenguaje.



### Clase 1: Sintaxis, variables, expresiones y estructuras

## Características de PHP

- Es multiplataforma.
- Orientado a desarrollo Web y Scripting.
- Amplia conectividad con base de datos.
- Muy extensivo por medio de módulos.
- Posee licencia libre.
- Abundante documentación y en varios idiomas.



### Clase 1: Sintaxis, variables, expresiones y estructuras

## Aplicaciones desarrolladas con PHP

- Facebook. (Sitio web de redes sociales)
- Moodle. (Plataforma educativa)
- PhpMyAdmin. (Herramienta de administración de base de datos MySQL)
- Drupal. (Sistema de gestión de contenidos)
- Joomla. (Sistema de gestión de contenidos)
- PhpPgAdmin. (Herramienta de administración de base de datos PostgreSQL)





### Clase 1: Sintaxis, variables, expresiones y estructuras

## Requisitos para desarrollar con PHP

#### Del lado del Servidor:

- Servidor WEB Apache. (<httpd.apache.org>)
- PHP 5 o superior. ([www.php.net](http://www.php.net))
- Soporte del lenguaje PHP en el servidor Web.
- Soporte de un base de datos en PHP5.

#### Del lado del cliente:

- Entorno de desarrollo o editor de textos (eclipse, notepad plus).
- Herramienta de administración de base de datos (MySQL GUI-Tools, MySQL Workbench, phpMyAdmin).
- Navegador WEB (Firefox + plugins).



### Clase 1: Sintaxis, variables, expresiones y estructuras

## Embebiendo código PHP dentro de HTML

- **Método 1**  

```
<?php  
    echo "Método recomendado para incrustar código";  
?>
```
- **Método 2**  

```
<? echo "Método reducido para incrustar código" ?>
```
- **Método 3**  

```
<script language="php">  
    echo ("muchos editores (como FrontPage) no  
    aceptan instrucciones de procesado");  
</script>
```
- **Método 4**  

```
<% echo ("Opcionalmente, puedes usar las etiquetas ASP"); %>
```



### Clase 1: Sintaxis, variables, expresiones y estructuras

## Separación de instrucciones

- Método 1  
    <?php  
        echo "Punto y coma separa una instrucción de otra";  
    ?>
- Método 2  
    <?php  
        echo "El cierre de un bloque de código implica un punto y coma"  
    ?>



### Clase 1: Sintaxis, variables, expresiones y estructuras

## Comentarios

- **Método 1**  

```
<?php  
    echo "Prueba de comentarios"; // Comentario de una linea al estilo C  
?>
```
- **Método 2**  

```
<?php  
    /* Comentario de múltiples lineas, primera linea  
       segunda linea del comentario. */  
    echo "Prueba de comentarios"  
?>
```
- **Método 3**  

```
<?php  
    echo "Comentario final"; # Comentario de una linea al estilo Shell  
?>
```



### Clase 1: Sintaxis, variables, expresiones y estructuras

## Tipo de datos

Posee 8 tipos de datos primitivos

- Tipo de datos escalares  
boolean  
Integer  
float  
string
- Tipo de datos compuestos  
array  
object
- Tipo de datos especiales  
resource  
NULL



### Clase 1: Sintaxis, variables, expresiones y estructuras

## Tipo de datos (continuación)

Funciones relativas a los tipos de **datos primitivos**

- `gettype()`: devuelve el tipo de una variable.
- `is_type()`: nos comprueba el **tipo de una variable** dada, pudiendo utilizar las siguientes funciones.
  - `is_array()`
  - `is_boolean()`
  - `is_integer()`
  - `is_float()`
  - `is_object()`
  - `is_string()`
  - `is_resource()`
- `var_dump()`: muestra el **tipo y el valor** de una variable.



### Clase 1: Sintaxis, variables, expresiones y estructuras

## Tipo de datos (continuación)

Ejemplos de los tipos de datos primitivos

- Tipo **integer** (números enteros)

```
$a = 1234; // número decimal
```

```
$a = -123; // un número negativo
```

```
$a = 0123; // número octal (equivalente al 83 decimal)
```

```
$a = 0x1A; // número hexadecimal (equivalente al 26 decimal)
```

- Tipo **float** (números reales)

```
$a = 1.234;
```

```
$b = 1.2e3;
```

```
$c = 7E-10;
```

- Tipo **boolean** (lógico)

```
$foo = true; // asigna el valor TRUE a $foo
```



### Clase 1: Sintaxis, variables, expresiones y estructuras

## Tipo de datos (continuación)

### Consideraciones del tipo string

- **'simples'**: admite los caracteres de escape `\` (comilla simple) y `\\` (barra). Las variables NO se expanden
- **"dobles"**: admite más caracteres de escape, como `\n`, `\r`, `\t`, `\\`, `\$`, `\"`. Los nombres de variables SÍ se expanden
- Para acceso a un carácter de una cadena se debe realizar de la siguiente manera: `$inicial = $nombre[0];`

### Ejemplos de los tipos de datos primitivos

- Tipo string (cadenas de texto)

```
$a = 9;
print 'a vale $a\n';           // muestra a vale $a\n
print "a vale $a\n";          // muestra a vale 9 y avanza una línea
print "<IMG SRC='logo.gif'>";    // muestra <IMG SRC='logo.gif'>
print "<IMG SRC=\"logo.gif\">";  // muestra <IMG SRC=\"logo.gif\">
```





### Clase 1: Sintaxis, variables, expresiones y estructuras

## Variables en PHP

- El nombre de una variable es **sensible a mayúsculas y minúsculas**.
- Comienzan siempre con un **\$**.
- El nombre de la variable puede estar **formado por letras y números**, además de algunos caracteres especiales como **\_**.
- Poseen un **Ámbito** sobre el que tienen inferencia. Dicho ámbito puede ser:
  - **Global**: tiene inferencia dentro de todo el script.
  - **Local**: tiene inferencia dentro de una función.
- Variables **predefinidas**: permiten recuperar datos externos a una aplicación o de entorno. Ejemplos:
  - **\$GLOBALS** - **\$\_SERVER** - **\$\_GET** - **\$\_POST**
  - **\$\_COOKIES** - **\$\_FILES** - **\$\_ENV** - **\$\_REQUEST** - **\$\_SESSION**



### Clase 1: Sintaxis, variables, expresiones y estructuras

## Variables en PHP (continuación)

Ejemplos:

```
$numero = 12;  
print ("El valor de numero es: ".$numero);
```

El valor de numero es: 12

```
$palabra = "No voy a usar Hola Mundo!!!!.";   
print ("El valor de palabra es: ".$palabra."<br>");  
print ("Pero podemos tratarla como array. <br>");  
print ("En donde el primer valor seria: ".$palabra[0]);
```

El valor de palabra es: No voy a usar Hola Mundo!!!!.  
Pero podemos tratarla como array.  
En donde el primer valor seria: N



### Clase 1: Sintaxis, variables, expresiones y estructuras

## Variables variables en PHP

- Se pueden crear los nombres de las variables dinámicamente.
- El proceso se lleva a cabo en base al valor de una variable previamente declarada.

Ejemplo:

```
$saludo_es = "Hola!!!";  
$saludo_en = "Hello!!!";  
$idioma = "es";  
$saludar = "saludo_" . $idioma;  
print $$saludar;
```

Hola!!!

```
$saludo_es = "Hola!!!";  
$saludo_en = "Hello!!!";  
$idioma = "en";  
$saludar = "saludo_" . $idioma;  
print $$saludar;
```

Hello!!!



### Clase 1: Sintaxis, variables, expresiones y estructuras

## Constantes en PHP

- Se pueden crear haciendo uso de la palabra reservada `define`.
- No comienzan con un signo `$`.
- La sintaxis general para la creación sería:

```
bool define (nombre, valor [, bool $case_insensitive =false])
```

En donde:

`nombre`: es el nombre de la constante a definir.

`valor`: el valor que se le va a asignar a dicha constante.

`[, bool $case_insensitive =false]`: indica si se define sensible a mayúsculas y minúsculas.

Ejemplo:

```
define ("NOMBRE","Esteban","true");  
print NOMBRE;  
Esteban
```



### Clase 1: Sintaxis, variables, expresiones y estructuras

## Expresiones y Operadores en PHP

- Operadores **Aritméticos**  
+, -, /, %, \*, ++, --
- Operadores de **Comparación**  
==, !=, <, >, <=, >=
- Operadores **Lógicos**  
and (&&), or (||), ! (not)
- Operadores de **Cadena**
  - concatenación,
  - .= asigna el valor previo y concatena un nuevo valor.
- Operadores de **Asignación**  
= asigna un valor dado.



### Clase 1: Sintaxis, variables, expresiones y estructuras

## Expresiones y Operadores - Ejemplos

- Operadores **Aritméticos**

```
$a= 5; $b= 4;  
print "Suma: ". $a + $b;  
print "Resta: ". $a - $b;  
print "Producto: ". $a * $b;
```

→ Suma: 9  
→ Resta: 1  
→ Producto: 20

```
$a= 1;  
$a++  
$a--
```

→ \$a= 1  
→ \$a= 2 (incrementa el valor en 1)  
→ \$a= 0 (decrementa el valor en 1)

- Operadores de **Comparación**

- ```
$a= 4; $b= 5;  
$a == $b  
$a != $b  
$a > $b
```

→ false  
→ true  
→ false



### Clase 1: Sintaxis, variables, expresiones y estructuras

## Expresiones y Operadores - Ejemplos

- Operadores Lógicos

```
$a= 4; $b= 5;
```

```
( ($a == 4) && ($b == 5) )
```

→ true

```
( ($a == 4) || ($b == 12) )
```

→ true

```
( ($a == 4) and ($b == 12) )
```

→ false

- Operadores de Cadena

```
$a= "No voy a usar"; $b= " Hola mundo!!!!!!";
```

```
$mensaje= $a.$b;
```

```
print $mensaje;
```

→ No voy a usar Hola mundo!!!!!!

```
$a= "No voy a usar"; $a.= " Hola mundo!!!!!!";
```

```
print $a;
```

→ No voy a usar Hola mundo!!!!!!



### Clase 1: Sintaxis, variables, expresiones y estructuras

## Estructuras de Control

### Estructuras Selectivas

if – else  
switch

### Estructuras Iterativas

for  
foreach  
while





### Clase 1: Sintaxis, variables, expresiones y estructuras

## Estructuras de Control Selectivas

Estructura Selectiva **if - else**

```
if (condicion){  
    opción o sentencia verdadera;  
}else{  
    opción o sentencia falsa;  
}
```

Estructura Selectiva **if - else** anidada

```
if (condicion1){  
    opción o sentencia verdadera;  
}else if(condicion2){  
    opción o sentencia verdadera (condición2);  
}else{  
    opción o sentencia falsa (condición1);  
}
```



### Clase 1: Sintaxis, variables, expresiones y estructuras

## Estructuras de Control - Ejemplos

Estructura Selectiva **if - else**

```
if (4 < 5){  
    print ("4 es menor que 5 !!!");  
}else{  
    print ("Este resultado no puede ser....");  
}
```

Estructura Selectiva **if - else** anidada

```
if (5 < 4){  
    print ("Este resultado no puede ser....");  
}else if(4 < 3){  
    print ("Este resultado no puede ser....");  
}else{  
    print (" 3 es el valor mas chico.")  
}
```



### Clase 1: Sintaxis, variables, expresiones y estructuras

## Estructuras de Control Selectivas

Estructura Selectiva **switch**

```
switch ($variable){  
  
    case condición_1:  
        sentencia_1  
        sentencia_2  
        break;  
  
    case condición_2:  
        sentencia_2  
        break;  
  
    case condición_n:  
        sentencia_n  
        break;  
  
    default :  
        sentencia default  
  
}
```



### Clase 1: Sintaxis, variables, expresiones y estructuras

## Estructuras de Control - Ejemplos

Estructura Selectiva **switch**

```
$a = 3;  
switch ($a){  
  
    case (1):  
        print ("El valor es 1");  
        break;  
    case (2):  
        print ("El valor es 2");  
        break;  
    case (3):  
        print ("El valor es 3");  
        break;  
    default :  
        print ("No es ninguno de los valores.")  
}
```

Resultado:

El valor es 3



### Clase 1: Sintaxis, variables, expresiones y estructuras

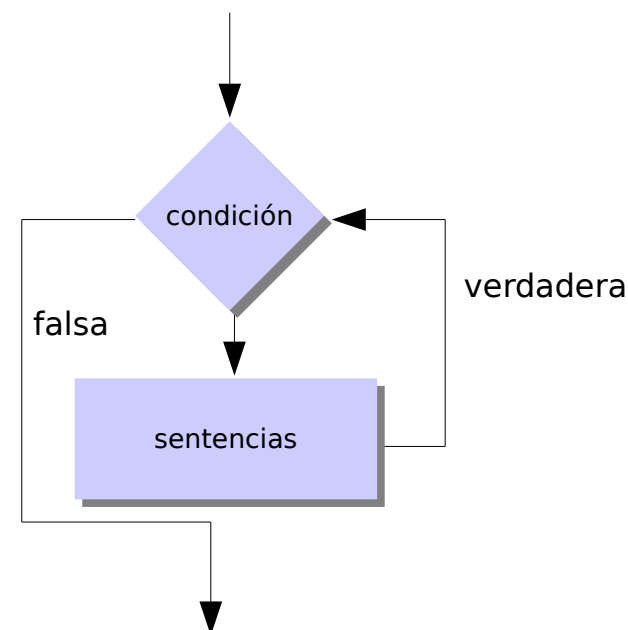
## Estructuras de Control Iterativas

Estructura Repetitiva **while**

```
while (condición){  
    sentencia_1  
    sentencia_2  
    sentencia_n  
}
```

## Estructuras de Control - Ejemplos

```
$i = 1;  
while ($i <= 3){  
    print ("Fila ".$i);  
    print ("<br>");  
    $i++;  
}
```



Resultado:  
Fila 1  
Fila 2  
Fila 3



### Clase 1: Sintaxis, variables, expresiones y estructuras

## Estructuras de Control Repetitivas

Estructura Selectiva **for**

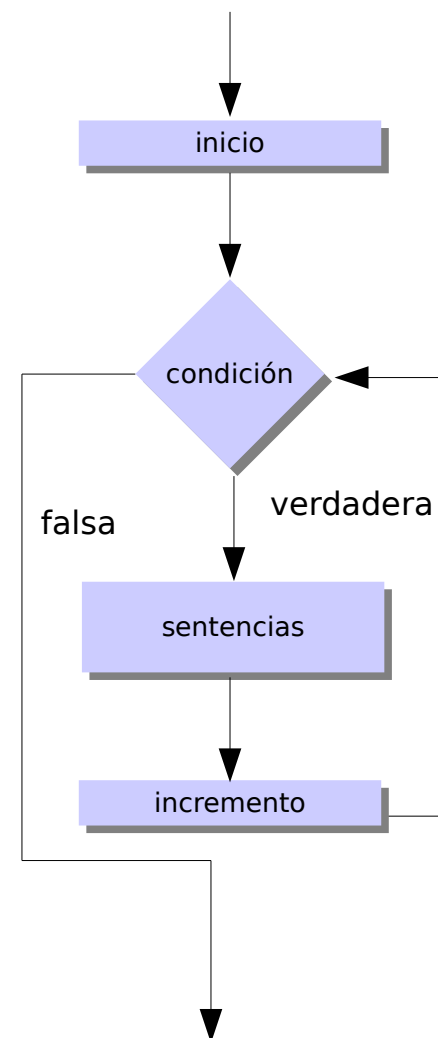
```
for (inicio; condición; incremento){  
    sentencias  
}
```

## Estructuras de Control - Ejemplo

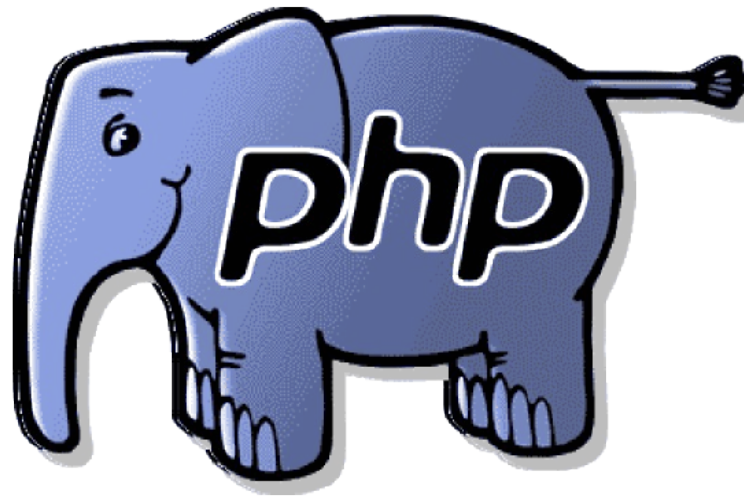
```
for ($i=1; $i <=3; $i++){  
    print ("Fila ".$i);  
    print ("<br>");  
}
```

Resultado:

Fila 1  
Fila 2  
Fila 3



**¿Dudas?**



**¿Consultas?**



## Información de contacto

### Web:

<http://www.gugler.com.ar>

<http://campusvirtual.gugler.com.ar>

<http://www.facebook.com/gugler.com.ar>

<http://www.twitter.com/cgugler>

### Mail:

[contacto@gugler.com.ar](mailto:contacto@gugler.com.ar)

[academica@gugler.com.ar](mailto:academica@gugler.com.ar)

[administracion@gugler.com.ar](mailto:administracion@gugler.com.ar)