

TEMA 2: LENGUAJES DE LA WEB

(HTML5, CSS3)

EI1042 - Tecnologías y Aplicaciones Web

EI1036- Tecnologías Web para los Sistemas de Información (2022/2023)

Professora: Dra. Dolores M^a Llidó Escrivá



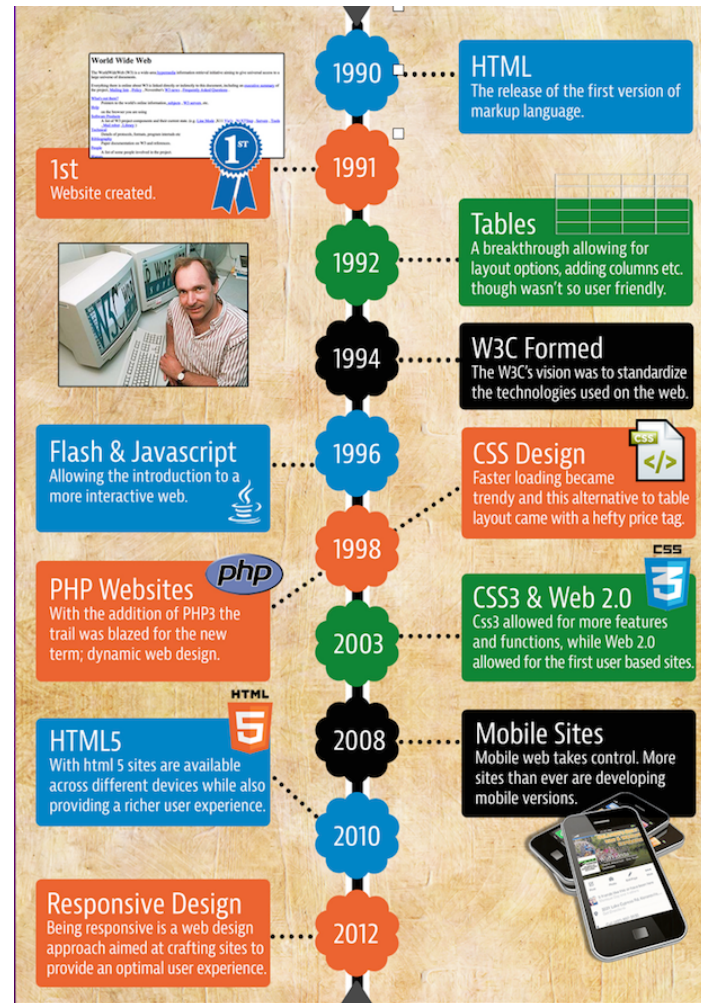
Universitat Jaume I.

ÍNDICE

1. HTML: HyperText Markup Language
2. Renderizado HTML
3. Accesibilidad/usabilidad
4. HTML5
5. Estructura Semántica
6. Formularios
7. CSS: Sintaxis CSS
8. Tipos selectores
9. Medidas/Unidades
10. CSS3
11. Modelo de Cajas: flotantes/flexibles
12. Discriminación por dispositivos.
 - Media-Type
 - Media-Queries
13. Bibliografía

1 HTML: HYPERTEXT MARKUP LANGUAGE

- Lenguaje de marcas para publicar documentos multimedia hipertexto.
- Los documentos HTML son ficheros de texto con marcas de formato que son independientes del SO, lenguaje de programación y programas.
- Las marcas o etiquetas indican cómo representar los contenidos de los documentos.
 - Elemento con contenido: `<html>...</html>`
 - Elemento vacío: `
` o `
</br>`
 - Etiqueta con atributos:
`<body class="principal">...</body>`
- Extensiones de fichero: `.html` o `.htm`
- Cuestiones de estilo: etiquetas en minúsculas y atributos entre comillas dobles (XML).



<https://ontargetwebsolutions.com/insights/wp-content/uploads/2014/11/evolution-of-web-design-ok.jpg>

2. RENDERIZADO HTML

¿Qué es el renderizado(Presentación)?

- Los documentos HTML pueden ser renderizados en pantallas, sintetizadores de habla y dispositivos de Braille.
- Para indicar cómo aparece el renderizado se ha definido el CSS (Cascade Style Sheet).
- Cuando un documento HTML se renderiza se guarda en la memoria como un árbol *DOM* (Document Object Model).

RENDERIZADO BÁSICO EN EL NAVEGADOR: BLOCK/INLINE

Los elementos HTML se dividen en dos grandes tipos:

- `block`: tienden a ocupar todo el espacio disponible a lo ancho y, en caso de existir varios elementos, se sitúan unos debajo de otros (`p`, `aside`, `footer`, `div`, `table`, etc.).
- `inline`: ocupan el espacio necesario dentro de una línea y en caso de existir varios elementos se sitúan uno junto a otro en la misma línea siempre que haya espacio (`span`, `a`, etc.)

<https://developer.mozilla.org/es/docs/Web/HTML/Element>

ELEMENTOS EN LINEA:

- b, big, i, small, tt
- abbr, acronym, cite, code, dfn, em, kbd, strong, samp, time, var
- a, bdo, br, img, map, object, q, script, span, sub, sup
- button, input, label, select, textarea

EJEMPLO KBD

Si está conforme escriba sí. Si no, pulse esc

3 RECOMENDACIONES: ACCESIBILIDAD Y USABILIDAD

- Etiquetas y atributos en minúsculas.
- Organización lógica del documento según estándar: h1, h2, h3, ...
- No usar tablas (<table>) para estructurar la página.
- Intentar que sin el estilo CSS se pueda entender el contenido.
- Resaltar los textos tanto con forma y color distintos.
- No utilizar espacios en blanco consecutivos, mejor usa css :
margin, padding, span
- Uso preferente de etiquetas semánticas header, article....
frente a div

4. HTML 5:

<http://www.w3.org/TR/html5/>

<https://HTML.spec.whatwg.org/multipage/indices.HTML>

- Comienza con la definición del Doctype `<!DOCTYPE html>`. (`<?xml version="1.0" encoding="utf-8"?>` si es valido XML)
- Elementos obligatorios: `html`, `head`, `title` y `body`
- Validadores: <http://validator.w3.org> o <http://validator.w3.org/nu/>
- <http://caniuse.com> ¿Qué etiquetas puedo usar en los navegadores?

ELEMENTOS HTML5 MULTIMEDIA

- Audio: `<audio src=" ... " ></audio>`
- Vídeo: `<video src=" ... "></video>`
- Animación: `<canvas></canvas>` (+ JavaScript)
- Soporte Imágenes vectoriales: SVG (scalable vector graphics)

CUESTIÓN

¿Diferencia href y src?

5. ESTRUCTURA SEMÁNTICA

```
<header> <hgroup>
<h1>Page title</h1>
<h2>Page subtitle</h2>
</hgroup> </header>
<nav>
  <ul>
    <li> Navigation...
  </ul>
</nav>
<section> <article>
  <header><h1>Title</h1> </header>
  Content...
</article>
<article>
  ...
</article>
</section>
<aside>
  Top links...
  <figure> 
  <figcaption>Chart</figcaption> </figure>
</aside>
<footer>
  Copyright © <time datetime="2010-11-08">2010</time>.
</footer>
```

HTML5: ESTRUCTURA SEMÁNTICA

- `<p>`: Mínimo bloque en HTML.
- `<div>`: Une bloques de distintos tipos, cuando no podemos asociar significado semántico a este conjunto.
- `<header>` : Encabezado de la página.
- `<footer>`: Pie de página.
- `<nav>`: Menú de navegación.
- `<hgroup>`: Agrupación titulares (Hn).
- `<article>`: Elementos con significado propio. Puede contener :
 - `<header>`, `<section>`, `<aside>` y `<footer>`.
 - Puede ser un documento entero.
- `<section>`: declaramos que el conjunto de etiquetas tiene un contenido relacionado. Equivalente a una sección de un periódico que engloba varios artículos.
- `<aside>`: bloque que es sólo un añadido a los bloques que tiene al lado (ej.- anuncios). Son datos extra sin lo que podríamos pasar perfectamente pero que hemos decidido añadir en el documento.

6. FORMULARIOS: HTML DINÁMICO

- ¿Qué es una página dinámica?
 - ¿Cómo se crea una página dinámica?
 - ¿Cómo funciona el formulario?
 - ¿alguna alternativa a los formularios?
- Programación cliente:
Javascript+eventos+asíncrono Programación servidor:
Php/java/javascript+BD

ATRIBUTOS FORM

```
<form id="identificador" action="programa" method="post"
      enctype="multipart/form-data">
```

- `method`: `get(defecto)/post`
- `action` (obligatorio): Indica la URL o función JS que procesará la info del formulario.
- `enctype`: El tipo mime empleado para codificar el contenido del form

https://developer.mozilla.org/es/docs/Web/HTTP/Basics_of_HTTP/M

```
<form action="javascript:console.log(this.fname.value);">
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <label for="lname">Last name:</label>
  <input type="text" id="lname" name="lname"><br><br>
  <input type="submit" value="Submit">
</form>
```

ETIQUETAS DE LOS FORMULARIOS

- Input
- Select
- Textarea
- Button

<http://www.w3.org/TR/html5/forms.html>

7.CSS: CASCADE STYLE SHEETS

- HTML utiliza un conjunto de etiquetas predefinidas, cuyo significado es bien conocido. Un navegador sabe cómo debe mostrar estos elementos al usuario.
`h2 {color:blue;}`
- Las CSS (Cascade Style Sheets): son un conjunto de distintas reglas de estilo unidas a un documento que permiten definir estilos para mostrar a los elementos de HTML

[<https://www.w3c.es/Divulgacion/GuiasReferencia/CSS21>]

CSS: EN NAVEGADORES

¿Por qué no son iguales los documentos HTML en todos los navegadores?

- Los navegadores tienen su propio conjunto de reglas de estilo.
- Los navegadores no soportan el estándar CSS 100%

CSS: VENTAJAS

- Separar el contenido del formato.
- Reducir el tiempo de descarga.
- Dar más control sobre el renderizado de HTML.
- Utilizar un mismo estilo para todo el portal web para que tenga una apariencia uniforme y consistente.
- Cada usuario pueden aplicar sus propias reglas de estilo.
- Las reglas se aplican de forma jerárquica o en cascada.
- Se combinan reglas del navegador con las del diseñador

CSS: SINTAXIS

Una regla de estilo se compone de:

- Selector: enlace entre el documento HTML y el estilo
- Declaración: describe el efecto de la regla

```
selector1, selector2 {prop1:value1; prop2:value2;...}
```

```
h1 {color:red}  
h1,h2,h3 {color:blue; text-align:center}
```

¿Prioridad de las reglas?

- Orden Declaración
- Especificidad del selector

[Ver Funcionamiento CSS Navegador](#)

¿CÓMO SE ENLAZA CSS CON HTML?

- Dentro de la definición de etiquetas en el HTML (No abusar)

```
<b style="font-size:14pt;color:red">...</b>
```

- En HTML dentro de la cabecera: etiqueta <style>

```
<style type="text/css">  
  body { color: purple; background-color: #d8da3d}  
</style>
```

- En ficheros externos .css e importarlos:<link>/@import

```
<link rel="stylesheet"  
type="text/css"  
href="splash.css">
```

```
<style type="text/css">  
  @import url(externo.css);  
  h1 {color: gray;}  
</style>
```

8. TIPOS DE SELECTORES

```
<h1>Directorio1</h1>
<div id="directorio">
  <h2>Directorio1.2</h2>
  <h2>Directorio1.3</h2>
  <p class="cita">En en lugar de la mancha...</p>
  <aside class="cita"> Referencia Miguel </aside>
</div>
```

- Etiquetas:

```
h1 {color:blue}
```

- Identificador de elemento (atributo `id`):

```
#directorio {  
position: absolute;  
left: 0px;  
top: 70px;  
background: red;  
width: 20%;  
}
```

- Clases de elementos (atributo `class`):

```
.cita {font-weight: bold; color: red;}  
p.cita {background-color: red;}
```

TIPOS DE SELECTORES

- Pseudo-Clases :
 - :link cualquier enlace no visitado de la página.
 - :visited cualquier enlace ya visitado de la página.
 - :active enlace en el que estamos en este momento.

```
a:visited {color: gray;}
```

- Pseudo-Elementos

```
p:first-letter {color: red;}  
p:first-line {color: gray;}
```

- En función del contexto

```
td b {color: green}  
ul b {color: #074A}
```

El atributo Color del CSS soporta nombre inglés o HEXA, RGB y HSL

9. UNIDADES CSS

- Medidas relativas:
 - `em` (ems, "font-size" o tamaño de la fuente 12px)
 - `ex` (x-height, altura de la letra "x" minúscula de la fuente actual)
 - `px` (pixels, relativa a la resolución de pantalla)
- Medidas absolutas:
 - `in` (pulgadas; 1in=2.54cm)
 - `cm` (centímetros; 1cm=10mm)
 - `mm` (milímetros)
 - `pt` (puntos tipográficos; 1pt=1/72in)
 - `pc` (picas; 1pc=12pt)
- Unidades de medida relativas:
 - Porcentajes : `margin-left:20%` margen izquierdo de un 20% del ancho del elemento contenedor
 - Proporcionales: `margin-left:3*` margen izquierdo de $\frac{3}{4}$ partes del elemento contenedor

TAMAÑO DE LA FUENTE: FONT-SIZE

- Absolutos: [xx-small | x-small | small | medium | large | x-large | xx-large]
- Relativos: [larger | smaller]

EJERCICIO 1

Convierte un elemento "span" en un botón con una clase enlaceBoton.
Usando las reglas de background y border.

- Buscando en estas guías
<https://www.w3.org/community/webed/wiki/CSS>
- Programando en <https://codepen.io/martinwolf/pen/qlFdp>

¿Que pasa si un elemento "a" le pones también la clase enlaceBoton.

EJERCICIO 2

- Crea un pequeño formulario para registrarse en un gimnasio que utilice:
 - campos *nombre* y *DNI* con `input type="text"`
 - campo *sexo* con `input type="radio"`
 - campo *fecha* con un `input type="date"`
 - campo *deportes* con `select multiple`.

Trata de encontrar una solución consultando la guía de HTML y CSS poner más atributos a estos campos y con el mejorar la apariencia del formulario. Por ejemplo una imagen de fondo al formulario, los textos con el tamaño requerido y con una fuente de distintos colores... Además usa para los campos obligatorios el atributo "required"

¿Que parámetros se envían al servidor?

10. CSS3

- Transformaciones
- Transiciones
- Animaciones
- Fuentes propias
- Columnas
- Transparencia
- Degradados
- Bordes redondeados

EJEMPLOS CSS3

```
<style>
  #gira:hover {
    transform:rotate(-30deg); }
</style>

<p id="gira" >Esto hace algo
<a href="#"> Acción CSS3Transform</a>Se ve o no se ve.</p>
```

Esto hace algo [Acción CSS3Transform](#) Se ve o no se ve.

Más ejemplos: <http://estelle.github.io/animation/30.html#slide1>

11. MODELO DE CAJAS

- Los elementos HTML son de 2 tipos: block e inline.
- La propiedad `display` altera el tipo de caja de un elemento.
 - `inline`: el elemento no es un bloque, se renderiza en línea con otros elementos de ese tipo.
 - `block`: el elemento se comporta como un bloque, podemos poner márgenes y bordes..
 - `none`: el elemento no se muestra; el efecto es como si no existiera, por lo que su espacio será ocupado por otros.
 - `inherit`: se heredan las características del elemento padre.
 - `flex`: el elemento es flexible en posición horizontal y vertical.
 - `inline-block`: el elemento de bloque pero que se renderiza en línea con otros elementos.

EJEMPLO

```
<p> El <span style="background:#ffeebb;"> SPAN </span> por de  
<span href="aqui" style="display:block;background:#ffeebb;">  
Span que ya no es inline, es block.</span></p>  
<p style="display:inline-block;background:#ffeebb;" >  
Ahora este párrafo no es block</p>  
<p style="background:#ffeebb;">Y no ocupa toda la extension d
```

El SPAN por defecto es inline.

Span que ya no es inline, es block.

Ahora este párrafo no es block

Y no ocupa toda la extension como este.

PROPIEDADES DE LAS CAJAS

```
header, footer {  
  display: block;  
  margin: 4px;  
  padding: 5px;  
  min-height: 100px;  
  border: 1px solid #eebb55;  
  border-radius: 7pt;  
  background: #ffeebb;  
}
```



PROPIEDAD OVERFLOW

Propiedad que especifica que hacer con el contenido excedente en un elemento a nivel de bloque

`overflow: scroll`

- `visible` (por defecto): Contenido no recortado, podría ser dibujado fuera de la caja contenedora.
- `hidden`: Contenido recortado.
- `scroll`: Contenido recortado y el Navegador, usa barras de desplazamiento.
- `auto`: Depende del Navegador.

ELIPSIS TEXT

```
<div id="container">  
  <p class="truncate" >This is a text that should be truncated
```

```
.truncate {  
  white-space: nowrap;  
  overflow: hidden;  
  text-overflow: ellipsis;  
}  
  
.truncate:hover {  
  white-space: normal;  
  overflow: visible;  
}
```

MODELO DE CAJAS: Position

dos

- **Static**: Posición predeterminada en HTML.
- **Relative**: La posición de la caja se ajusta en relación a su posición normal dentro de la página. La caja siguiente se sitúa como si esta no se hubiera desplazado.
- **Absolute**: Las cajas son quitadas del flujo normal, su posición se especifica con las propiedades 'left', 'right', 'top', y 'bottom'. No tienen ninguna influencia sobre la posición de las cajas siguientes.
- **Fixed**: Como el anterior, pero se posiciona en función de la ventana del navegador. Esta caja no se mueve de posición al hacer scroll.

```
#dos{
display:block;
position: absolute;
top: 15px;
padding: 5px 5px 5px 5px;
margin-left: 0px; /*poner izquierda*/
border-style: solid;
}
<div id="uno">uno</div>
<div id="dos">dos</div>
```

uno

CAJAS FLOTANTES: Float

- **Left** : Genera un bloque a la izquierda donde posicionar las siguientes cajas si caben. De modo que el contenido de la caja se posiciona a la izquierda arriba de ese bloque.
- **Right** : Lo mismo a la izquierda
- **None** : No deja elementos flotantes.

```
<style>
#dos0 {float:right; border-style:solid;}
#uno0 {float:left; border-style:solid;}
</style>
<div id="uno0">uno1</div>
<div id="dos0">dos1</div>
```

Si al lado ponemos un párrafo este se alinea a la derecha de la imagen.

uno1

dos1

CAJAS FLEXIBLES: DISEÑO FLEXIBLE

Un diseño flexible permite para alterar el ancho y alto de sus elementos para ajustarse lo mejor posible al espacio disponible en cualquier dispositivo. Un contenedor flexible expande sus elementos para rellenar el espacio libre, o los comprime para evitar que rebasen el área prevista.

- Un contenedor flexible (Flex container) se define usando los valores flex o inline-flex en la propiedad display del contenedor o elemento "padre" que contiene los elementos flexibles.
- Cada hijo de un contenedor flex se convierte en un elemento flexible (Flex item), que permite cajas a su derecha o izquierda.

<https://developer.mozilla.org/es/docs/Web/CSS/flex>

COLUMNS

```
<style>
  .content-box {
    columns: 3 auto;
  }
</style>
<p>Usando Columns</p>
<div class="content-box">
  <div>uno</div>
  <div>due</div>
  <div>tre</div>
</div>
```

Usando Columns

uno

due

tre

Ejemplo

12. CSS: DISCRIMINACIÓN POR DISPOSITIVOS

MEDIA-TYPE

- Permite categorizar los dispositivos de usuario
- Se usaron originalmente para seleccionar en HTML4 distintos CSS en función del dispositivo del usuario.
- Pero son insuficientes como una manera de discriminar entre dispositivos con diferentes
- *Media Types*: all, braille, embossed, handheld, print, projection, screen, speech, tty, tv.

```
<link rel="stylesheet" type="text/css"
href="st1.css" media="print">

<link rel="stylesheet" type="text/css"
href="st2.css" media="screen">
```

CSS3: MEDIA QUERIES

- Las consultas de medios (media queries) son simples filtros que pueden aplicarse a los estilos CSS.
- Facilitan el cambio de estilos según el tipo de dispositivo (media-type) o las características del dispositivo (media-feature), como el tipo de pantalla, el ancho, el alto, la orientación e incluso la resolución.

```
@media screen and (min-width:600px) {  
  nav {  
    float: left;  
    width: 25%;  
  }  
  section {  
    margin-left: 25%;  
  }  
}
```

```
@media (query)  
{  
  //CSS Rules used when __query matches  
}
```

EJEMPLO MEDIA-QUERIES

```
<link rel="stylesheet" media="(max-width: 640px)"
href="max-640px.css">
<link rel="stylesheet" media="(min-width: 640px)"
href="min-640px.css">
<link rel="stylesheet" media="(orientation: portrait)"
href="portrait.css">
<link rel="stylesheet" media="(orientation: landscape)"
href="landscape.css">
<style>
  @media (min-width: 500px) and (max-width: 600px) {
    h1 { color: fuchsia;}
    .desc:after {
      content: "In fact, it's between 500px and 600px wide.";
    }
  }
</style>
```

MEDIA FEATURE

- `min-width` : Reglas aplicadas a cualquier ancho de ventana de navegador que supere el valor definido en la consulta.
- `max-width` : Reglas aplicadas a cualquier ancho de ventana de navegador que sea inferior al valor definido en la consulta.
- `min-height` : Reglas aplicadas a cualquier alto de ventana de navegador que supere el valor definido en la consulta.
- `max-height` : Reglas aplicadas a cualquier alto de ventana de navegador que sea inferior al valor definido en la consulta.
- `orientation=portrait` : Reglas aplicadas a cualquier navegador cuyo alto de ventana sea superior o igual al ancho.
- `orientation=landscape` : Reglas aplicadas a cualquier navegador cuyo ancho sea superior al alto.

LECTURAS RECOMENDADAS

- Webgrafía:
 - Guía HTML y CSS : <https://developer.mozilla.org/en-US/docs/Web>
 - Guía referencia HTML
<https://html.spec.whatwg.org/multipage/indices.html>
 - Guía general CSS:
<http://www.w3.org/community/webed/wiki/CSS>
 - Guia de etiquetas y reglas css activas en los navegadores:
<http://caniuse.com>
 - Guía Accesibilidad:
<http://w3c.es/Divulgacion/GuiasBreves/Accesibilidad>
- Bibliografía:
 - Head first web design. O'reilly. Ethan Watrall & Jeff Siarto.
 - HTML5 y CSS3. Editor: ENI (Septiembre 2011) Autor : Luc VAN LANCKER ISBN : 978-2-7460-6816-2

¿DUDAS?



