

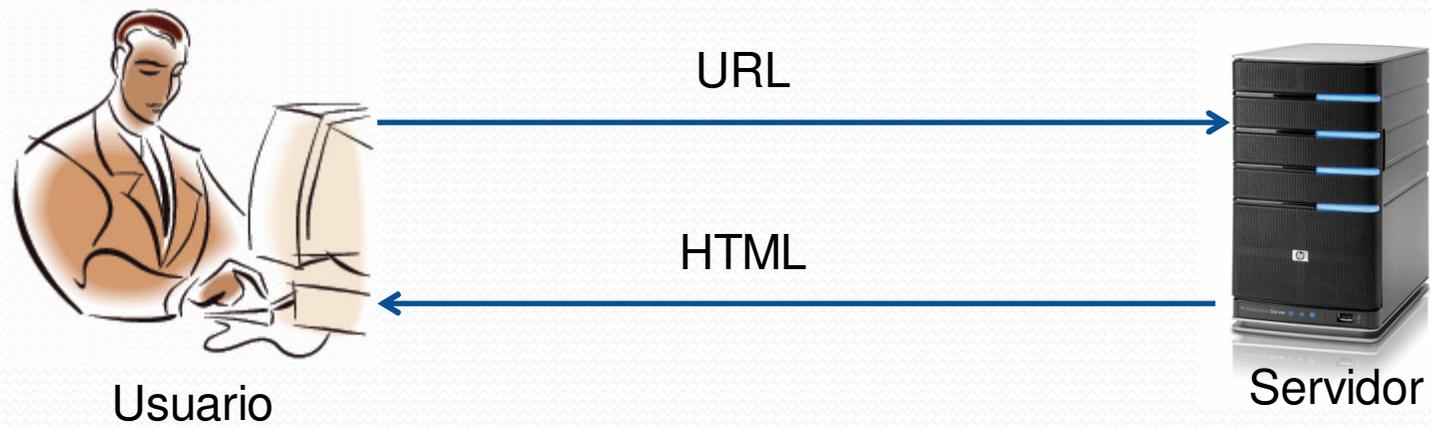
# Guía Básica de PHP

Taller de Proyecto

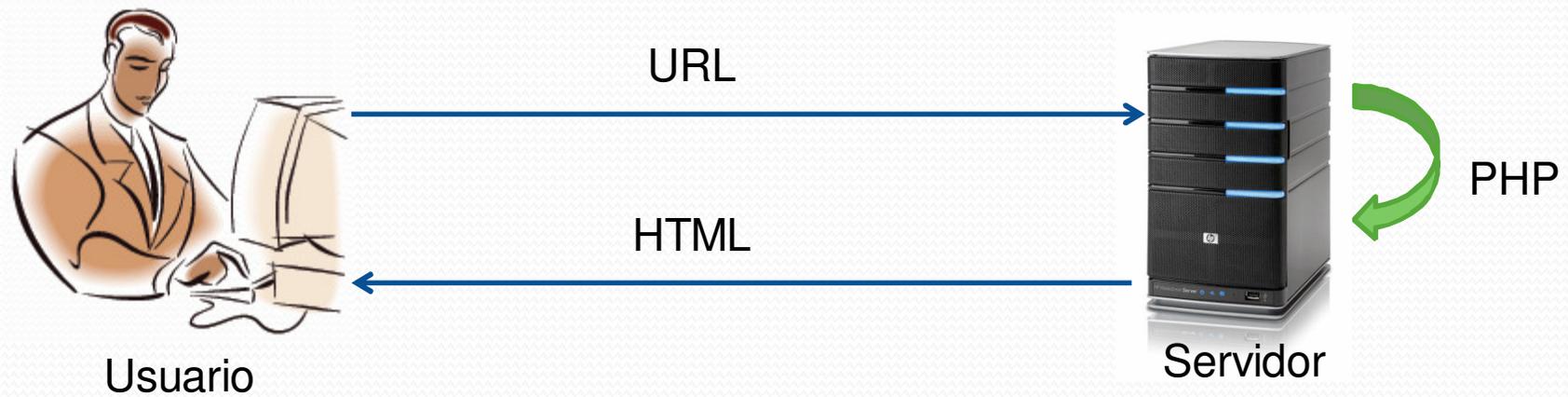
# ¿Qué es PHP?

- Acrónimo de “Hypertext Preprocessor”
- Lenguaje interpretado de alto nivel
- Embebido en páginas HTML
- Sintaxis similar a C, Java y Perl
- Permite crear página dinámicas

# ¿Cómo Funciona HTML?



# ¿Cómo Funciona PHP?



# PHP: Hola Mundo!

- Crear un archivo llamado hola.php

```
<html>
<body>
  <h1>Ejemplo 1</h1>

  <?php print("Hola Mundo!"); ?>

</body>
</html>
```

- <http://www.cec.uchile.cl/~<user>/hola.php>

# PHP: Hola Mundo!

- Otro ejemplo usando variables

```
<html>
<body>
<?php
$texto = "hola mundo!";
?>

<h1>Otro Ejemplo</h1>
<?=$texto ?>
</body>
</html>
```

- <http://www.cec.uchile.cl/~<user>/hola.php>

# PHP: variables

- Se debe anteponer un signo '\$'
- No pueden llevar espacio
- No hay que declarar el tipo de dato

## JAVA

```
Int numero = 1;  
String frase = new String('hola mundo');
```

```
<?php  
$variable = "hola";  
$variable_2 = 'chao';  
$entero = 5;  
$decimal = 1.1232242;
```

# PHP: operadores

- Los mismos que en java...

```
<?php  
  
$a = 20;  
$b = 3;  
  
print( $a + $b ); // 23  
print( $a - $b ); // 17  
print( $a * $b ); // 60  
print( $a / $b ); // 6.666666666666676  
print( $a % $b ); // 2
```

# PHP: Arreglos

- No se requiere declarar el largo

```
<?php
$frutas = array( "naranja", "manzana", "durazno" );
$frutas[3] = "pera";
$frutas[] = "melón"; <- tomará automáticamente la pos. 4
$frutas[23] = "papaya"
```

- Resultado

0	1	2	3	4	23
naranja	manzana	durazno	pera	melón	papaya

# PHP: Arreglos asociativos

- En vez de usar números, usamos strings

```
<?php
$lista = array(
    'fruta' => 'manzana',
    'verdura' => 'lechuga',
    'numero' => 4,
    1 => 'algo',
);
$lista['cereal'] = 'maiz';
```

- Resultado

fruta	verdura	numero	1	cereal
manzana	lechuga	4	algo	maiz

# PHP: Arreglos asociativos

- Ejemplo

```
<?php
$lista = array(
    'fruta' => 'manzana',
    'verdura' => 'lechuga',
);

print( 'una fruta es '.$lista['fruta'] );
print( 'una verdura es '.$lista['verdura'] );
```

# PHP: Iteradores

- Parecidos a java. Ej:

```
$lista = array( 'paula', 'javier', 'manuel', 'pedro' );  
  
for( $i=0; $i < count( $lista ); ++$i ){  
    print( 'El valor en la posición '.$i.' es '.$lista[$i].' <br />' );  
}
```

- Equivalente:

```
$lista = array( 'paula', 'javier', 'manuel', 'pedro' );  
$i = 0;  
while( $i < count( $lista ) ){  
    print( 'El valor en la posición '.$i.' es '.$lista[$i].' <br />' );  
    ++$i;  
}
```

# PHP: Iteradores

- También es equivalente a:

```
$lista = array( 'paula', 'javier', 'manuel', 'pedro' );  
  
$i = 0;  
do {  
    print( 'El valor en la posición '.$i.' es '.$lista[$i].' <br />' );  
    ++$i;  
} while( $i < count( $lista ) );
```

# PHP: Iteradores

- ¿Cómo recorro un arreglo asociativo?

```
$lista = array(  
    'fruta' => 'manzana',  
    'verdura' => 'lechuga',  
);
```

- Así:

```
foreach( $lista as $llave => $valor ){  
    print( $llave.' es un '.$valor.' <br />' );  
}
```

# PHP: Condiciones

- `&&`             $\leftarrow$  y
- `||`              $\leftarrow$  o
- `==`             $\leftarrow$  igual a
- `!=`             $\leftarrow$  distinto que
- `<`               $\leftarrow$  menor que
- `>`               $\leftarrow$  mayor que
- `<=`             $\leftarrow$  menor o igual a
- `>=`             $\leftarrow$  mayor o igual a
- `===`            $\leftarrow$  ¿? (prueben `if( 0 == false )` )

# PHP: Condiciones

```
$lista = array( 'javier', 'juan', 'paula', 'manuel' );  
  
for ( $i =0; $i < count( $lista ); ++$i ) {  
  
    if( $lista[$i] == 'javier' ) {  
        print( "este es javier!" );  
    } else if( $lista[$i] == 'juan' ) {  
        print( "y de donde salio Juan?" );  
    } else {  
        print( "El resto cae aquí" );  
    }  
}  
}
```

# PHP: Formularios

- ¿Cómo recibimos información del usuario?

```
<html>
<head>
  <title>Ejemplo de Formulario</title>
</head>
<body>
  <h1>Formulario de Ingreso</h1>
  <form method="get" action="validar.php">
    <input type="hidden" name="accion" value="entrar" />
    Nombre: <input type="text" name="nombre" size="40" /><br />
    Password: <input type="password" name="password" size="20" />
    <input type="submit" name="boton" value="Entrar" />
  </form>
</body>
</html>
```

# PHP: Formularios

- Y nuestro archivo validar.php sería:

```
<?php
```

```
print( 'El nombre ingresado es ' . $_REQUEST['nombre'] );  
print( '<br />El apellido es ' . $_REQUEST['apellido'] );
```

# PHP: Formularios

- Las variables enviadas las recibe PHP en el arreglo `$_REQUEST`
- Hay otras variables que se llenan dependiendo del method del form:
  - `<form method="get" ...>`      `<- $_GET[]`
  - `<form method="post" ...>`      `<- $_POST[]`
- `$_REQUEST = $_COOKIES + $_POST + $_GET;`

# PHP: Formularios

- POST (La variable se envía de forma invisible)

<http://www.hola.com>

- GET (Las variables son visibles en la dirección)

<http://www.hola.com?id=Hola&pass=password>

est - Buscar con Google - Mozilla Firefox

Inicio Editar Ver Historial Marcadores Herramientas Ayuda

  <http://www.google.cl/search?q=test&ie=utf-8&oe=utf-8&aq=t&rls=org.mozilla:es-AR:official&client=firefox-a>

# PHP: Funciones útiles

- Manejo de archivos

```
<?php

// meter el contenido de un archivo en un arreglo
$lineas = file( $archivo );

// leer el contenido de un archivo en un solo string
$todo = file_get_contents( $archivo );

// escribir el contenido de una variable en un archivo
file_put_contents( 'archivo.txt', $variable );

// revisar si existe un archivo
print( file_exists( $archivo ) ? "Si existe" : "nones..." );
```

# PHP: Funciones útiles

- Manejo de arreglos

```
<?php

// separar un string en varios pedazos
$nombrres = "pedro#antonio#vargas#naser";
$lista = explode( "#", $nombrres );
// array( 'pedro', 'antonio', 'vargas', 'naser' )

// concatenar un arreglo con un string
$lista = array('pedro', 'antonio', 'vargas', 'naser' );
$nombrres = implode( "#", $lista );
// "pedro#antonio#vargas#naser"

// sacar el ultimo elemento del arreglo
array_pop( $lista );

//sacar el primer elemento del arreglo
array_shift( $lista );
```

# PHP: Funciones útiles

- ¿Cómo Redirigo al usuario a otra página?
  - Se envía un header especial al browser llamado Location

```
<?php  
  
if( $_REQUEST['password'] != 'hola' ){  
    header( 'Location: error.php' );  
    exit(); <- es muy importante que vaya esto  
} else{  
    print( "Bienvenido !!" );  
}
```

# PHP: Referencias

- ¿Donde puedo consultar las funciones de PHP?
  - <http://cl.php.net> <- castellano
  - <http://www.php.net> <- ingles

# Expresiones regulares

- Sirven para calzar patrones.

```
$texto = "pedro#vargas#26/07/1981|javier#villanueva#05/09/1981#";  
  
if( ! preg_match( '/(.+)#(.+)#(\d{2})\V(\d{2})\V(\d{4})/' , $texto, $calces ) ){  
    exit( 'No encontré nada' );  
}
```

- \$calces

0	1	2	3	4	5
...	pedro	vargas	26	07	1981

# Expresiones regulares

- `^` <- calza al inicio de la linea
- `$` <- calza al final de la linea
- `.` <- calza cualquier caracter excepto `\n`
- `+` <- aparece 1 o mas veces
- `*` <- aparece 0 o mas veces
- `()` <- agrupa un calce y lo guarda en `$c[1], $c[2]...`
- `\d` <- cualquier digito
- `\D` <- cualquier no digito
- `\s` <- cualquier espacio (espacios, tabs)
- `\S` <- cualquier no espacio
- `[A-Z]` <- de la A a la Z
- `[^A-Z]` <- que no contenga de la A a la Z



# Expresiones regulares

- Documentación:
  - <http://cl.php.net/manual/es/reference.pcre.pattern.syntax.php>