



# Diseño web

## Tema 2

# Lenguajes de marcas. HTML5

- Permite crear páginas web usando simplemente un editor de textos y un navegador web.
- Extensión .html o .htm.
- Editores WYSIWYG (What you see is what you get): Dreamweaver, Kompozer, Amaya...

# Leng. de marcas. Conceptos básicos

- HTML: lenguaje de marcas que sirve para describir el contenido y estructura de las páginas web, que pueden ser interpretadas por un navegador web.
- Sí puede: enlazar a otros archivos, diseñar formularios,...
- No puede: procesar los datos de los formularios. Requiere lenguaje del lado del servidor (PHP, JSP...)

# Leng. de marcas. Conceptos básicos

- XHTML: variante más restrictiva y ordenada de HTML que utiliza la sintaxis propia de XML.

# Leng. de marcas. Estándares

- Se estandariza para homogeneizar su uso.
- HTML5: última versión.

# Normas

- Etiquetas
  - Definen los elementos.
  - Se delimitan por los signos < y >.
  - Pueden ser abiertas (una sola indica el principio y el fin: <hr>) o cerradas (<p></p> o bien </br>).

EJERCICIO I

# Normas

- Atributos

- Configurar las características propias de las etiquetas.
- Se escriben a continuación de la palabra reservada de la etiqueta de apertura.
- Ejemplo:

`<a href="www.google.es" target="_blank">`

# Normas

- Comentarios

- Líneas de texto que no son interpretadas por el navegador. No se visualizan en la página.
- Realizar aclaraciones sobre el código fuente.
- Se escriben entre `<!--` y `-->`
- Ejemplo:
- `<!-- Esto es un comentario -->`



# Estructura básica

`<html>`

Inicio del documento

`<head>`

Inicio de la cabecera

`<title>`

Inicio del título

`</title>`

`</head>`

`<body>`

Inicio del cuerpo

`</body>`

`</html>`

# Estructura básica

`<!Doctype>`

Indica el tipo de documento que estamos creando.

Es la primera línea del archivo.

EJERCICIO 2

# Estructura básica

- Cabecera del documento: `<head>`
  - Se definen los elementos generales: título, codificación de caracteres (UTF8...) y la hojas de estilos asociada entre otros.

```
<head>
```

```
    <title>Mi primera página web</title>
```

```
</head>
```

# Estructura básica

- Cabecera del documento: <head>
  - <meta>: indica las palabras clave y contenido de la página. Usada por los buscadores.
  - Tiene dos atributos: name (indica el nombre de la página) y content (indica el contenido o palabras clave).

<head>

    <meta name =“description” content=“Página personal de Javi”>Mi primera página web

</head>

- Refresco automático:

<meta http-equiv=“refresh” content=“15”;  
url=http://ww.google.es>

- <base href=<http://www.mipagina.es/recursos>>

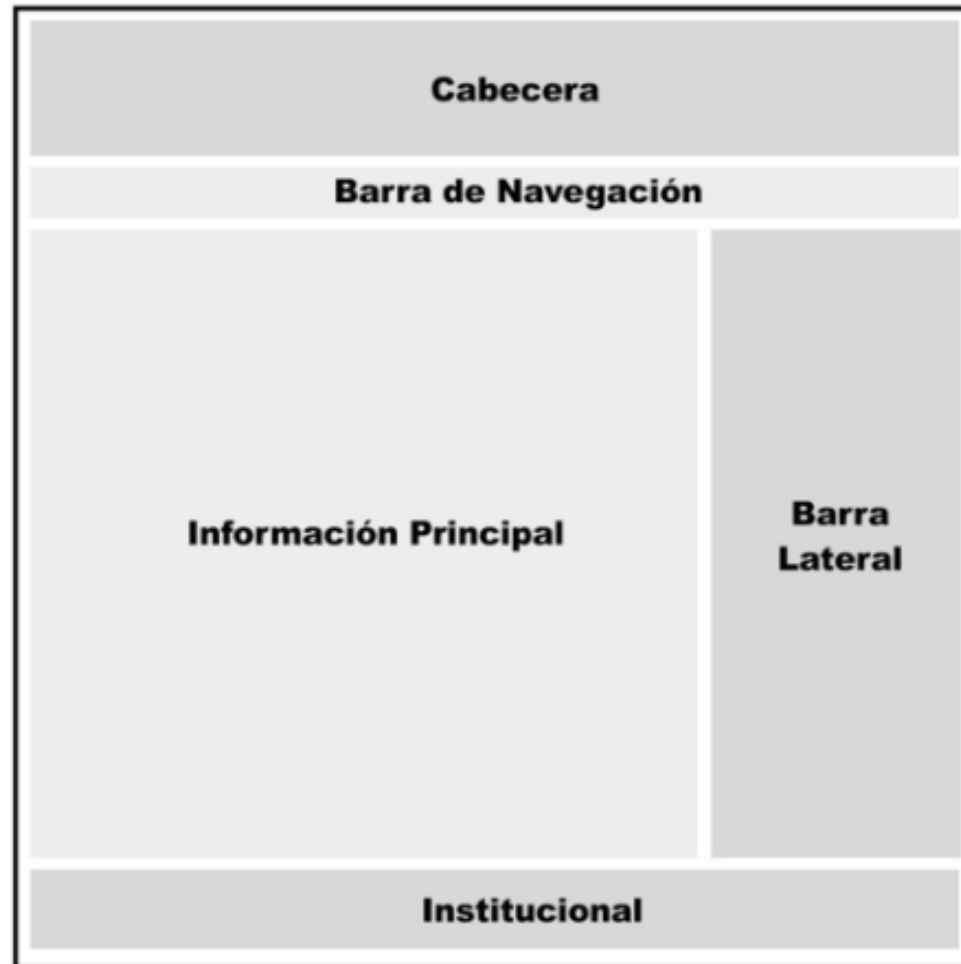
# Estructura básica

- Cuerpo del documento: `<body>`
  - Contiene el cuerpo de la página web.

EJERCICIO 3,4,5

EJERCICIO PÁGINA 25

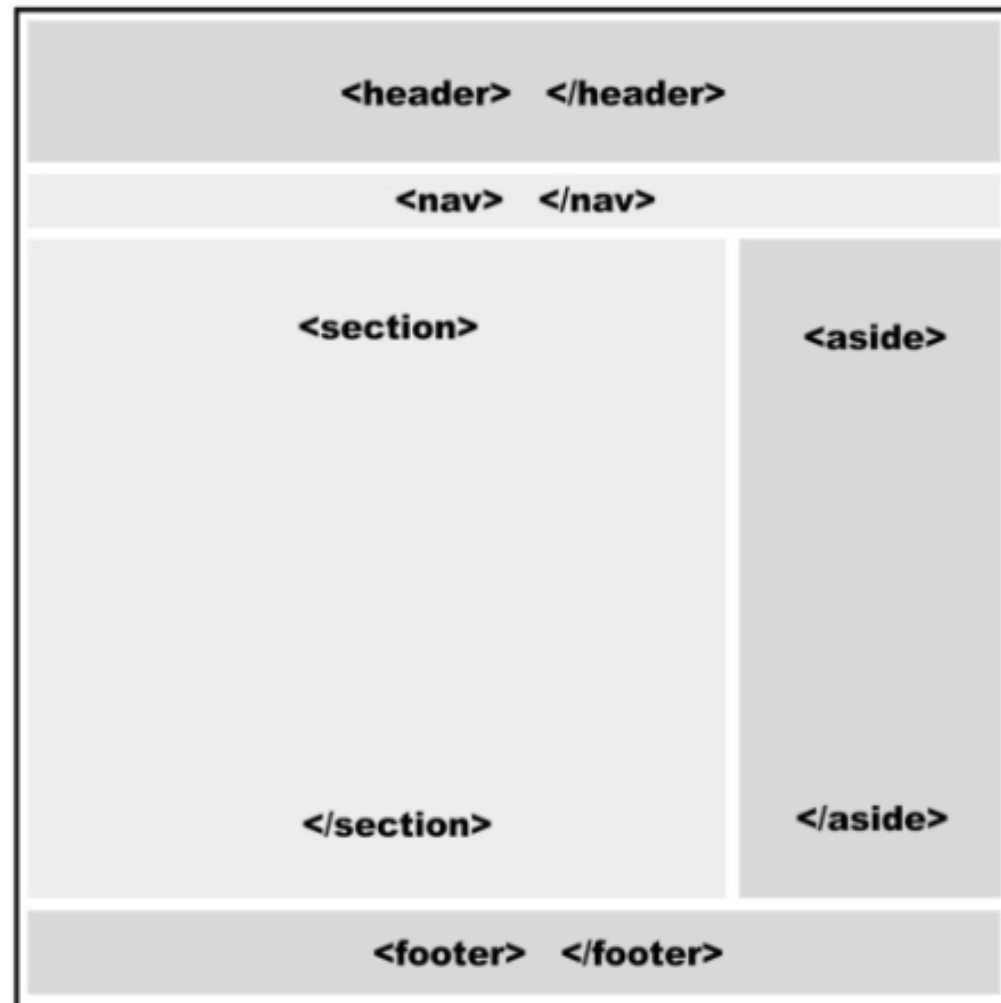
# Estructura básica



# Estructura básica

- Nuevas etiquetas HTML5: definir contenidos agrupando etiquetas.
  - `<header>`: encabezado de página.
  - `<nav>`: enlaces de navegación.
  - `<article>`: artículo publicado.
  - `<section>`: sección de un documento o artículo.
  - `<aside>`: barra lateral
  - `<footer>`: pie de página

# Estructura básica





# Espaciados, saltos de línea y párrafos

- `<pre></pre>`: obliga al navegador a visualizar el texto respetando las tabulaciones, espacios, etc.
- `<br>` representa un salto de línea.
- `<hr>` salto de párrafo a nivel semántico: cambio de escena en una historia o similar.
- `<blockquote>`: insertar citas en forma de bloque. Sangra y admite el atributo cite.
- `<p></p>`: nuevo párrafo.

# Cabeceras

- `<h1></h1>...<h6></h6>`
  - Crear títulos o encabezados de diferente tamaño.
  - H1 es el nivel mayor.
  - Incluyen detrás un doble salto de línea.

EJERCICIO 6,7,8

# Semántica en textos

- Etiquetas que dan algún significado a una parte del contenido:
- `<b>...</b>` → (bold) resalta el texto en negrita.
- `<i>...</i>` → (italic) resalta el texto en cursiva.
- `<u>...</u>` → (underline) resalta el texto subrayando.
- `<s>...</s>` → muestra el texto tachado.

EJERCICIO 9

# Listas de elementos

- Listas numeradas.
  - `<ol></ol>`
  - Cada elemento se define por `<li></li>`
  - Atributo `start=número` → indica qué número es el primero de la lista. Por defecto 1.
  - Atributo `type=tipo` → indica el tipo de numeración usada.
    - 1: expresar números.
    - a/A: letras minúsculas/mayúsculas.
    - i/I: números romanos en minúscula/mayúscula.

# Listas de elementos

- Listas sin numerar.
  - `<ul></ul>`
  - Cada elemento se define por `<li></li>`
  - Atributo `type=disk, circle, square` → indica el tipo de marca.

# Listas de elementos

- Listas de definición o glosario.
  - Muestran términos junto con su respectiva definición.
  - `<dl></dl>`
  - Los términos se indican con `<dt></dt>`.
  - La definición se indica con `<dd></dd>`.

EJEMPLO PÁGINA 30

# Imágenes

- Etiqueta `<img>` + atributo `src`.
- Atributo `alt`: indicar texto alternativo que represente el contenido de la imagen.
- `<figure>` y `<figcaption>`: permiten englobar una imagen y su contenido adicional o leyenda.

EJERCICIO 10

EJEMPLO PÁGINA 31

# Vínculos o hiperenlaces

- Permite acceder a otro recurso (otra web, otro lugar de la propia página, archivo...)
- Etiqueta `<a>` + atributo href.
- Atributo target opcional.



# Vínculos o hiperenlaces

- Anclas y vínculos internos.

`<a id="nombreancla">TEXTO ANCLADO </a>`

`<a href="#nombreancla">IR A TEXTO </a>`

EJEMPLO PÁGINA 33

EJERCICIOS 12, 13 y 14

# Tablas

- No deben ser usadas como herramienta de diseño.
  - `<table></table>`
  - Atributo border: “1” o “” para mostrar borde.
  - `<caption></caption>`: título de la tabla.
  - `<tr></tr>`: nueva fila.
  - `<td></td>`: nueva columna.
  - `<th></th>`: columnas de cabecera.
  - Combinar celdas: atributos `rowspan` y `colspan`.

# Formularios

- Recopilar datos de los visitantes de la web.
- La información se envía a un programa del servidor web a través de una url, o al correo del administrador a través de un botón.
- Etiqueta `<form></form>`
- Atributos: action, method, enctype,

# Formularios

- Etiqueta `<input>`: crear diferentes elementos de entrada interactivos para editar la información del formulario.
- Atributo `name`: nombre del elemento.
- Atributo `type`: especifica controles en el formulario.
  - `Hidden`: entrada oculta.
  - `Text` y `search`: editar línea de texto.
  - Etc. (ver libro).

# Formularios

- Etiqueta `<select>`: crear lista desplegable con opciones especificadas en la etiqueta `<option>`.
- Atributos:
  - Name: nombre de la lista.
  - Size: número de opciones a **mostrar**.
  - Multiple: permite seleccionar varias opciones de la lista.

EJERCICIOS 22 y 23

EJEMPLOS PÁGINA 35, 38 y 39

# Hojas de estilos CSS

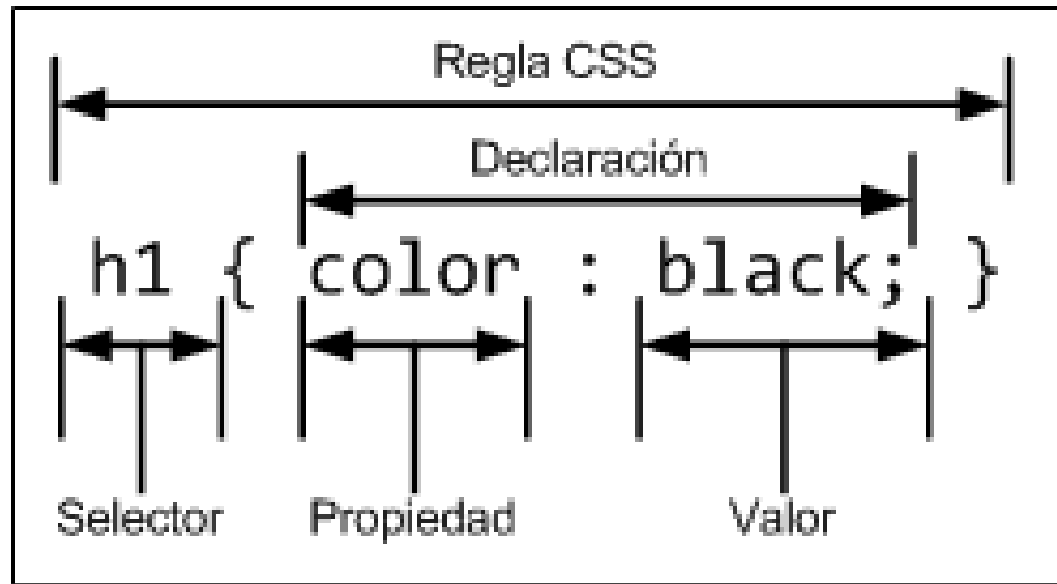
- Dota de estilo a los elementos HTML.
- Se separa el contenido de la forma en que se presenta.
- CSS 3

# Hojas de estilos CSS

- Hoja de estilos: conjunto de especificaciones que definen el formato de los elementos HTML de una página web.
- Formada por reglas de estilo.
- Se recogen en un fichero .css.

# Hojas de estilos CSS

- Regla de estilos:





# Formas de aplicar estilos

- Estilo local o en línea:
  - Insertar el estilo en cada etiqueta HTML con el atributo *style*.
  - `<p style="color:green">HOLA</p>`

# Formas de aplicar estilos

- Hoja de estilos interna
  - Incrustada en el documento HTML dentro de etiqueta <head>.

```
<head> <meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />
    <title>Textos en dreamweaver</title>
    <style type="text/css">
        h3 {border: medium solid #360; color: #FF0; background-color:
#0C0; }
        p { background-color: #C9F; letter-spacing: 5px; color: #FFF;
word-spacing: 10px; }
    </style>
</head>
```

# Formas de aplicar estilos

- Hoja de estilos externa
  - Almacenada en un archivo distinto al HTML.
  - Forma óptima de aplicar estilos: separa completamente el contenido y el formato.

```
<head>
```

```
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
    <title>Textos en dreamweaver</title>  
    <link rel="stylesheet" type="text/css" href="hojaEstilo.css">
```

```
</head>
```

- Rel: relación entre los ficheros → stylesheet
- Type: tipo de recurso enlazado → text/css

# Selectores

- Selector universal: \*.
  - Las propiedades se aplican a todos los elementos de la página.
  - \* {color : blue}
- Selector de tipo
  - El estilo se aplica a todos los elementos del tipo seleccionado.
  - P {color : blue}

# Selectores

- Selector descendente
  - Selecciona los elementos que se encuentran dentro de otro elemento (entre etiquetas de apertura y cierre).

# Selectores

- Selector descendente

```
p span { color: red; }
```

```
<p>
  ...
  <span>texto1</span>
  ...
  <a href="">...<span>texto2</span></a>
  ...
</p>
```

# Selectores

- Selector de hijos
  - Selecciona un elemento que es hijo directo de otro.

```
p > span { color: blue; }
```

```
<p><span>Texto1</span></p>
```

```
<p><a href="#"><span>Texto2</span></a></p>
```

# Selectores

- Selector adyacente (hermanos)
  - Selecciona elementos que se encuentran justo a continuación de otro.



# Selectores

- Selector adyacente (hermanos)

```
h2 { color: green; }  
h1 + h2 { color: red }
```

```
<body>  
<h1>Titulo1</h1>  
  
<h2>Subtítulo</h2>  
...  
  
<h2>Otro subtítulo</h2>  
...  
</body>
```

# Selectores

- Selector hermanos en general
  - Seleccionan elementos que comparten el mismo padre (hermanos no adyacentes).
  - `h1 ~ pre {color : #00FF00}`
  - `<h1>Título</h1>`
  - `<p>Párrafo</p>`
  - `<pre>Elemento seleccionado</pre>`

# Selectores

- Selector de clase: **class**.
  - Agrupa los elementos por clases para aplicar estilos de forma conjunta.

```
.destacado { color: red; }
```

```
<body>  
  <p class="destacado">Lorem ipsum dolor sit amet...</p>  
  <p>Nunc sed lacus et <a href="#" class="destacado">est adipiscing</a> accumsan...</p>  
  <p>Class aptent taciti <em class="destacado">sociosqu ad</em> litora...</p>  
</body>
```

# Selectores

- Identificadores: **id**.
  - Atributo opcional de los elementos HTML cuya finalidad es identificarlo de forma única.
  - Se usa para aplicar estilos a un único elemento.

```
#destacado { color: red; }
```

```
<p>Primer párrafo</p>
```

```
<p id="destacado">Segundo párrafo</p>
```

```
<p>Tercer párrafo</p>
```

# Selectores

- Pseudoclases.
  - Permiten clasificar los elementos HTML según el estado en el que se encuentren.

Pseudoclase	Descripción
<b>:link</b>	Aplica estilos cuando el enlace no ha sido visitado todavía.
<b>:visited</b>	Aplica estilos cuando el enlace ha sido visitado anteriormente.

Pseudoclase	Descripción
<b>:hover</b>	Aplica estilos cuando pasamos el ratón sobre un elemento.
<b>:active</b>	Aplica estilos cuando estamos pulsando sobre el elemento.

# Selectores

- Selector por atributo
  - Seleccionan elementos HTML en función de sus atributos y valores de esos atributos.

# Selectores

- Selector por atributo
  - [nombre\_atributo]: selecciona los elementos que tienen establecido el atributo nombre\_atributo independientemente de su valor.

```
/* Se muestran de color azul todos los enlaces que tengan  
un atributo "class", independientemente de su valor */  
a[class] { color: blue; }
```

# Selectores

- Selector por atributo
  - `[nombre_atributo=valor]`: selecciona los elementos que tienen establecido el atributo `nombre_atributo` con un valor igual a `valor`.

```
/* Se muestran de color azul todos los enlaces que tengan  
un atributo "class" con el valor "externo" */  
a[class="externo"] { color: blue; }
```



# Selectores

- Selector por atributo
  - [nombre\_atributo~=valor]: selecciona los elementos que tienen establecido el atributo nombre\_atributo y al menos uno de los valores es igual a valor.

# Selectores

- Combinación de selectores
  - Selecciona elementos con un class="especial" que estén dentro de cualquier elemento con un class="aviso".

```
.aviso .especial { ... }
```

# Selectores

- Combinación de selectores
  - Selecciona los enlaces con un atributo id igual a inicio que estén dentro de un elemento tipo `<li>` con un atributo class igual a destacado que forma parte de una lista `<ul>` con un atributo id igual a menuPrincipal.

```
ul#menuPrincipal li.destacado a#inicio { ... }
```

# Selectores

- Unión de selectores
  - Aplicar reglas de estilo a varios elementos.
  - Se separan los selectores por comas.

```
ul, ol {color : #00FF00;}
```

# Elementos <div> y <span>

- Ayudan a estructurar el documento y sirven como selectores en CSS.
- Agrupación con <span>
  - Asignar estilo a una parte del texto.

```
span.azul {color : blue;}
```

```
<p> Frase con <span class="Azul"> color azul </span> </p>
```

# Elementos `<div>` y `<span>`

- Agrupación con `<div>`
  - Sirve para crear secciones (capas o cajas) o agrupar contenido.

# Colores, fondos, textos y fuentes

- Colores

- Palabras clave, RGB hexadecimal, RGB numérico y RGB porcentual.

<b>maroon</b> #800000	<b>red</b> #ff0000	<b>orange</b> #ffa500	<b>yellow</b> #ffff00	<b>olive</b> #808000
<b>purple</b> #800080	<b>fuchsia</b> #ff00ff	<b>white</b> #ffffff	<b>lime</b> #00ff00	<b>green</b> #008000
<b>navy</b> #000080	<b>blue</b> #0000ff	<b>aqua</b> #00ffff	<b>teal</b> #008080	
<b>black</b> #000000	<b>silver</b> #c0c0c0	<b>gray</b> #808080		

# Colores, fondos, textos y fuentes

- Colores

- RGB: se indica la cantidad de rojo, verde y azul.

- Decimal: 0 – 255

```
p { color: rgb(71, 98, 176); }
```

- Porcentual:

```
p { color: rgb(27%, 38%, 69%); }
```

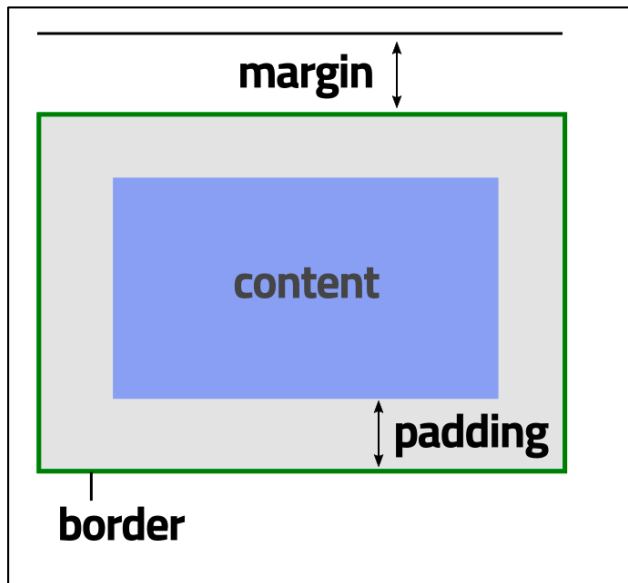
- Hexadecimal

```
p { color: #4762B0; }
```



# Bloques, cajas o capas

- Etiqueta <div>
- Permiten asignar estilos a los elementos agrupados.
- Propiedades de cajas: border, border-top, margin, margin-bottom, padding.



EJERCICIO PÁG. 50

# Tamaño y posicionamiento

- *Position*: establecer posición de elementos HTML.
  - *static*: mismo orden en el que aparecen los elementos.
  - *absolute*: se sitúan en la posición indicada en *top*, *bottom*, *left* y *right* respecto su primer antecesor con posicionamiento distinto a *static*.
  - *fixed*: se sitúan en la posición indicada respecto a la ventana, unos encima de otros.
  - *relative*: posición relativa respecto a la que hay por defecto.

# Lenguajes de script de navegador

- Modificar de forma dinámica el contenido.
- Validar los datos del formulario.
- Obtener efectos gráficos.
- NO manejan BBDD → PHP (script del servidor).

# Introducción a Javascript

- Lenguaje interpretado por el navegador.
- Dos formas:
  - Incluir el código en el propio documento HTML.
  - Almacenar el código en un fichero externo.

```
<html>
  <head>
    <title>Código Javascript en documento externo</title>
    <script type="text/javascript" src="codigo.js"></script>
  </head>
  <body>
    <p>HOLA MUNDO JAVASCRIPT</p>
  </body>
</html>
```

# Sintaxis y características

- Se ignoran los espacios en blanco superfluos.
- Se distinguen mayúsculas y minúsculas.
- No hace falta declarar el tipo de dato de la variable.
- Las sentencias pueden agruparse en bloques delimitados por llaves.
- El ; ayuda a separar las sentencias Javascript.

# Sintaxis y características

- Las sentencias se ejecutan en el orden en que aparecen.

```
<script>  
    document.write (" <h1> Esto es una cabecera H1 </h1>");  
    document.write (" <p> Esto es un párrafo </p>");  
</script>
```

- A veces interesa que la sentencia se ejecute cuando se produzca un determinado evento.

# Ventanas emergentes

- Ventana de alerta: muestra cadena de caracteres y botón de aceptar para cerrar la ventana.

**alert (cadena);**

- Ventana de confirmación: muestra cadena de caracteres y dos botones “Aceptar” y “Cancelar”.

**confirm ( cadena );**

# Ventanas emergentes

- Ventana de entrada: muestra una cadena y una caja de texto inicializada a valor inicial. Devuelve una cadena de caracteres con el contenido introducido por el usuario.

**prompt ( cadena, inicial );**



# Variables y operadores

- Variable: almacenar valor o hacer referencia a otra variable. Pueden locales (var) o globales. Se inicializan.

```
var numero1, numero2 = 1;  
numero1 = 3;
```

- Variables globales: pueden usarse en cualquier función creada en el script.

# Variables y operadores

- Operadores: asignar valor a las variables, realizar operaciones matemáticas y comparar variables.
  - Aritméticos: +, -, \*, /, \, %, ++, --
  - Relacionales: >, <, ==, !=, <=, >=
  - Lógicos: &&, ||, !

# Operador asignación

- Guarda un valor en una variable.

```
var numero1 = 3;
```

# Operadores matemáticos

- Realizar operaciones.

```
var numero1 = 10;
```

```
var numero2 = 5;
```

```
resultado = numero1 / numero2; // resultado = 2
```

```
resultado = 3 + numero1; // resultado = 13
```

```
resultado = numero2 - 4; // resultado = 1
```

```
resultado = numero1 * numero 2; // resultado = 50
```

# Arrays

- Colección de variables.
- EJEMPLO:

```
var dia1 = "Lunes";  
var dia2 = "Martes";  
...  
var dia7 = "Domingo";
```

```
var dias = ["Lunes", "Martes", "Miércoles", "Jueves", "Viernes", "Sábado", "Domingo"];
```

# Arrays

- Para acceder a un elemento del array:

```
var diaSeleccionado = dias[0];    // diaSeleccionado = "Lunes"  
var otroDia = dias[5];           // otroDia = "Sábado"
```

EJEMPLO PÁG. 54

EJERCICIO 4 (pdf)

# Estructuras básicas de control

- Estructuras de control de flujo: alterar el orden de ejecución, tomar decisiones lógicas o repetir un bloque de instrucciones.

# Variables, operadores y estructuras básicas de control

- Estructuras de control de flujo:

- Condicional simple:

**if (condición) { Bloque de instrucciones }**

- Condicional doble:

**if (condición) { Bloque1 de instrucciones }**  
**else {Bloque2 de instrucciones }**



# Variables, operadores y estructuras básicas de control

- Condicional doble:

**if (condición) { Bloque1 de instrucciones}**  
**else {Bloque2 de instrucciones }**

```
var edad = 18;  
  
if(edad >= 18) {  
    alert("Eres mayor de edad");  
}  
else {  
    alert("Todavía eres menor de edad");  
}
```

# Variables, operadores y estructuras básicas de control

- Estructuras de control de flujo:

- Mientras-Hacer

- while (condición) { Bloque de instrucciones}**

- Hacer-Mientras

- do {Bloque de instrucciones} while (condición)**

- Mientras

- for (inicialización; condición, actualización)  
{Bloque de instrucciones}**

# Variables, operadores y estructuras básicas de control

- Estructuras de control de flujo:

- Mientras

**for (inicialización; condición, actualización)**  
**{Bloque de instrucciones}**

```
var mensaje = "Hola, estoy dentro de un bucle";  
  
for(var i = 0; i < 5; i++) {  
    alert(mensaje);  
}
```