

SPA 322

Susanna Allés Torrent

2.3. El lenguaje de la web (HTML / CSS)

El lenguaje de marcado con el que se crean las páginas web es el HTML HyperText Markup Language. Se llama hipertexto porque es simplemente un texto que contiene links. Cuando abris vuestro navegador, en realidad estáis viendo el resultado del código en el que está escrito la página. Las secciones, los títulos, los menús, los colores no existen como tal, como en un documento impreso, sino que es código.

El HTML es lo que determina la estructura de la página, mientras que el diseño se delega a otro lenguaje: CSS que corresponde a Cascading Style Sheet.

Para empezar podemos crear una página de prueba. Podéis hacerlo de la siguiente manera:

- Abrid el terminal
- Cread una nueva carpeta llamada, por ejemplo, HTML: `mkdir HTML`
- Cread un nuevo documento html: `touch prueba1.html`

Para manipular el fichero HTML necesitaremos un editor de texto, os podéis bajar Bbedit o Notepad++, por ejemplo.

Una página web estática es un conjunto de archivos HTML que están ubicados en un servidor. Por lo general se estructura con carpetas diferentes, una para cada tipo de archivo y/o sección. Por ejemplo,

Imaginemos que tenemos una página que se llama “cursos” y que en su interior tenemos una página de bienvenida (`index.html`, que abre el navegador por defecto), y tres diferentes links que nos llevan a tres carpetas diferentes.

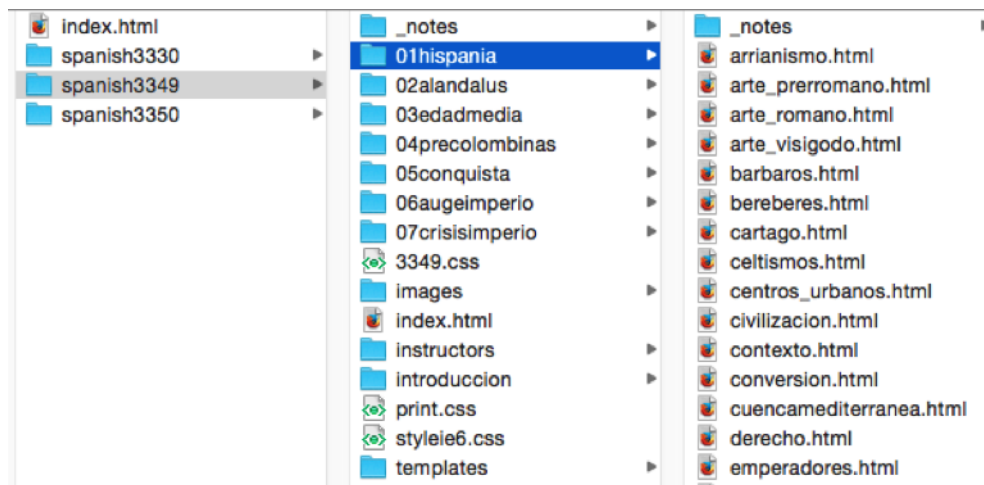


Figure 1: Estructura página HTML

Imaginemos ahora que navegando llegamos a la página `emperadores.html`; en el navegador aparecerá:

</cursos/spanish3349/01hispania/emperadores.html>

Figure 2: Estructura página HTML

El funcionamiento es exactamente el mismo que cuando navegamos por nuestro Finder, de manera que si queremos modificar una página en concreto, deberemos abrir y manipular ese archivo, en este caso `emperadores.html`.

HTML

El lenguaje HTML tiene su propia sintaxis, donde el elemento principal son las etiquetas (tags) que se componen de una etiqueta de inicio `<>` (opening tag) y una etiqueta de cierre `</>` (closing tag). Las etiquetas se anidan unas dentro de las otras, como si fueran un árbol genealógico. Recordad: todo lo que se abre se cierra.

```
<etiqueta1>
  <etiqueta2>
    <etiqueta3>Esto es el contenido</etiqueta3>
```

```
</etiqueta2>
</etiqueta1>
```

Hay algunos conceptos preliminares a retener:

- Una etiqueta se compone del elemento (o nombre), y a veces de un atributo que conlleva obligatoriamente un valor:

```
<elemento atributo="valor">contenido</elemento>
```

- A veces una etiqueta puede estar vacía, es decir, no tener contenido y se señala así: `<etiqueta/>`. En HTML es el caso, por ejemplo, de los saltos de línea `
` o de las imágenes ``.

El documento HTML se compone de diversas secciones obligatorias:

- `<!DOCTYPE html>` Indica al navegador que el tipo de documento es html
- `<html>` es la etiqueta que englobará todo el contenido, tanto el `<head>` como el
- `<head>` comprende todas las informaciones del fichero que no aparecerán en el navegador
- `<title>` es el título que aparece en la pestaña del navegador
- `<body>` es propiamente el cuerpo de la página, lo que sí veremos, por tanto, en el navegador.

La estructura de cualquier documento es la siguiente:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Título de mi página</title>
  </head>
  <body>
  </body>
</html>
```

Estructuras y elementos HTML básicos:

- `<p>` se utiliza para crear párrafos
- `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>` sirven para señalar los títulos, de mayor a menor

- `el link` La etiqueta `<a>` sirve para crear todo tipo de enlaces
- `` para inserir imágenes al interno de la página. Este elemento debe contener un atributo `scr` que debe contener la localización del fichero de la imagen. Si quisiéramos convertir nuestra imagen en un link, no tendríamos más que anidar la imagen al interno de una etiqueta `<a>`: ``

Para crear listas:

- `` ordered lists, creará una lista numérica; en su interior se anidaran los ítems
- `` unordered list, creará una lista; en su interior se anidaran los ítems
- `` list item

Las listas se pueden anidar unas dentro de las otras, por ejemplo:

```
<ol>
    <li>item</li>
    <ul>
        <li>subitem</li>
    </ul>
    <li>item</li>
    <li>item</li>
</ol>
```

Comentar el código es siempre muy importante, para ello podemos utilizar una etiqueta especial que no nos aparecerá en el navegador: `<!-- comentario -->`

Typografías simples

- `` - negrita
- `` - cursiva
- `<i>` - cursiva

Tablas

`<table>` `<tr>` (table row) corresponde a las filas `<td>` (table data) corresponde a las columnas

```
<table border="1px">
  <tr>
    <td>Fila 1: Columna 1</td>
    <td>Fila 1: Columna 2</td>
  </tr>

  <tr>
    <td>Fila 2: Columna 1</td>
    <td>Fila 2: Columna 2</td>

  </tr>

  <tr>
    <td>Fila 3: Columna 1</td>
    <td>Fila 3: Columna 2</td>

  </tr>
</table>
```

Opcionalmente podemos utilizar:

`<thead>` Para el título, que contiene `<tr>` y `<th>` `<tbody>` Para los datos de la tabla

```
<table border="1px">
  <thead>
    <tr>
      <th>Título columna 1</th>
      <th>Título columna 2</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Fila 1: Columna 1</td>
      <td>Fila 1: Columna 2</td>
    </tr>
```

```

        <tr>
            <td>Fila 2: Columna 1</td>
            <td>Fila 2: Columna 2</td>

        </tr>

        <tr>
            <td>Fila 3: Columna 1</td>
            <td>Fila 3: Columna 2</td>

        </tr>
    </tbody>
</table>

```

Si queremos combinar celdas usaremos **colspan** seguido del signo = y entre comillas el número de celdas a combinar:

```

<thead>
    <tr>
        <th colspan="2">Título columna 1</th>
    </tr>
</thead>

```

Divisiones

Las divisiones es uno de los elementos más importantes y se utiliza para crear divisiones en la página: **<div>**

Podéis realizar una primera prueba añadiendo ancho y alto de la división, y también color:

```

<div style="width:50px; height:50px; background-color:red"></div>

<div style="width:50px; height:50px; background-color:blue"></div>

<div style="width:50px; height:50px; background-color:green"></div>

```

Span

- `` Para dar estilo a pequeñas partes de la página (por ej. palabras o grupo de palabras).

Algunos tutoriales útiles:

- Codecademy: HTML & CSS
- W3Schools

CSS: Estilo

Para dar estilo a una página podemos proceder de tres maneras distintas:

I. Insertar la orden de diseño al interno de las etiquetas. Ej. `<p style="font-size:12px">`

II. Añadir dentro del `<head>` el código CSS, de la siguiente manera:

```
<head>
  <title>Título de la página</title>
  <style type= "text/css">
    (...)
  </style>
</head>
```

III. Crear un fichero diferente que contenga todas las instrucciones CSS: `fichero.css` que se señalará en el `<head>` del documento HTML:

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

La opción más utilizada es la III, pues si creamos, pongamos por caso, un sitio web estático no deberemos cambiar las órdenes cada vez que creamos una página o fichero HTML, sino que todas las instrucciones estarán en un fichero separado y lo deberemos cambiar una sola vez.

La sintaxis de las CSS es sencillo pero puede llegar a complicarse bastante. La estructura básica de una regla CSS es la siguiente:

```
selector{
propiedad: valor;
}
```

El selector puede ser cualquier etiqueta o elemento HTML o bien cualquier clase o identificador. La propiedad es la característica que quiere modificarse, como el color de fondo o de la fuente, el tamaño y familia de la fuente, etc. Mientras que el valor es propiamente el aspecto particular que queremos dar a ese o esos elementos HTML.

Los comentarios al interno de las CSS se señalan así: `/* Comentario */`

Veamos qué comandos son los más básicos:

Fuentes de letras:

`font-size: Xpx;`

- Familias de fuente (Arial, Verdana, Impact, ...): `font-family: Arial;`

Hay diferentes tipos de letras: serif, sans-serif, cursive... cada ordenador tiene las suyas. Normalmente se opta por ofrecer diversas fuentes, en caso que la máquina no tenga la primera, pasa a la segunda:

`font-family: Tahoma, Verdana, sans-serif;`

Las medidas pueden darse en píxeles `px` (ej. `font-size:15px;`) y establecer un tamaño fijo, o bien en `em` que otorga un tamaño relativo a la pantalla: `1em` corresponde al tamaño por defecto, `0.5em` a la mitad del tamaño, y `2em` al doble.

Colores:

`color: green;`

Los colores pueden expresarse de diferentes formas: - colores sencillos con sus nombres: `black`, `pink`, `violet`, `green`, `yellow` - con valores hexadecimales (`#` y 6 dígitos). Vid. HTML Color Picker. `background-color: red;`

Justificación del texto:

`text-align:center;` center, right, left

Tamaños:

height: la altura se expresa en píxeles **width:** el ancho también se expresa en píxeles

Es útil especialmente para las div:

```
div{
    background-color:#cc0000;
    height:100px;
    width:100px;
}
```

Decoración del texto:

Podemos dar un tipo especial a determinados elementos. Por ejemplo, en el caso de los links, `<a>`, podemos señalar de qué color lo queremos y si debe ir subrayado o no:

```
a{
    text-decoration: none;
}
```

Bordes:

Especialmente en las tablas, aunque también en los elementos `<div>`, es útil utilizar los bordes. Por ejemplo, si tenemos una tabla:

```
selector{
    border: 1px solid black;
}
```

border-radius: para redondear los bordes

Las reglas CSS pueden y deben darse de manera general, sin necesidad de añadir el atributo `style` a cada etiqueta. Para ello utilizaremos los atributos `@class` y `@id`. - `@class` sirve para dar estilo a una clase de elementos que comparten una serie de características. - `@id` sirve para dar un identificador único y por tanto otorgar una serie de características propias.

Por ejemplo:

`<div class="texto">` suponemos que hay más de una `<div>` que queremos con las mismas características. En la CSS deberíamos señalarlo así, por ejemplo:

```
.texto{
    font-size:15px;
    font-family:Arial;
    color:blue;
}
```

Mientras que una `<div id= "contenedor">` señala que es la única en todo el documento que habrá unas características concretas. En la CSS se reflejaría con el símbolo `#`:

```
#contenedor{
    background-color: yellow;
}
```

Pseudo-selectores

Muy utilizado cuando queremos modificar parcialmente el comportamiento de un elemento, como en los links. Hay tres tipos diferentes:

- `a:link`: link que no se ha todavía consultado
- `a:visited`: link consultado
- `a:hover`: cuando pasamos el cursor por encima.

```
selector:pseudo-class-selector{
    propiedad:valor}
```

```
a:hover {
    color: #cc0000;
    font-weight: bold;
    text-decoration: none;
}
```

Posición de los elementos `<div>`

- Apariencia:
 - `display` puede contener 5 valores:

- `block`: los sitúa en bloque
- `inline-block`: los sitúa en bloque sin sus medidas originales
- `inline`: los sitúa en bloque sin sus medidas originales
- `none`: desaparecen
- Margen:
 - `margin`: espacio que hay fuera del elemento
 - `margin-top`: /*some value*/
 - `margin-right`: /*some value*/
 - `margin-bottom`: /*some value*/
 - `margin-left`: /*some-value*/

Pueden marcarse todos a la vez empezando por el de arriba: `margin: 1px 2px 3px 4px;`

- `border`: el borde es lo que delimita el elemento propiamente dicho
- `padding`: es el espacio entre el contenido y el borde
- Floating:
 - `float:right`
 - `float:left;`
- Position: Hay cuatro tipos de posición:
 - `absolute`: se posiciona en relación al primer elemento padre que no tiene `position:static`; si no hay tal elemento, el elemento con posición absoluta se coloca en relación al elemento `<html>`.
 - `static`: La posición por defecto es `static`.
 - `relative`: es la más utilizada
 - `fixed`: para fijar un elemento en la pantalla (los menús por ejemplo)

Actualmente existen diferentes opciones para construir un sitio web estático. Bootstrap y Foundation son dos de ellos, así como Jekyll.