**4.1 Introducción Al Lenguaje**

JavaScript es un lenguaje de programación usado principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones activadas al pulsar botones y ventanas con mensajes de aviso al usuario.

Técnicamente, JavaScript es un lenguaje de programación interpretado, y no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.

A pesar de su nombre, JavaScript no guarda ninguna relación directa con el lenguaje de programación Java. Legalmente, JavaScript es una marca registrada de la empresa Sun Microsystems. Su nombre es exclusivamente por marketing, ya que Java era la palabra de moda en el mundo informático y de Internet de la época de desarrollo.

**Cómo incluir JavaScript en documentos XHTML**

La integración de JavaScript y XHTML es muy flexible, ya que existen al menos tres formas para incluir código JavaScript en las páginas web en esta sección solo se presentaran dos de ellas.

**Incluir JavaScript en el mismo documento XHTML**

El código JavaScript se encierra entre etiquetas <script></script> y se incluye en cualquier parte del documento. Aunque es correcto incluir cualquier bloque de código en cualquier zona de la página, se recomienda definir el código JavaScript dentro de la cabecera del documento (dentro de la etiquetas <head></head>):

Este método se emplea cuando se define un bloque pequeño de código o cuando se quieren incluir instrucciones específicas en un determinado documento HTML que completen las instrucciones y funciones que se incluyen por defecto en todos los documentos del sitio web.

El principal inconveniente es que si se quiere hacer una modificación en el bloque de código, es necesario modificar todas las páginas que incluyen ese mismo bloque de código JavaScript.

**Definir JavaScript en un archivo externo.**

Las instrucciones JavaScript se pueden incluir en un archivo externo de tipo JavaScript que los documentos XHTML enlazan mediante las etiquetas <script></script> . Colocadas entre las etiquetas <head></head>.

Se pueden crear todos los archivos JavaScript (\*.js) que sean necesarios y cada documento XHTML puede enlazar tantos archivos JavaScript como necesite.

Este método requiere definir el atributo src, que es el que indica la URL correspondiente al archivo JavaScript que se quiere enlazar.

Los archivos de tipo JavaScript son simples documentos de texto con la extensión js que se pueden crear con cualquier editor de texto como Notepad, Wordpad, UltraEdit, Vi, etc.

La principal ventaja de enlazar un archivo JavaScript externo es que se simplifica el código XHTML de la página, además de la capacidad de reutilizar el mismo código JavaScript en páginas del sitio web y que cualquier modificación realizada en el archivo JavaScript se refleja inmediatamente en todas las páginas XHTML que lo enlazan.

**4.2 Estructuras De Control**

Las estructuras de control nos permiten controlar el flujo del programa: tomar decisiones, realizar acciones repetitivas etc, dependiendo de unas condiciones que nosotros mismos establezcamos. Así podemos hacer un script que nos salude cada día de la semana de una manera diferente. O por ejemplo hacer un script que nos pida la contraseña una y otra vez hasta que suministremos la opción correcta.

**1.- La toma de decisiones if-else:**

Quizá la estructura de control más básica y la más utilizada sea if-else:

if (condicion) {  
// instrucciones que hay que ejecutar si la condición es verdadera  
} else {  
// Instrucciones que hay que ejecutar si la condición es falsa  
}  
En castellano sería algo así:

Si (Tal cosa es cierta) {  
//  Haz esto  
} En caso contrario {  
// Haz esto otro  
}  
Veamos un ejemplo real: En un formulario preguntaremos al usuario como se llama, pasaremos las variables a la página “saludos.php” . En saludos.php mostraremos un saludo diferente en función de si el usuario soy yo o es cualquier otro

Formulario

<html>  
<head>  
<title> Ejemplo básico de If </title>  
</head>  
<body>

<form action="saludos.php" method="get">

**¿Quién eres?**<br />  
<input type="text" name="nombre" />

<input type="submit" name="reset" value=" Enviar " />  
</form>  
</body>  
</html>

saludos.php

<html>  
<head>  
<title> Ejemplo básico de If </title>  
</head>  
<body>  
<h1> Página de bienvenida</h1>  
<?php // Empieza el código php  
if ($nombre=="Coder") {  // Si el nombre es Coder  
echo(" Hombre $nombre! Ya estás cotilleando por aquí otra vez!"); // saludo especial  
}else {  
echo("Hola, $nombre, gracias por visitarnos"); // saludo para los demás  
}  
?>  
</body>  
</html>  
  
La cláusula else (la parte del if-else que indica lo que hay que hacer si la condición es falsa) es opcional. No es necesario ponerla. Por ejemplo puedo querer un script que sólo me salude a mi. En este caso, cambiaría saludos.php de la siguiente manera:

<html>  
<head>  
<title> Ejemplo básico de If </title>  
</head>  
<body>  
<h1> Página de bienvenida</h1>  
<?php // empieza el código php  
if ($nombre=="Coder") { // Si el nombre es Coder  
echo(" Hombre $nombre! Ya estás cotilleando por aquí otra vez!"); // Saludo para Coder  
}  
?>  
</body>  
</html>

**NOTA:**Cuidado con el operador ==, es importante recordar que hay que utilizar un doble signo igual. Si utilizas sólo “=”, estarás utilizando el operador de asignación, es decir estableciendo que $nombre es igual a Coder con lo que la condición siempre sería cierta

Las condiciones pueden complicarse mucho más, supongamos que queremos dejarte pasar sólo sí además de llamarte Coder, te sabes la contraseña, que en nuestro caso es “foros”:

**Formulario:**

<html>  
<head>  
<title> Ejemplo básico de If </title>  
</head>  
<body>

<form action="saludos.php" method="get">

**¿Quién eres?**<br />  
<input type="text" name="nombre" /><br />  
**¿Y tu Contraseña?**<br />  
<input type="text" name="pass" />

<input type="submit" name="reset" value=" Enviar " />  
</form>  
</body>  
</html>

saludos.php

<html>  
<head>  
<title> Ejemplo básico de If </title>  
</head>  
<body>  
<h1> Página de bienvenida</h1>  
<?php // Empieza el código php  
if ($nombre=="Coder" and $pass =="foros") {  
// Si el nombre es Coder y la contraseña foros  
echo(" Hombre $nombre! Ya estás cotilleando por aquí otra vez!"); // saludo especial  
}else {  
echo("ahhhhhhhhhhh como no te sabes el nombre y la contraseña, no puedes pasar"); // saludo para los demás  
}  
?>  
</body>  
</html>

Además tenemos la instrucción elseif, que como su nombre indica, es una combinación de if y else. Añade una nueva condición en el caso de que la primera fuera falsa y sólo ejecuta las instrucciones si esta nueva condición es verdadera. Es decir,

Veamos un ejemplo:

if ($a > $b) {  
// Ejecuta esto si a es Mayor que b  
echo("a es mayor que  b");

} elseif ($a == $b) {  
// si a no es mayor que be, comprueba si a es igual a b  
echo("a es igual que  b");  
} else {  
// si no se cumplen las anteriores condiciones ejecuta ésta  
echo ("a es menor  que b");  
}

**2- Bucles: While**

If else nos permitía decidir si queríamos ejecutar o no una instrucción en función de una condición, los bucles nos permiten utilizar una condición para determinar cuantas veces queremos ejecutar una instrucción.:

While (condicion) {

// Instrucciones a ejecutar una y otra  
//vez mientras la condición sea cierta  
}

El funcionamiento es muy parecido al del if-else, la diferencia estriba en que si la condición es cierta y se ejecuta la orden, la condición vuelve a comprobarse una y otra vez mientras siga siendo cierta. En cambio en el cambio del if-else sólo se comprueba la condición una vez, si es cierta se ejecuta la instrucción y se sigue con el programa.

**Veamos un ejemplo:**

$cantidad = 1;  
while ($cantidad < 10) {  
echo ("$cantidad ");  
$cantidad++;  
}

Esto traducido a castellano sería:

$cantidad = 1; $cantidad es igual a uno  
while ($cantidad < 10) Mientras el valor de la variable $cantidad sea menor que 10  
echo (“$cantidad “); Escribe el valor de cantidad  
$cantidad++ (suma 1 al valor que tenga $cantidad, también podríamos haber escrito $cantidad = $cantidad +1)

Veamos como funciona el script paso a paso,

$Cantidad es igual a 1  
1 es menor que 10  
Escribo 1 en pantalla  
$cantidad = 2 (1+1)  
2 es menor que 10  
Escribo 2 en pantalla  
$cantidad = 3 (2+1)…

Si ejecutamos este script, en pantalla veríamos 1 2 3 4 5 6 7 8 9

**3- Bucles: do While**

Los bucles do while, son muy parecidos a los bucles while, sólo que comprobamos si la condición es cierta al final del bucle en lugar de al principio. Así nuestras instrucciones se ejecutan siempre al menos una vez, lo que no ocurre en las condiciones while si la condición es falsa.

$cantidad = 1;  
do {  
echo ($cantidad);  
} while ($cantidad<1);

Traducido sería

$cantidad = 1

do ( // ejecuta  
echo ($cantidad); escribe el valor de cantidad  
while ($cantidad<1); mientras cantidad sea mayor que 1 -> la condición es falsa luego el programa sale del bucle, pero ha escrito una vez el valor de $cantidad. El resultado en pantalla sería 1

**4- Bucles: for**

Los bucles for funcionan exactamente igual que en C . Con una sola instrucción controlamos todo el bucle, así que para los vaguetes está muy bien.

for (valor inicial; condicion; ejecutar en cada blucle) {

// Instrucciones a ejecutar mientras la condición sea cierta  
}

**Ejemplo**

for ($cantidad = 1; $cantidad <10; $i++) {  
echo("$cantidad");  
}

Traducido es igual a:

Cantidad es igual a 1, mientras cantidad sea menor que 10 escribe cantidad en pantalla y después súmale 1 al valor de cantidad. Es decir, es exactamente el mismo ejemplo que vimos en el bucle while pero de otra manera.

**5- Swich/case**

Nos permite ejecutar un bloque de código u otro dependiendo del valor de una variable. Veamos un ejemplo.  
Primero para asignar un valor a la variable, vamos a hacernos un formulario:

<html>  
<head>  
<title> Ejemplo de switch </title>  
</head>  
<body>

<form action="switch.php" method="get">

**¿Quién eres?**<br />

<input type="radio" name="quien" value="cgus" /> cgus<br />  
<input type="radio" name="quien" value="xela" /> xela<br />  
<input type="radio" name="quien" value="adanedhel" /> adanedhel<br />  
<input type="radio" name="quien" value="jorgito" /> jorgito<br />  
<input type="submit" name="reset" value=" Enviar " />  
</form>  
</body>  
</html>

Switch.php

<html>  
<head>  
<title>Ejemplo de Swich</title>  
</head>

<?php

switch($quien) {  
case "cgus": // Primera posibilidad

echo (" $quien es todo un leader ");  
break;  
case "adanedhel": // segunda posibilidad  
echo ("$quien es un elfo");  
break;  
case "xela": // tercera posibilidad  
echo ("$quien es normal");  
break;  
default: // Posiblidad si no es ninguna de las anteriores  
echo ("$quien es un desconocido");  
}  
?>  
</body>  
</html>

Tenéis más información sobre las estructuras de control en el manual de php

<http://www.php.net/manual/es/control-structures.php>

**Sintaxis alternativa**

PHP ofrece una sintaxis alternativa para alguna de sus estructuras de control; a saber, if, while, for, y switch. Puede hace el código más legible en algunos casos, especialmente en las páginas multiuso que vamos a ver a continuación:

If con construcción alternativa

<?php if ($a==5): ?>  
A es igual a 5  
<?php endif; ?>

if con elseif y else en el formato alternativo:  
  
if ($a == 5):  
print "a es igual a 5";  
print "...";  
elseif ($a == 6):  
print "a es igual a 6";  
print "!!!";  
else:  
print “a no es ni 5 ni 6”;  
endif;

While utilizando la construcción alternativa

$i = 1;  
while ($i <= 10):  
print $i;  
$i++;  
endwhile;

For con la construcción alternativa

for (expr1; expr2; expr3): sentencia; ...; endfor;

**Páginas Multiuso**

Páginas multiuso son aquellas que pueden mostrar un contenido completamente diferente dependiendo de alguna condición. De esta forma, podemos hacer varias cosas en un solo documento.

La estructura habitual de una página multiuso es:

<html>  
<head>  
<title> Página Multiusos </title>  
</head>  
<body>

<php?  
If (condición) { ?>  
<!-- HTML que se mostará si la condición es cierta -->  
<?php }else{ ?>  
<!-- html que se mostratará si la condición es falsa -->  
<?php } ?>

</body>  
</html>

O utilizando la sintaxis alternativa que acabamos de ver:

<html>  
<head>  
<title> Página Multiusos </title>  
</head>  
<body>

<php?  
if (condición): ?>  
<!-- HTML que se mostará si la condición es cierta -->  
<?php else: ?>  
<!-- html que se mostratará si la condición es falsa -->  
<?php endif; ?>

</body>  
</html>

Esto puede parecer un poco confuso. Pero es precisamente una de las características que están haciendo muy popular al PHP: su capacidad de pasar de php a html en cualquier momento y todas las veces que sea necesario.

**Ejemplo**

Para nuestro ejemplo vamos a utilizar el ejemplo que vimos para switch y utilizar una sola página (multiuso.php) para el formulario y las respuestas en lugar de las dos que teníamos antes

<html>  
<head>  
<title> Página Multiusos </title>  
</head>  
<body>  
<?php  
if (!isset($quien)): ?>

<html>  
<head>  
<title> Ejemplo de switch </title>  
</head>  
<body>

<form action="multiuso1.php" method="get">

**¿Quién eres?**<br />

<input type="radio" name="quien" value="cgus" /> cgus<br />  
<input type="radio" name="quien" value="xela" /> xela<br />  
<input type="radio" name="quien" value="adanedhel" /> adanedhel<br />  
<input type="radio" name="quien" value="jorgito" /> jorgito<br />  
<input type="submit" name="reset" value=" Enviar " />  
</form>  
</body>  
</html>

<?php else:

switch($quien) {  
case "cgus": // Primera posibilidad

echo (" $quien es todo un leader ");  
break;  
case "adanedhel": // segunda posibilidad  
echo ("$quien es un elfo");  
break;  
case "xela": // tercera posibilidad  
echo ("$quien es un normal");  
break;  
default: // Posiblidad si no es ninguna de las anteriores  
echo ("$quien es un desconocido");  
}

endif;  
?>

</body>  
</html>

**for each**

Es un tipo de bucle especial para matrices. Cómo no hemos visto el tema de las matrices todavía, de momento no lo vamos a explicar. Tocaremos este tema al hablar de las matrices.