



JUAN G. GUTIÉRREZ
HTML5, CSS3 Y JS
PRIMERA EDICIÓN

Contents

Tutorial de introducción a HTML5, CSS3 y al pensamiento computacional con JavaScript	2
Introducción	2
Cómo usar el curso	2
Índice de contenido	2
Objetivo	3
Semana 1: HTML5 - Sintaxis básica y tablas	3
¿Cómo funciona Internet?	3
Lenguajes de marcas	4
Primer ejemplo	5
Etiquetas más comunes de HTML	5
Las nuevas etiquetas de HTML5	5
Estructura básica	5
Títulos y párrafos	7
Formato de palabras	8
Imágenes	9
Uso de audio y video en HTML5	9
Formularios en HTML5	9
Listas	11
Plantilla de una página Web con hiperenlaces	12
Editores de texto	12
Semana 2: HTML5 - El modelo de cajas	12
La etiqueta <div>	14
Altura y anchura de una caja	16
Margen	16
Borde	16
Laterales	17
Más opciones para los bordes	17
Esquinas redondeadas	18
Sombras	18
Elementos flotantes	18
Ejercicio semana 2	19
Semana 3: CSS3 - Con un poco de estilo. Menús	20
Ejercicio horario de clase	20
Semana 4: JavaScript - ¡Hola Mundo!	20
Semana 5: JavaScript - Variables, clases, métodos y funciones	20
Ejercicio: Adivina en qué número pienso	20
Semana 6: JavaScript - El árbol DOM	20

Semana 7: JavaScript - Pensamiento computacional. Algoritmos	20
Ejercicio: La letra del DNI	20
Semana 8: JavaScript - Almacenando datos	20

Tutorial de introducción a HTML5, CSS3 y al pensamiento computacional con JavaScript

Introducción

Este curso ha sido realizado para ayudar a nuestros alumnos durante los meses de confinamiento por la pandemia del COVID-19. Primavera del 2020.

Este obra está bajo una licencia de Creative Commons Reconocimiento-CompartirIgual 4.0 Internacional.

Para seguir el curso te animamos a visitar el canal de Youtube del autor además de seguir estas explicaciones.

Si estás viendo una copia en formato PDF o ePub del libro, recuerda que siempre puedes acceder a la última versión en el repositorio de Github.

Cómo usar el curso

Para cada entrega (semana) vamos creando una rama nueva, de tal manera que si quieres seguir el proyecto viendo los vídeos, sólo necesitarás cambiar a la rama de la semana del vídeo.

Índice de contenido

1. Semana 1: HTML5 - Sintaxis básica y tablas.
 - Ejercicio: Plantilla de una página Web con hiperenlaces.
2. Semana 2: HTML5 - El modelo de cajas.
 - Ejercicio: Esqueleto de una página Web.
3. Semana 3: CSS3 - Con un poco de estilo. Menús.
 - Ejercicio horario de clase.
4. Semana 4: JavaScript - ¡Hola Mundo!
 - Ejercicio: mostrar por consola las tablas de multiplicar.
5. Semana 5: JavaScript - Variables, clases, métodos y funciones.
 - Ejercicio: Adivina en qué número pienso.
6. Semana 6: JavaScript - El árbol DOM.
 - Ejercicio: Modificar el contenido de nuestra página al pulsar botón.
7. Semana 7: JavaScript - Pensamiento computacional. Algoritmos.

- Ejercicio: La letra del DNI.
8. Semana 8: JavaScript - Almacenando datos.
- Ejercicio: Guardando información en LocalStorage.

Objetivo

Al terminar el curso, habrás aprendido cómo funciona HTML5, CSS3 y algo de JavaScript y habrás generado un juego como éste:

Semana 1: HTML5 - Sintaxis básica y tablas

¿Cómo funciona Internet?

Cuando queremos visitar una página Web normalmente nos vamos a nuestro buscador favorito y hacemos la consulta. Pero si nos fijamos en la barra de direcciones, cuando estamos en esa Web, ahí hay escrito lo que llamamos un **identificador uniforme de recurso o URL** (Uniform Resource Locator), es decir una dirección que nos permite acceder de forma inequívoca a un recurso de un determinado servidor.

El formato general de un URL es: **protocolo://máquina.directorio.archivo** aunque también pueden añadirse otros datos: **protocolo://usuario:contraseña@máquina:puerto.directorio**

Por ejemplo: https://es.wikipedia.org/wiki/Localizador_de_recursos_uniforme:

- **https** es el protocolo
- **es** es el servidor para España de Wikipedia
- **wikipedia.org** es el dominio donde está la página
- **/wiki/Localizador_de_recursos_uniforme** es el archivo que estamos consultando

Primer ejemplo

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    Aquí va el texto que vemos en el navegador.
  </body>
</html>
```

Etiquetas más comunes de HTML

Las nuevas etiquetas de HTML5

Como decíamos con anterioridad, HTML5 plantea un nuevo esquema y secciones de un documento: <section>, <article>, <nav>, <header>, <footer>, <aside>.

Estructura básica

Sea el siguiente ejemplo adaptación de los tutoriales de Mozilla:

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <nav>

    </nav>
    <section>
      <h1>El fiero conejo</h1>
    <section>
      <h1>Introducción</h1>
      <p>En esta sección presentamos al conocido mamífero.
    </section>
    <section>
      <h1>Hábitat</h1>
      <p>El conejo, como fiero depredador, necesita un entorno con abundantes zorros c
    </section>
    <aside>
      <p>otros estudiosos del conejo
```

```

    </aside>
  </section>
</body>
</html>

```

En el anterior ejemplo tenemos que:

- La etiqueta `<!DOCTYPE html>` indica que se trata de un documento HTML.
- La etiqueta `<html lang="es">` y su pareja que cierra al final `</html>` sirven para indicar que dentro está el documento HTML. Fíjate en el atributo **lang="es"**, sirve para indicar que el documento está en español. Si lo cambiamos por **lang="en"**, ¿qué idioma crees que sería el del documento? Correcto, inglés (**E**nglish).
- `<header>` y `</header>`: sirven para definir un bloque de contenido que hará las veces de título de la página web.
- `<footer>` y `</footer>`: define el pie de página de nuestra web.
- `<nav>` y `</nav>`: donde incluiremos diferentes enlaces para que el usuario pueda desplazarse entre las partes de nuestro sitio web.
- `<section>` y `</section>`: para definir grandes secciones de nuestra página.
- `<article>` y `</article>`: marca los límites de un contenido específico, como una entrada de un blog o un artículo en general.
- `<aside>` y `</aside>`: se emplea para definir un contenido que está relacionado con la página, pero que se debe considerar como separado del contenido principal.

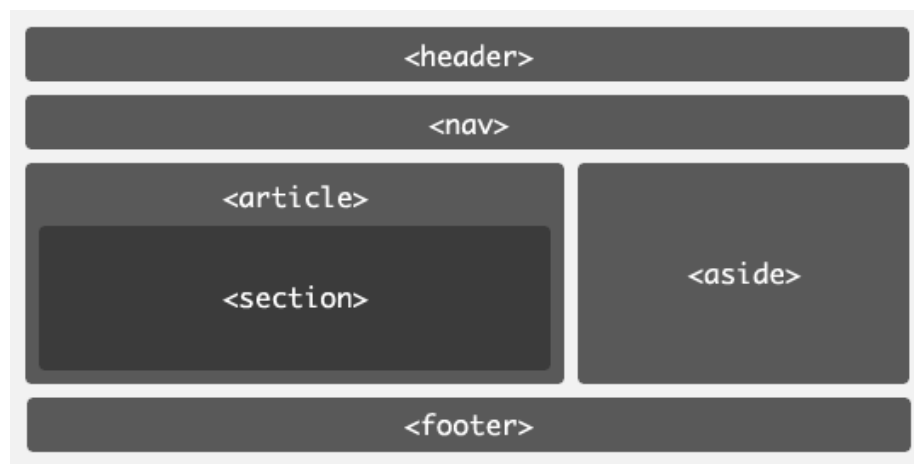


Figure 1: Esquema general de una página Web.

Los elementos en HTML usualmente son bien elementos “en línea” o bien elementos a “nivel de bloque”. Un elemento en línea ocupa sólo el espacio acotado por las etiquetas que lo definen. Un elemento en bloque pueden ser varias líneas.

Títulos y párrafos

El lenguaje HTML es muy cuidadoso con la organización de la información, por lo que lo primero que debemos conocer es cómo estructurar los títulos y cómo definir los párrafos de texto. Esto serían elementos de bloque.

Para dar formato a una palabra o conjunto de palabras (negrita, cursiva, subrayado, tachado. . .) usamos etiquetas que definen elementos “en línea”.

Párrafos Las etiquetas `<p>` y `</p>` se emplean para definir un bloque de texto que se comporta como un párrafo. Normalmente no dejaremos nunca una porción de texto suelta por la página web, sino que la rodearemos con esas etiquetas. El editor de texto se encargará de hacerlo por nosotros pero, si estamos usando otro tipo de editor, debemos asegurarnos de qué sucede.

Títulos Las etiquetas `<h1>` y `</h1>` se utilizan para definir un texto como título, indicando que es una cabecera (la h viene de header, cabecera en inglés) que queremos destacar sobre el resto del texto. Junto a `<h1>` contamos con `<h2>`, `<h3>` y así hasta `<h6>` para definir diferentes títulos, de mayor a menor importancia.

Una página web bien diseñada contará con estos encabezados para definir los distintos apartados del texto, con sus diferentes niveles. En la figura se puede observar cómo hemos incorporado algunos encabezados, en este caso `h1` y `h2`, a nuestro texto. Se consigue añadiendo el texto y a continuación seleccionando el encabezado deseado en cuadro de la parte izquierda.

Cada uno de los niveles de encabezado tiene una apariencia diferente de tamaño y tipo de letra. Este aspecto se puede modificar como veremos un poco más tarde. Saltos de línea y líneas separadoras

Para complementar las opciones de separación del texto, contamos con dos etiquetas más:

- `
` inserta un salto de línea en el texto. No genera un nuevo párrafo, sino que parte la línea en dos. Es un elemento puntual, que no lleva etiqueta de cierre.
- `<hr>` inserta un salto de línea en el texto, pero mostrando una línea horizontal visible.

Formato de palabras

Aunque existen etiquetas para poner el texto en negrita: `` o ``, cursiva `<i>`, etc. hoy día ya están obsoletas en HTML5. Ahora deberíamos cercar con `` (de *abarc*ar en inglés) el texto que queremos decorar y darle formato con CSS.

Ejemplos:

```
<p>
  Esto es un párrafo con el siguiente texto en rojo
  <span style="color: red;"> y esto un span dentro de un párrafo. </span>
</p>
<p>
  Y esto otro párrafo con el siguiente texto en negrita
  <span style="font-weight: bold;"> y esto un span dentro de un párrafo. </span>
</p>
<p>
  Y esto otro párrafo con el siguiente texto en cursiva
  <span style="font-style: italic;"> y esto un span dentro de un párrafo. </span>
</p>
<p>
  Y esto otro párrafo con el siguiente texto en negrita
  <span style="text-decoration: underline;"> y esto un span dentro de un párrafo. </span>
</p>
```

Esto da lugar en un navegador a la siguiente salida:

Esto es un párrafo con el siguiente texto en rojo **y esto un span dentro de un párrafo.**

Y esto otro párrafo con el siguiente texto en negrita **y esto un span dentro de un párrafo.**

Y esto otro párrafo con el siguiente texto en cursiva y *esto un span dentro de un párrafo.*

Y esto otro párrafo con el siguiente texto en negrita y esto un span dentro de un párrafo.

Figure 2: uso de span

Hiperenlaces o anclas

El elemento ancla o hiperenlace `<a>` crea un enlace a otras páginas de internet, archivos o incluso partes dentro de la misma página, direcciones de correo, o cualquier otra URL:

```
<a href="https://www.youtube.com/juanguedu">Mi canal de Youtube</a>
```

Imágenes

la etiqueta `` se emplea para insertar una imagen en la página web, pero por si sola no funciona correctamente. Necesita que le incorporemos un parámetro en el que indiquemos qué imagen será la que se muestre. Quedaría así:

```

```

En el ejemplo siguiente, además de indicar qué imagen se mostrará, establecemos el tamaño que ocupará en la pantalla:

```

```

Uso de audio y video en HTML5

Los elementos `<audio>` y `<video>` permiten la manipulación de nuevo contenido multimedia.

Imagina que tienes un vídeo que has grabado con el móvil en formato MP4 (Android) o MOV (iPhone), ¿cómo insertarlo en una página Web? Sencillo, sería algo así:

```
<video src="mi_pelicula.mp4" autoplay poster="imagen_inicial.jpg">
  Su navegador no soporta la etiqueta video.
</video>
```

Si nuestro navegador es moderno, veremos el vídeo, en caso contrario veríamos el mensaje *“Su navegador no soporta la etiqueta video”*.

Formularios en HTML5

Cuando abres un navegador y vas a la página de tu buscador favorito, **eso es un formulario**, fíjate cómo hay una caja donde escribes el texto a buscar y luego un botón para enviar que pulsaremos para ir a la siguiente página con los resultados de la búsqueda.

Aquí tienes un ejemplo de formulario. Fíjate en la etiqueta `<input>`. Verás que según el tipo de información que quieres consultar, es el atributo **type** de cada entrada de texto:

```
<p>Esto es un típico formulario Web</p>
<form action="getform.php" method="get">
  <label>Nombre: <input type="text"></label><br>
  <label>Apellido: <input type="text"></label><br>
  <label>E-mail: <input type="email"></label><br>
  <label>Edad: <input type="number" min="18" max="120"><br>
  <input type="submit" value="Submit">
</form>
```

Esto dará lugar a un formulario como éste:

Esto es un típico formulario Web

Nombre:

Apellido:

E-mail:

Edad:

Figure 3: Formulario información personal

Además el trabajar con HTML5 nos ofrece otra ventaja como es aprovechar su mejora de los formularios web: Validación de restricción (p.ej. si un campo es un número, podemos indicar entre qué números deberá estar comprendido), varios atributos nuevos, nuevos valores para `<input>` como el atributo `type` y el nuevo elemento `<output>`.

Observa cómo funciona esto de las restricciones en el ejemplo anterior cuando intentamos introducir una letra en la edad:

Esto es un típico formulario Web

Nombre:

Apellido:

E-mail:

Edad:

Introduce un número

Figure 4: Formulario Restricción Edad

Listas

Para declarar listas usamos o bien la etiqueta `` (*del inglés `ordered list`*) si queremos una lista numerada u **ordenada** o bien la etiqueta ``, si queremos una lista desordenada (*del inglés `unordered list`*).

El elemento `` del inglés **list item** o elemento de lista, declara cada uno de los elementos de una lista.

Ejemplo:

```
<ul>
  <li>primer elemento</li>
  <li>segundo elemento</li>
  <li>tercer elemento</li>
</ul>
```

dará como salida:

- primer elemento
- segundo elemento
- tercer elemento

Figure 5: Lista ordenada

Sin embargo, el siguiente ejemplo:

```
<ol>
  <li>primer elemento</li>
  <li>segundo elemento</li>
  <li>tercer elemento</li>
</ol>
```

dará como salida:

1. primer elemento
2. segundo elemento
3. tercer elemento

Figure 6: Lista ordenada

¿Has visto la diferencia entre usar `` y ``?

Plantilla de una página Web con hiperenlaces

Ahora que ya tenemos una ligera noción de qué es eso del HTML te proponemos el siguiente ejercicio: Vamos a crear una Web a partir del ejemplo de estructura básica. La página debe tener un título, un autor y al menos un hiperenlace. Recuerda guardar el archivo con la extensión html (“ejemplo.html”).

Editores de texto

Si no puedes localizar en el equipo un editor de texto plano con el que poder hacer las prácticas y tareas del proyecto, puedes instalar uno de los siguientes:

- Microsoft Visual Studio Code (si no lo quieres instalar también hay opción de usarlo en modo “portable”)
- Geany
- Notepad++

En los vídeos y en clase tenemos Visual Studio Code, luego ante duda te recomendamos ése.

En Windows, el más ligero y fácil de usar es Notepad++.

En Linux, el más sencillo es Geany.

Si no quieres instalar nada, puedes trabajar on-line con alguna Web como CodePen.

¿Qué tal, has podido crear tu primera página Web, ha sido fácil?

Semana 2: HTML5 - El modelo de cajas

Antes de comenzar con el modelo de cajas vamos a ver cómo se organizan los archivos de una página Web.

Una página Web se organiza en archivos (realmente HTTP lo llama objetos). Cuando hacemos una petición HTTP (nuestro navegador en nuestro PC llama al servidor Web en el equipo remoto donde está hospedada la página) no sólo nos descargamos texto, también puede haber imágenes, sonidos, vídeo. . .

Fíjate en esta petición (hemos activado el modo “Desarrollador Web” del navegador para que se vea la consola que tienes debajo):

Yo he visitado una página, pero sin embargo se han completado 5 solicitudes (mira en la esquina inferior izquierda). Para ver esa página hemos descargado 5 objetos: el fichero HTML, la hoja de estilo Bootstrap, el JavaScript jQuery, el JavaScript Bootstrap y el icono de la página Web o “favicon” (éste ha dado error 404, no lo ha encontrado).

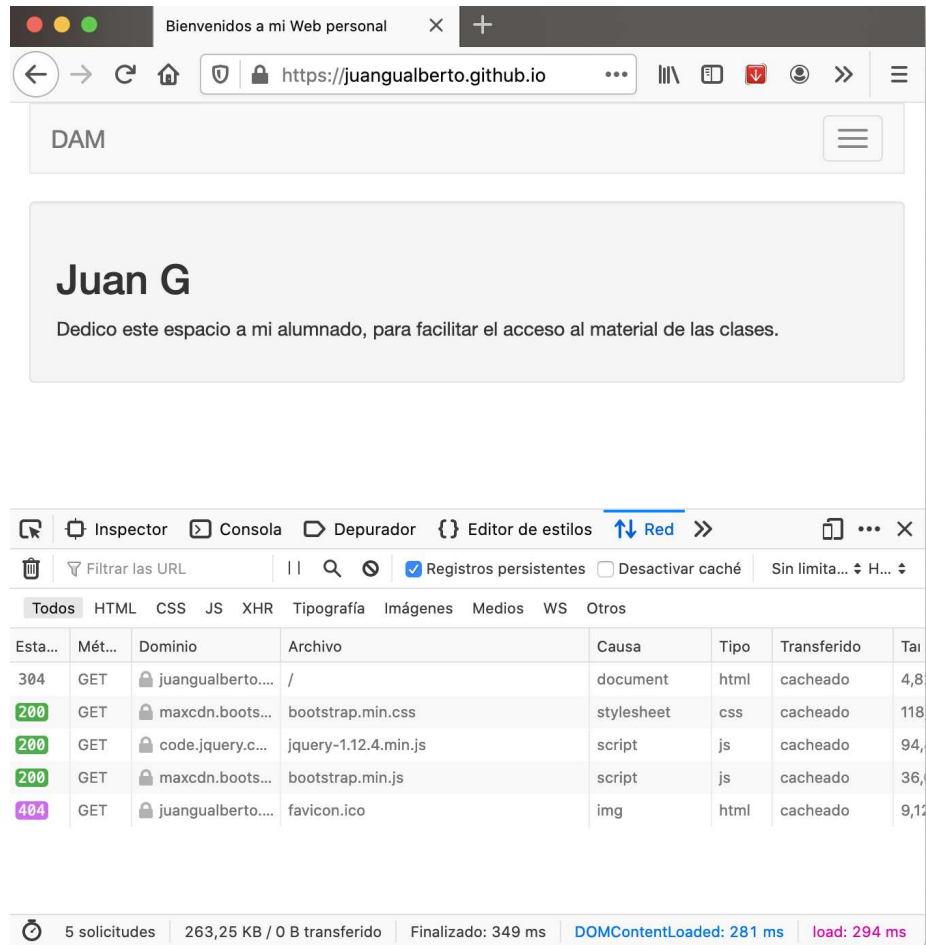


Figure 7: Firefox en modo Desarrollador Web

Como puedes observar, por sencilla que sea una Web, siempre tendremos varios archivos, luego **es importante organizar el contenido en carpetas** para que quede todo bien **ordenado**.

Un ejemplo de organización podría ser éste (muy importante, intenta **siempre** que estén todos los nombres de archivos y carpetas **en minúsculas**):

- el fichero index.html en la carpeta raíz de la página
- todas las fotos e imágenes en la carpeta **img**
- las hojas de estilo (para dar formato, colores, espaciado) al texto en la carpeta **css**
- el código JavaScript para dinamizar en la carpeta **js**



Figure 8: Estructura de archivos

La etiqueta <div>

Cada elemento HTML que podemos encontrar en una página web se encuentra rodeado de una caja con varias propiedades que pueden ser modificadas. Es lo que llamamos el modelo de cajas.

Estas opciones nos proporcionan un gran control sobre cómo debe situarse cada elemento.

Caja de un elemento HTML

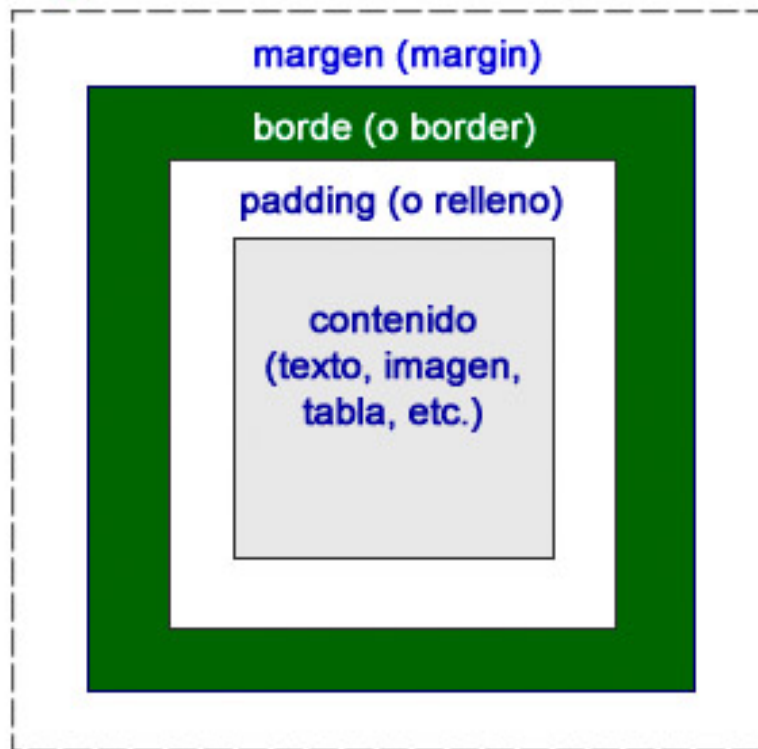


Figure 9: modelo de cajas

Altura y anchura de una caja

Cada elemento HTML de una página web cuenta con una anchura y una altura específica. En muchos casos esas dimensiones se las proporciona el propio contenido, como en un párrafo o una imagen, por ejemplo. Esos valores de anchura (`width`) y de altura (`height`) pueden ser modificados mediante las hojas de estilo, gracias a las propiedades del mismo nombre.

Así podemos hacer párrafos más estrechos, imágenes que se sobredimensionen o simplemente ajustar diferentes bloques, para que se acomoden correctamente en la pantalla.

Los valores `width` y `height` se acompañan de un valor numérico exacto o de un porcentaje, como en otras muchas propiedades.

```
body {  
principal { width: 400px;  
background-color: rgb(0, 126, 0);  
}
```

Margen

Comenzaremos con la imagen. Con tan sólo modificar su margen, observaremos cómo se distancia del resto de los elementos. Usaremos la propiedad `margin` seguida de un valor numérico o de un porcentaje:

```
img { margin: 20px; }
```

Relleno

Probaremos ahora a modificar su relleno, es decir, la distancia imaginaria entre un hipotético borde y la imagen propiamente dicha. Para ello emplearemos la propiedad `padding`, exactamente igual que hicimos con la anterior. Probemos con un valor menos exagerado:

```
img { padding: 5px; }
```

Borde

Si recargamos la página con esta incorporación, observaremos que, en efecto, la imagen se separa un poco más, esos 5 píxeles por cada lado, pero no es posible distinguir dónde acaba el efecto del margen y comienza el del relleno. Para poder diferenciar los valores, deberíamos tener un borde en la imagen.

```
img {  
border-width: 2px;  
border-style: solid;
```

```
border-color: #007000;
}
```

Con los conocimientos que tenemos ya de CSS podemos intuir con facilidad qué es lo que hace cada una de esas tres propiedades: en una definimos el grosor del borde, en otra el tipo de línea y en la última su color.

Laterales

Tanto margin, como padding y border se pueden emplear para modificar laterales de una caja, con independencia de los demás. Añadiendo a cada uno de ellos la variación -left (izquierda), -right (derecha), -top (arriba) o -bottom (abajo) conseguimos que sólo afecte al valor o valores indicados.

En la figura hemos aplicado estas propiedades para el título de la página:

```
h1 {
margin-top: 40px;
padding-left: 5px;
padding-right: 5px;
border-top-width: 2px;
border-top-style: dotted;
border-top-color: #007000;
border-bottom-width: 2px;
border-bottom-style: double;
border-bottom-color: #007000;
}
```

Más opciones para los bordes

Para los bordes podemos definir tres propiedades: su anchura, su estilo y su color. La anchura y el color se definen con las medidas habituales y los sistemas que ya hemos analizado. El estilo, por su parte, se basa en una serie de valores concretos:

```
dotted: punteado.
dashed: línea discontinua.
solid: línea continua.
double: línea doble.
groove: tipo de relieve.
ridge: tipo de relieve.
inset: tipo de relieve.
outset: tipo de relieve.
none: empleado para indicar que no habrá borde
```

Es cuestión de probar algunos modelos. El valor solid es la línea sencilla, la más empleada.

Como ya sucedía con otras propiedades, podemos reagrupar los valores referidos a los bordes en una sola propiedad genérica denominada `border`. Para ello estableceremos los valores separados por espacios y en el orden de tamaño, estilo y color, como en este ejemplo que haría la misma función que el recuadro anterior:

```
img { border: 2px solid #007000; }
```

Esquinas redondeadas

Con los estilos actuales podemos trazar un borde alrededor de una figura y que tenga sus esquinas redondeadas.

La propiedad que lo permite es `border-radius`, acompañada de un valor numérico. El ejemplo anterior, con la incorporación de esta propiedad, daría como resultado el rectángulo de la figura:

```
img {  
border: 2px solid #007000;  
border-radius: 25px;  
}
```

Sombras

Las modernas hojas de estilo proporcionan a cualquier elemento la capacidad de proyectar una sombra. Ya vimos que esto funcionaba con el texto, pero además contamos con la propiedad `box-shadow` para crear sombras en cualquier caja de nuestra página web, lo que hace que sea posible aplicárselo a cualquier elemento.

```
table { box-shadow: 8px 8px 6px #aaaaaa; }
```

Los valores que conforman la sombra son similares a los que vimos para las sombras de texto, es decir, desplazamiento horizontal, vertical, difuminado y color de sombra.

Elementos flotantes

Los elementos de una página web pueden reubicarse a la izquierda o a la derecha con tan sólo emplear la propiedad `float`, haciendo que el resto del contenido se sitúe alrededor de ese elemento.

En el siguiente ejemplo la regla:

```
img { float: left; }
```

provoca que el texto se sitúe alrededor de la imagen.

Ejercicio semana 2

Deberás crear todas las carpetas necesarias para ir haciendo la Web (css, js, img, snd...).

Hay que crear el fichero index.html con el siguiente contenido:

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="author" content="Juan Gualberto">
    <meta name="copyright" content="GNU GPLv3">
    <title>Buscaminas</title>
  </head>
  <body>
    <nav>
      <!-- Aquí irá el menú superior -->
    </nav>
    <section>
      <!-- Aquí irán las vías, las minas y el smiley -->
    </section>
    <section>
      <!-- Aquí estará el tablero de juego -->
    </section>
    <aside>
      <!-- Esto no será necesario y lo borrarémos -->
    </aside>
  </section>
  <footer>
    <!-- Aquí irá información adicional en el pie -->
  </footer>
</body>
</html>
```

Semana 3: CSS3 - Con un poco de estilo. Menús

Ejercicio horario de clase

Semana 4: JavaScript - ¡Hola Mundo!

Semana 5: JavaScript - Variables, clases, métodos y funciones

Ejercicio: Adivina en qué número pienso

Semana 6: JavaScript - El árbol DOM

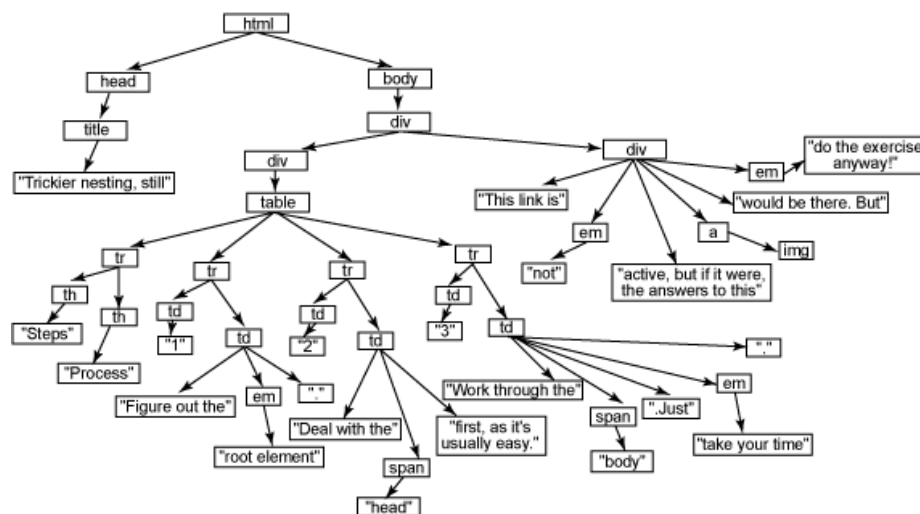


Figure 10: El árbol DOM

Semana 7: JavaScript - Pensamiento computacional. Algoritmos

Ejercicio: La letra del DNI

Semana 8: JavaScript - Almacenando datos