

ESCUELA DE SISTEMAS Y TECNOLOGÍAS

Transparencias de ANALISTA DE SISTEMAS
Edición 2020 Materia: Java Web

TEMA: HTML

Agenda (1)

- HTML
- Títulos
- Párrafos
- Entidades HTML
- Imágenes de Contenido
- Hipervínculos
- Listas de Elementos
- Tablas
- Formularios
- Etiquetas

Agenda (2)

- Campos de Texto
- Casillas de Verificación y Botones de Opción
- Listas de Selección
- Campos Ocultos
- Botones
- Campos HTML5
- Estilos CSS
- JavaScript

HTML (1)

- HTML significa HyperText Markup Language (Lenguaje de Marcado de Hipertexto). No es un lenguaje de programación.
- Se utiliza para definir la estructura y semántica de los elementos contenidos en una página web.
- No debe utilizarse para dar formato a los elementos de la página, solo para definir su estructura y darles significado.
- En sus primeras versiones proporcionaba atributos para dar algunos formatos básicos, hoy esta tarea se realiza mediante estilos CSS.
- Su código fuente es interpretado por los navegadores web (IE, Chrome, Firefox, Opera, Safari, etc.).

HTML (2)

- HTML ha evolucionado a lo largo del tiempo, dando lugar a diferentes versiones y variantes.

HTML	1991
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML	2000
HTML5	2014

- Todo documento HTML debe indicar la versión utilizada en un elemento ***DOCTYPE*** al comienzo del mismo. La sintaxis depende de la versión de HTML utilizada. En HTML5 es simplemente ***<!DOCTYPE html>***

HTML (3)

- Un documento HTML se compone de una jerarquía de elementos.
- Estos elementos se representan mediante etiquetas (o tags). Cada etiqueta tiene un nombre que describe lo que ésta representa, y va encerrado entre signos de menor y mayor:

<etiqueta>

- Algunos elementos se delimitan entre una etiqueta de apertura y otra de cierre colocando su contenido entre éstas (en la de cierre, el nombre va precedido de una barra).

<etiqueta>contenido</etiqueta>

- Otros elementos se representan con una única etiqueta con autocierre (se coloca un espacio y una barra luego del nombre):

<etiqueta />

HTML (4)

- Los elementos del documento y por lo tanto las etiquetas se pueden anidar unas dentro de otra (siempre cerrando primero los elementos más internos):

<etiqueta1><etiqueta2>contenido</etiqueta2></etiqueta1>

- Las etiquetas de inicio o autocerradas pueden contener atributos que proporcionan información adicional del elemento.
- Cada atributo tiene un nombre y siempre a continuación mediante un signo de igual (=) se le asigna un valor encerrado entre comillas dobles:

<etiqueta atributo1="valor" atributo2="valor" />

HTML (5)

- El contenido del documento HTML queda delimitado dentro del elemento html (etiquetas **<html>** **</html>**).
- Dentro de este elemento existen dos elementos, correspondientes a dos secciones bien diferenciadas del documento:
 - El elemento head (etiquetas **<head>** **</head>**) contiene información del documento que no será visualizada en la ventana del navegador (título, metadatos, hojas de estilo, scripts, referencias, etc.).
 - El elemento body (etiquetas **<body>** **</body>**) encierra todo el contenido que el usuario podrá ver en la ventana del navegador.

Títulos

- Para marcar un elemento como título se utilizan los elementos h1 ... h6 (etiquetas `<h1>` `</h1>`)
- Los números del 1 al 6 indican la jerarquía del título, no su tamaño.
- El navegador define una hoja de estilos predeterminada que típicamente muestra los títulos más importantes en un tamaño mayor, pero esto se puede cambiar mediante CSS.

Párrafos

- Los párrafos se definen mediante elementos `<p>` `</p>`.
- HTML no tiene en cuenta si se escriben múltiples espacios en el código fuente (se interpretan como uno solo).
- Tampoco tiene en cuenta saltos de línea en el código fuente (al renderizar la página se mostrará el texto en una única línea).
- Se puede indicar un salto de línea con el elemento `
`.
- Mediante la etiqueta `` `` se puede resaltar el texto (darle mayor importancia). El navegador típicamente muestra el texto en negrita pero esto se puede cambiar mediante CSS.
- Mediante la etiqueta `` `` se puede enfatizar el texto.
- Recordar que HTML define estructura y semántica (no formato).

Entidades HTML

- Hay algunos caracteres que no pueden ser escritos directamente en el código fuente HTML, debido a que significan algo para este lenguaje.
- Por lo tanto deben ingresarse mediante códigos especiales denominados entidades HTML.
- Cada entidad HTML comienza con un signo de ampersand (&), sigue con un identificador y finaliza con un punto y coma (;)
- La entidad HTML ` ` permite ingresar espacios múltiples.

Imágenes de Contenido (1)

- Las imágenes de contenido se definen mediante el elemento ****.
- El atributo **src** (obligatorio) permite indicar la url (local o remota) del archivo con la imagen a mostrar.
 - url local: típicamente se coloca una url relativa (imagenes/...)
 - url remota: se coloca una url absoluta (http://...)
- El atributo **alt** permite indicar un texto alternativo que se mostrará en caso de no poder mostrar la imagen (archivo eliminado, imágenes deshabilitadas, navegador en modo texto, etc.).
- Las imágenes ornamentales se colocan como fondo (background-image) de otros elementos mediante CSS.

Imágenes de Contenido (2)

- Los formatos de imagen más utilizados son:
 - **jpg**: para fotografías - Sin transparencia
 - **gif**: para dibujos, ilustraciones, elementos ornamentales - Transparencia completa
 - **png**: para dibujos, ilustraciones, elementos ornamentales, fotografías - Transparencia gradual

Hipervínculos (1)

- Los hipervínculos permiten navegar a otras páginas del mismo sitio, de otro sitio en el mismo u otro servidor, o a otro elemento dentro de la misma página.
- Generalmente se tratan de texto sobre el que se puede hacer clic, pero también pueden tratarse de imágenes.
- El elemento utilizado para crear hipervínculos es el elemento `<a>` ``.
- El atributo **href** permite indicar la url de destino. Puede utilizarse urls relativas o absolutas.
- El atributo **target** permite abrir el hipervínculo en una nueva ventana (o pestaña) si se le pasa el valor “_blank”.

Hipervínculos (2)

- Como se mencionó, los hipervínculos permiten navegar hacia un elemento específico dentro de una página. Para esto previamente debe definirse el atributo **id** del elemento destino con un nombre que lo identifique de forma única.
- Para que un hipervínculo navegue hacia dicho elemento se coloca el **id** del mismo, precedido de un signo de numeral (#) como valor del atributo **href**.
- También es posible agregar el numeral y el id del elemento al final de una **url** para navegar a un elemento determinado de otra página.

Lista de Elementos

- En HTML se puede crear varios tipos de listas de elementos. Las más comunes, sin embargo, son las listas sin orden y las listas con orden.
- Las listas sin orden se crean con el elemento ` `.
- Las listas con orden se crean con el elemento ` `.
- En ambos casos, los elementos de la lista se definen etiquetas ` `.
- Los elementos pueden contener texto, imágenes, etc

Tablas (1)

- Las tablas se utilizan para presentar información estructurada en filas y columnas.
- No deben utilizarse para definir la disposición del diseño de la página. Para esto se utilizan otros elementos de bloque como los div (**<div> </div>**).
- Las tablas se definen mediante el elemento **<table> </table>**
- En su interior se colocan las filas, definidas por elementos **<tr> </tr>**.
- A su vez dentro de cada fila se definen las celdas mediante elementos **<td> </td>**

Tablas (2)

- Se pueden utilizar elementos `<th>` `</th>` para definir celdas de cabecera.
- El atributo **colspan** permite combinar horizontalmente varios elementos *td* en uno solo. Se le pasa la cantidad de columnas que ocupará la celda.
- También existe un atributo **rowspan** para combinar verticalmente filas

Formularios (1)

- Los formularios permiten que los usuarios de una página ingresen información para procesarla con alguna finalidad (generalmente en el servidor).
- El ingreso se realiza a través de campos de tipo texto, casillas de verificación, listas desplegables, etc., mientras que el envío se produce generalmente al pulsar un botón de envío.
- Los formularios se definen mediante el elemento **<form>** **</form>**, y dentro se definen los campos del mismo.
- Atributo **action**: especifica la url del script o programa que procesará los datos enviados. Si no se utiliza este atributo, se asume que la url es la misma que la de la página actual.

Formularios (2)

- Atributo **method**: indica el método de envío de los datos, que puede ser alguno de estos dos:
 1. **get**: Es el método predeterminado. Los datos (parámetros del pedido) viajan al servidor en el “query string”, esto es, agregados a la url del action seguida de un signo de interrogación (?) y a continuación los nombres de los parámetros con sus respectivos valores indicados mediante el signo de igual (=). Cada parámetro se separa del siguiente mediante un signo de ampersand (&).
 2. **post**: los datos viajan al servidor codificados de manera similar (sin la url del action ni el signo de interrogación) pero dentro del cuerpo del pedido, por lo que no se visualizan en la barra de direcciones del navegador.

Etiquetas

- Las etiquetas se utilizan para mostrar los nombres de los campos dentro de un formulario.
- Al clicar sobre el texto de las mismas, el cursor se posiciona en el campo al que corresponden.
- Son útiles sobre todo para los campos de tipo casilla de verificación y botón de opción ya que brindan una superficie mayor para hacer clic.
- Se utiliza el elemento `<label> </label>`.
- El atributo **for** permite especificar el id del campo asociado (éste debe definir su atributo id con un nombre único).

Campos de Texto (1)

- Los campos de texto, así como varios tipos de campos dentro de un formulario se definen mediante el elemento **<input />**.
- Lo que indica que se trate de un campo de texto es el atributo **type** con el valor “*text*”.
- Si se pasa el valor “*password*” al atributo **type**, se mostrará un campo de texto con caracteres enmascarados.
- El atributo **name**, presente en todos los campos de formulario, define el nombre del parámetro del pedido que se enviará al servidor para procesar.
- El atributo **value** (también presente en (casi) todos los campos de formulario) permite indicar el valor pre cargado en el campo, al mostrar la página al usuario.

Campos de Texto (2)

- El atributo **placeholder** permite indicar en los campos de tipo texto, un texto explicativo que se muestra dentro del campo si éste se encuentra vacío (al escribir algo desaparece el texto explicativo).
- Otro tipo de campo de texto que se puede utilizar es el área de texto, que se define mediante el `<textarea> </textarea>`.
- Este elemento no tiene atributo *type*, ni *value*, pero sí **name**.
- El valor pre cargado puede especificarse como contenido del elemento entre sus etiquetas de apertura y cierre.

Casillas de Verificación / Botones de Opción (1)

- Las casillas de verificación (cuadradas) permiten elegir varias opciones.
- Los botones de opción (redondos) permiten elegir una de varias opciones mutuamente excluyentes.
- Ambos tipos de campo se crean mediante el elemento **input**, pero con diferente valor en el atributo **type**.
 - **checkbox**: para casillas de verificación
 - **radio**: para botones de opción
- El atributo **name** indica el nombre del parámetro del pedido que se enviará al servidor.

Casillas de Verificación / Botones de Opción (2)

- En el caso de los botones de opción, se utiliza el mismo **name** para las diferentes opciones a fin de hacerlas mutuamente excluyentes.
- El atributo **value** especifica el valor que se enviará al servidor en caso de encontrarse seleccionada la casilla o el botón de opción.
- Es importante tener en cuenta que en el caso de las casillas de verificación, si una casilla no se selecciona, no se envía el parámetro del pedido al servidor. Lo mismo ocurre si no se selecciona ningún botón de opción dentro de su grupo.
- El atributo **checked** con el valor “checked” deja previamente marcada la casilla o botón de opción al mostrar la página.

Lista de Selección (1)

- Las listas de selección permiten seleccionar opciones predefinidas con texto y valor.
- Pueden ser desplegables (en cuyo caso se puede seleccionar una sola opción) o no (en cuyo caso se puede configurar permitir o no selección múltiple).
- Se utiliza el `<select> </select>` para crearlas.
- El atributo **name** con el nombre del parámetro del pedido a enviar al servidor.
- Si se define el atributo **size** con un valor numérico entero, la lista no será desplegable y mostrará tantos valores como se indiquen.

Lista de Selección (2)

- Si se define el atributo **multiple** con el valor “multiple” se permitirá que el usuario seleccione varias opciones si no se trata de una lista desplegable.
- Dentro del elemento **select** se definen las opciones con elementos **<option> </option>**.
- Para cada elemento **option** se define el atributo **value** con el valor a enviar al servidor.
- El contenido de los elementos **option** será el texto que se mostrará al usuario para cada opción.
- El atributo **selected** con el valor “selected” marca como seleccionado el elemento dentro de la lista.

Campos Ocultos

- Los formularios permiten agregar campos ocultos para enviar valores al servidor que no son ingresados por el usuario (por ejemplo, valores pre cargados por el servidor al mostrar la página).
- Para esto se utiliza el elemento **input** con el valor “**hidden**” en el atributo **type**.
- Como ocurre con los otros tipos de campo, se definen los atributos **name** y **value** con el nombre del parámetro del pedido a enviar al servidor y su valor.

Botones

- El elemento `input` permite definir tres tipos diferentes de botones dentro de un formulario dependiendo del atributo **type**:
 - **submit**: Envía los datos ingresados en el formulario al servidor.
 - **reset**: Limpia el formulario, es decir, lo regresa a su estado inicial.
 - **button**: Botón multipropósito, generalmente para utilizar con javascript.
- Puede utilizarse el atributo **name**, como en cualquier otro campo.
- El atributo **value**, además de enviarse al servidor si se define **name**, indica el texto visible en el botón.

Campos HTML 5

- El elemento `input` permite definir tres tipos diferentes de botones dentro de un formulario dependiendo del atributo **type**:
- HTML5 provee algunos valores nuevos para el atributo **type** del elemento **input** (en caso de que el navegador no lo soporte se muestra como text):

- | | |
|------------------|----------|
| • color | • range |
| • date | • search |
| • datetime-local | • tel |
| • email | • time |
| • month | • url |
| • number | • week |

Estilos CSS (1)

- Los estilos CSS permiten dar formato a los elementos de un documento HTML.
- Cada elemento de un documento HTML tiene propiedades de estilo que se pueden cambiar para darles la apariencia deseada.
- Los estilos CSS se pueden aplicar de varias formas:
 1. Con una hoja de estilos definida en el propio documento HTML.
 2. Con una (o varias) hoja(s) de estilos definida(s) en otro(s) archivo(s) .css.
 3. A través del atributo style presente en todos los elementos HTML.
 4. A través de Javascript.

Estilos CSS (2)

- Para definir una hoja de estilos en el propio documento HTML, se utiliza `<style> </style>` dentro del elemento head del documento.
- Dentro de este elemento se define la hoja de estilos con la sintaxis de CSS.
- La hoja de estilos también se puede definir en otro archivo con extensión .css y referenciarla desde el documento HTML.
- Para referenciar una hoja de estilos externa se utiliza el elemento `<link />` dentro del elemento head del documento.
- Dentro del elemento link se define el atributo **rel** con el valor “**stylesheet**” y el atributo **href** con la **url** del archivo con la hoja de estilos.

Estilos CSS (3)

- Las hojas de estilos consisten en reglas de estilo formadas por:
 - Uno o varios selectores
 - Propiedades de estilo con valor asignado
- El atributo **style** presente en todos los elementos HTML, también permite aplicar propiedades de estilo directamente sobre el elemento (en este caso no se utiliza una hoja de estilos ya que no hace falta selector).
- Cada regla se forma con uno o varios selectores separados por coma (,), y luego entre llaves las distintas propiedades con su respectivo valor asignado por medio de dos puntos (:) y separadas una de otra por un punto y coma (;)

Estilos CSS (4)

- Los selectores pueden ser:
 - Elementos HTML (body, h1, p, li, input, etc.)
 - Clases CSS (aplicadas con el atributo **class** sobre los elementos HTML).
 - **Ids** de elementos dentro del documento HTML.
 - Una combinación de las anteriores (los espacios indican anidamiento).

JavaScript (1)

- JavaScript es un lenguaje de programación interpretado, soportado por la mayoría de los navegadores web.
- Por esta razón es muy utilizado para darle un comportamiento dinámico a los documentos HTML del lado del cliente.
- Entre otras cosas permite ejecutar condicionales, bucles, cambiar estilos, manipular el **DOM** (Document Object Model), etc.
- Para incluir un script JavaScript en un documento HTML se utiliza el elemento `<script> </script>`.
- El elemento script puede incluirse dentro del elemento **head** o dentro del elemento **body** (frecuentemente al final del mismo para asegurar que estén cargados todos los elementos HTML antes de hacer nada con ellos).

JavaScript (2)

- El script JavaScript puede programarse como contenido del elemento script o bien en un archivo aparte con extensión .js, y luego referenciarlo por medio del atributo src de éste elemento.

```
<script src="Javascript.js"></script>
<script>
    function cambiarTitulo() {
        document.body.style.background = 'red';
        document.getElementById('tituloPrincipal')
            .innerHTML = 'Prueba';
    }
</script>
```