Módulo 2 - Tema 2
HTML5 & CSS3: Intermediate

Antonio Cebrián

M2T2 - HTML5 & CSS3: Intermediate

CODE SPACE ACADEMY

Índice

HTML5

- Agrupar elementos: y
- Abreviaturas y citas
- Código y texto preformateado
- Elementos <meta>
- Tablas mejoradas
- Listas descriptivas
- Seccionamiento y elementos semánticos

CSS3

- Identificación de elementos
- Agrupamiento y anidamiento de los selectores
- Precedencia de las reglas CSS
- Pseudoclases, herencia
- Márgenes, relleno, bordes, fuentes y fondos
- Visualización de los elementos

Índice

CSS3

- Pseudo-elementos
- Layout: posicionamiento absoluto, flexbox y grid

1. HTML5

Agrupar elementos

Mientras que la mayoría de las etiquetas HTML aplican un significado (p hace un párrafo, h1 crea un encabezado), ****** y *** *** se utilizan para agrupar elementos HTML.

Son lo que se conoce también como elementos contenedores.

La diferencia entre ambos es que ****** es un elemento de línea y generalmente se usa para agrupar HTML dentro de una línea (por ejemplo una palabra de un párrafo).

Por su parte, un elemento *** *** (division) es un elemento de bloque, algo equivalente a tener un salto de línea antes y después, y suele agrupar bloques de código.

Ver ejemplo en GitHub

Abreviaturas

La etiqueta ***<abbr>*** se utiliza para marcar una abreviatura, una forma abreviada de una palabra o frase. La frase expandida que representa la abreviatura se puede definir en el valor del atributo ***title=""***. Esto es opcional pero recomendado.

Ver ejemplo en GitHub

1. HTML5

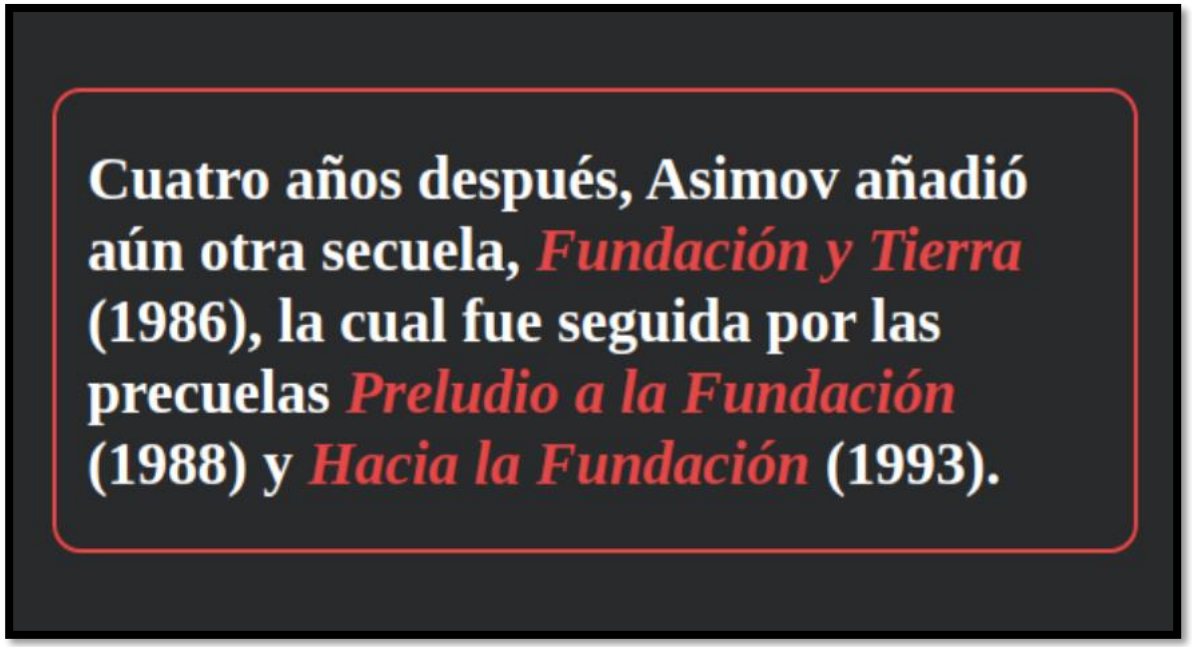
Citas

Para las citas se utilizan ***<blockquote>*** y ***<q>***. El primero se utiliza principalmente para citas multilínea, y el segundo para citas de una línea.
Si la fuente de la cita se puede encontrar en la Web, el atributo cite="" se puede usar para indicar su origen.

Ver ejemplo en GitHub

Además del atributo ***cite=""***, también hay una etiqueta ***<cite>***, y se suele usar para definir el título de una obra, como un libro.

Ver ejemplo en GitHub



1. HTML5

Código y texto preformateado

Si queremos incluir algún código, por ejemplo JavaScript, en nuestros archivos HTML, lo incluimos en etiquetas ***<code>***. También podíamos usarlo para incluir una fórmula matemática.

La etiqueta ***<pre>*** la utilizaremos cuando queramos mostrar un texto que incluya espacios en blanco y saltos de línea.

Ver ejemplo en GitHub

Meta

Los elementos <meta> de metainformación, aportan información de la página.

El atributo ***charset=""*** se puede usar para definir el conjunto de caracteres de un documento HTML.

El elemento ***<meta> "viewport"***, permite definir qué área de pantalla está disponible al renderizar un documento, nivel de escalado y el zoom que debe mostrar inicialmente.

Más info en MDN

1. HTML5

Mejorando el elemento <table>

Si queremos destacar el contenido de la primera fila, a modo de encabezado de la tabla, cambiaremos los ***<td>*** de la primera fila por elementos ***<th>***.

También es posible definir el intervalo de filas o columnas que ocupa una celda, si le añadimos el atributo ***colspan=""*** o ***rowspan=""***.

Ver ejemplo en GitHub

Listas descriptivas

Las listas de descripción comienzan con el elemento ***<dl>***, y se usan para contener una lista de términos y sus definiciones.

tienen elementos ***<dt>***, para especificar los términos, seguidos de elementos ***<dd>***, que son las descripciones asociadas a los elementos ***<dt>***.

Ver ejemplo en GitHub

Seccionamiento

El encabezado de una página o sección, formado por etiquetas ***<header>***, contiene la introducción del contenido o la navegación.

A su vez, la navegación ***<nav>*** define un conjunto de enlaces, para navegar a través del sitio web.

Por su parte, el ***<footer>*** es el pie de página de la sección en la que está contenido.

1. HTML5

En elemento ***<main>*** especifica el contenido principal de un documento HTML, y por lo general, suele ir ubicada entre los elementos ***<header>*** y ***<footer>***.

Ver ejemplo en GitHub

El elemento ***<section>*** representa una sección genérica de un documento, mientras que ***<article>*** se puede usar para marcar una sección independiente de contenido, que tenga significado por sí misma.

Ver ejemplo en GitHub

Un elemento ***<aside>*** se utiliza para incluir contenido que difiere del contenido donde está ubicado, pero que está relacionado indirectamente.

El elemento ***<figure>*** se utiliza principalmente para ilustraciones, diagramas o fotos, mientras que ***<figcaption>*** describe el contenido de ***<figure>***, a modo de título.

Ver ejemplo en GitHub

2. CSS3

Selectores más específicos

Es posible utilizar selectores personalizados, además de los selectores propios de los nombres de etiqueta.

Los más habituales son los selectores de clase. Se incluyen en el atributo ***class=“”***, y se escriben en CSS con un punto al principio.

También se pueden asociar identificadores, con el atributo ***id=“”***, escribiendo el nombre del identificador comenzando por el símbolo #.

Ver ejemplo en GitHub

Los selectores se pueden agrupar y anidar, así asignamos las mismas propiedades a varios selectores sin repetir la regla.

Precedencia de las reglas CSS

Por defecto, se aplican los estilos predefinidos del navegador, y se sobrescribirán por la hoja de estilos que nosotros hayamos definido.

Dentro de la hoja, se tiene en cuenta la especificidad de los selectores, y en caso de tener la misma, los estilos que aparecen en último lugar se imponen.

Una etiqueta tiene especificidad 1, una clase especificidad 10 y un id especificidad 100.

2. CSS3

Selectores más específicos

Hay una excepción, las declaraciones marcadas como !important, que serán consideradas de máxima prioridad.

Ver ejemplo en GitHub

Pseudoclases

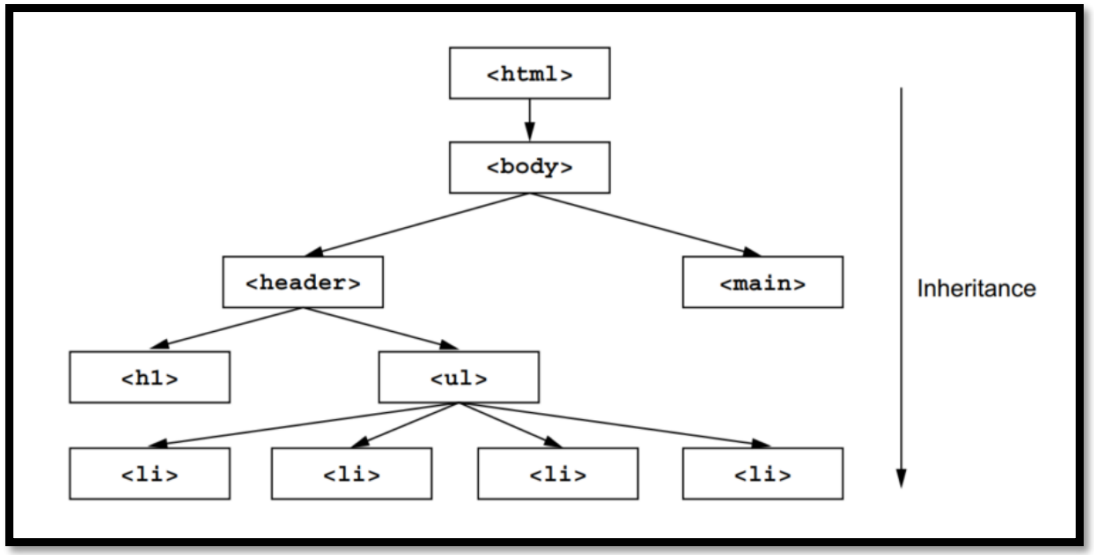
Las pseudoclases especifican un estado o una relación con el selector. Se escriben de la forma ***selector:pseudoclase*** {}.
Hay pseudoclases dinámicas, que responden a un evento, como ***:active***, ***:hover*** o ***:focus***.

Ver ejemplo en GitHub

Referencia completa de las pseudoclases.

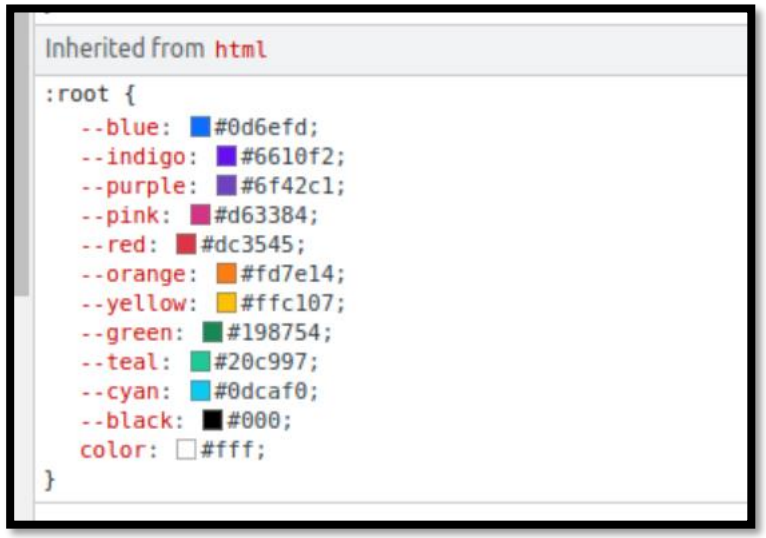
Herencia

Si un elemento no tiene un valor en cascada para una determinada propiedad, puede heredar su valor de un elemento antecesor.



2. CSS3

Podemos apreciar este comportamiento mediante las DevTools del navegador.



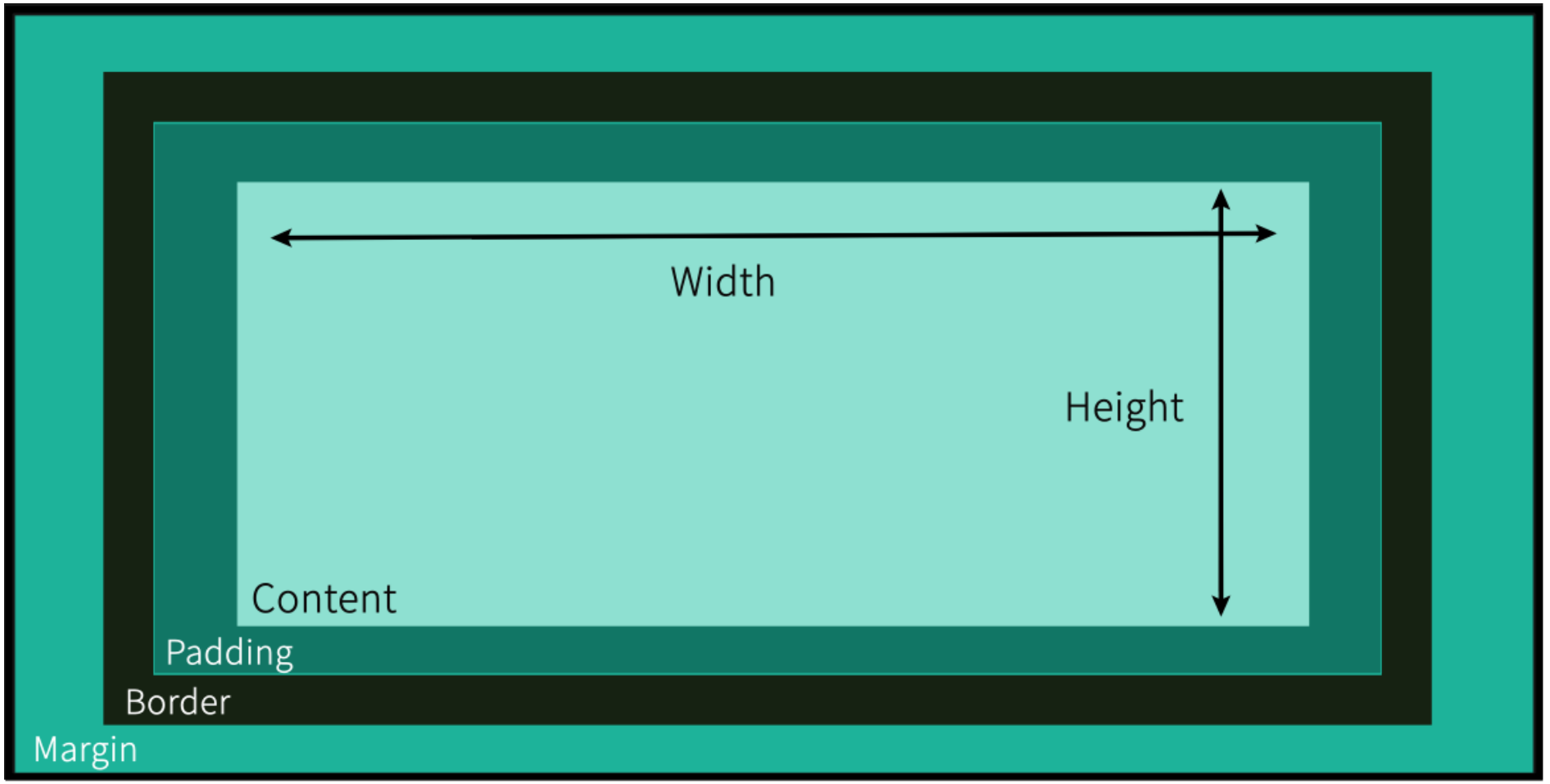
A veces, puede ser necesario heredar cuando un valor en cascada lo impida. Se puede conseguir con la palabra clave: ***inherit***.

Hay una forma de restablecer los valores por defecto de un estilo dado. Mediante la palabra clave: ***initial***.

Márgenes, relleno, bordes, fuentes y fondos

Las propiedades ***margin*** y ***padding*** representan el espacio desde el borde de un elemento hacia el exterior, y hacia el interior respectivamente.

2. CSS3



2. CSS3

Algunas propiedades de CSS permiten asignar una lista de valores, por ejemplo, ***margin*** resume ***margin-top*** , ***margin-right*** , ***margin-bottom*** y ***margin-left*** en la forma de ***margin: arriba derecha abajo izquierda.***

Otra agrupación posible es 2 a 2, siendo los 2 primeros valores arriba y abajo, y los 2 siguientes derecha e izquierda.

Además, se pueden agrupar 3 valores, el primero se aplica a la parte superior , el segundo a la derecha e izquierda , el tercero a la parte inferior.

https://developer.mozilla.org/es/docs/Web/CSS/margin

Padding puede aplicarse exactamente del mismo modo.

https://developer.mozilla.org/es/docs/Web/CSS/padding

Por otra parte, la propiedad ***border*** permiten especificar el estilo, el ancho y el color del borde de un elemento.

https://developer.mozilla.org/es/docs/Web/CSS/border

La propiedad ***font*** puede resumir ***font-style***, ***font-weight***, ***font-size***, ***line-height*** y ***font-family***.

2. CSS3

Respecto a la propiedad ***background***, es un resumen de:

- ***background-color*** → muestra un color de fondo.
- ***background-image*** → ubicación (path) de la imagen.
- ***background-repeat*** → determina si la imagen de fondo se repite o no. Su valor puede ser:
  - **repeat**, el equivalente a un efecto "azulejo" en todo el fondo.
  - **repeat-y**, repitiendo en el eje y, de arriba abajo.
  - **repeat-x**, igual pero de lado a lado, en el eje x.
  - **no-repeat**, la imagen no se repite.
- ***background-position*** → que puede ser top, center, bottom, left, right, una longitud o un porcentaje, o cualquier combinación, por ejemplo top right.

Ver ejemplo en GitHub

Hay más propiedades para editar los fondos:

- ***background-attachment*** → Especifica si un fondo está fijo en el viewport o si se desplaza junto con el contenido del cuadro que lo contiene.
- ***background-clip*** → La parte de una caja sobre la que se pinta un fondo (color o imagen).
- ***background-image*** → La imagen de fondo de una caja.

2. CSS3

- ***background-origin*** → El área de posicionamiento del fondo, es decir, la parte de un cuadro con la que se relaciona la posición del fondo.
- ***background-size*** → El tamaño de una imagen de fondo.

Ver ejemplo en GitHub

La referencia completa en MDN

Visualización de los elementos

Los tipos principales de visualización son en línea, en bloque y ninguno, y pueden modificarse con la propiedad ***display,*** y los valores ***inline***, ***block*** y ***none***.

El valor ***inline*** hace que los elementos se muestren siguiendo el flujo de la línea.

El valor ***block*** hace que un elemento ocupe todo el ancho del elemento que lo contiene.

Los elementos ***inline-block*** fluyen con el texto y demás elementos como si fueran elementos en línea y respetan el ancho, el alto y los márgenes verticales.

Ver ejemplo en GitHub

2. CSS3

Pseudoelementos

Los pseudo-elementos funcionan de forma similar a las pseudo-clases:

- selector**::pseudo-elemento** { propiedad: valor; }

El pseudoelemento ***::first-letter*** aplica estilos a la primera letra de la primera línea de un elemento a nivel de bloque , siempre que no esté precedido por otro contenido (como imágenes o tablas en línea).

El pseudo-elemento ***::first-line*** aplica estilos a la primera línea de un elemento de nivel de bloque.

Ver ejemplo en GitHub

Los pseudo elementos ***::before*** y ***::after*** se utilizan junto con la propiedad content: para colocar el contenido a ambos lados de una caja sin tocar el HTML.

El valor de la propiedad ***content:*** puede ser una cadena de caracteres entrecomillada o una imagen de la forma ***url(ruta/nombre_imagen.jpg)***.

Ver ejemplo en GitHub

2. CSS3

Layout

Hay varias propiedades que nos ayudan a ubicar los bloques de contenido de la web.

Mediante la propiedad ***position:*** podemos definir si la posición de una caja es absoluta, relativa, estática o fija:

- ***static*** → es el valor predeterminado, se posiciona según el flujo normal del contenido.
- ***relative*** → es muy parecido a static pero la caja se puede desplazar respecto a su posición original con las propiedades top, right, bottom y left.
- ***absolute*** → posiciona el elemento respecto al bloque que lo contenga, mediante top, right, bottom y left .
- ***fixed*** → posiciona de forma absoluta pero en relación a la ventana del navegador, por lo que el elemento se verá siempre en el mismo lugar de la pantalla, incluso cuando la página se desplaza.

La propiedad ***float*** permite desplazar elementos a izquierda o derecha con respecto a su contenedor, permitiendo que el resto de los elementos fluyan a los lados, según hacia donde flote.

Ver ejemplo en GitHub

2. CSS3

Respecto a la altura de los elementos, está determinada por su contenido, y si la establecemos de manera explícita, su contenido puede desbordarse.

Se puede controlar este comportamiento con la propiedad ***overflow***, que acepta los valores:

- ***visible*** → Por defecto. El desbordamiento no se recorta. El contenido se muestra fuera de la caja del elemento si excede su altura.
- ***hidden*** → El desbordamiento se recorta y el resto del contenido será invisible.
- ***scroll*** → El desbordamiento se recorta y se agrega una barra de desplazamiento para ver el resto del contenido.
- ***auto*** → Similar a scroll, pero agrega barras de desplazamiento cuando es necesario.

2. CSS3

Métodos actuales para el Layout

Quizá el más habitual actualmente es ***Flexbox (Flexible Box Layout)***.

Aplicar ***display: flex*** a un elemento lo convierte en un ***contenedor flex***, y sus hijos directos se convierten en ***elementos flex***.

Los elementos flex se alinean uno al lado del otro, de izquierda a derecha, todos en una fila.

El contenedor flexible llena el ancho disponible como un elemento de bloque, pero los elementos flexibles no tienen por qué ocupar la anchura de su contenedor.

Además, los elementos flexibles tienen todos la altura determinada de forma natural por su contenido.

Ver ejemplo en GitHub

Las propiedades del contenedor flexible son:

- ***flex-direction*** → define en qué dirección el contenedor quiere apilar los elementos flexibles.
- ***flex-wrap*** → especifica si los elementos flexibles deben ajustarse o no, es decir, rellenan la fila hasta que no caben más y entonces pasan a la siguiente.

2. CSS3

- ***flex-flow*** → es una propiedad abreviada para configurar las propiedades flex-direction y flex-wrap (flex-flow: row wrap;).
- ***justify-content*** → se utiliza para alinear los elementos flexibles en el eje horizontal.
- ***align-items*** → se utiliza para alinear los elementos flexibles en el eje transversal.
- ***align-content*** → Controla el espacio entre las líneas en el eje transversal.

Ver ejemplo en GitHub: Menú de navegación con flexbox

Guia completa en CSS -Tricks

Una referencia visual para flexbox (flexbox cheat sheet)

Junto a flexbox, hay una nueva especificación llamada ***Grid Layout Module***.

Grid permite definir un diseño bidimensional de columnas y filas y luego colocar elementos dentro de la cuadrícula.

El elemento contenedor de la cuadrícula, llevará la propiedad ***display: grid***, y sus elementos hijos que serán elementos de la cuadrícula.

El contenedor grid se comporta como un elemento de bloque, llenando el 100% del ancho disponible.

Los elementos grid pueden ocupar, desde una celda de la cuadrícula hasta varias filas o columnas.

2. CSS3

Mediante las propiedades ***grid-template-columns*** y ***grid-template-rows*** definimos cuál será el tamaño de cada una de las columnas y filas de la cuadrícula.

Podemos asignar fácilmente valores proporcionales mediante la unidad ***fr***, que representa una fracción de columna o fila.

Sería equivalente a flex-grow en flexbox. Por ejemplo:

- ***grid-template-columns: 1fr 1fr 1fr;***

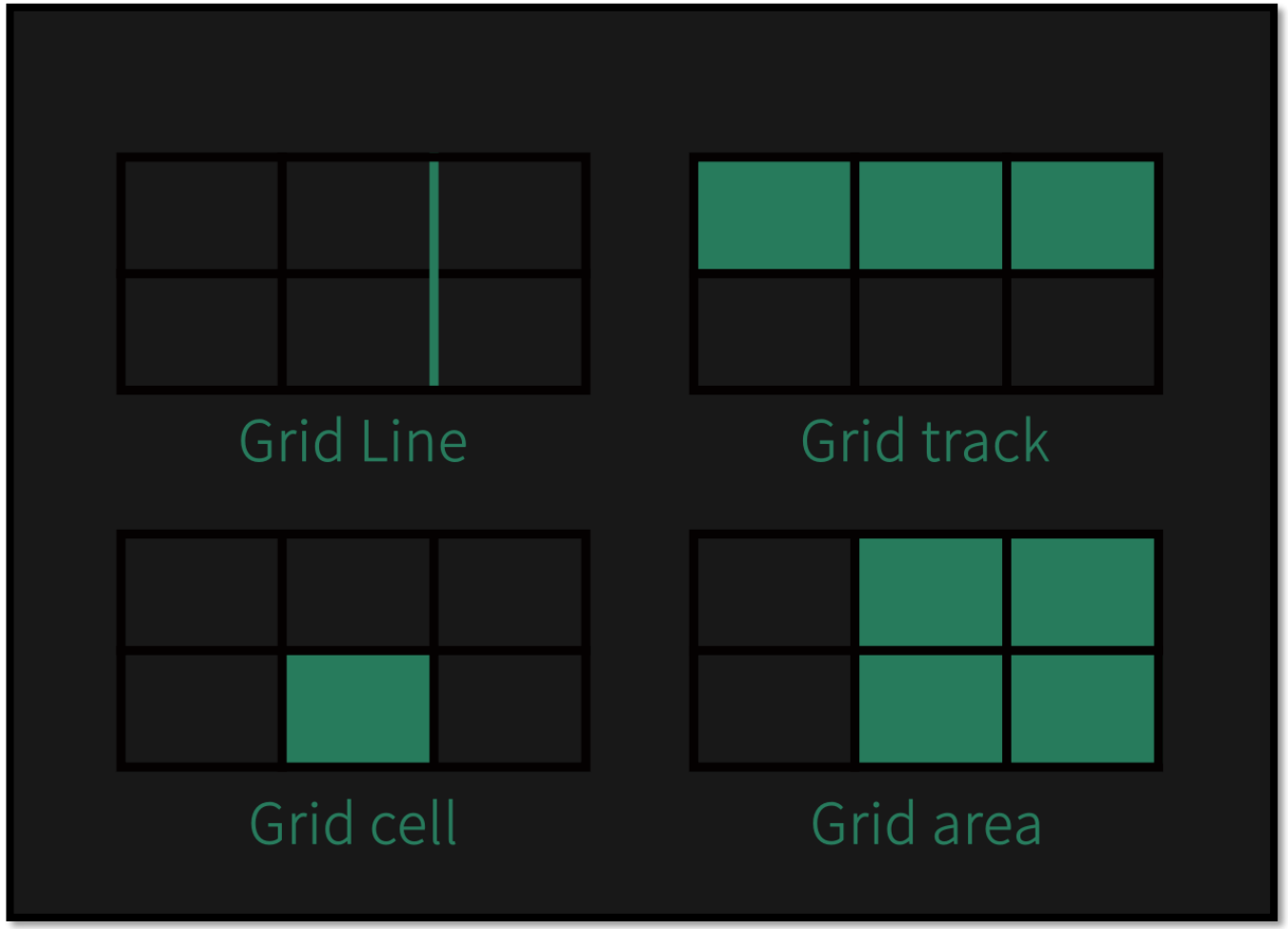
Declara tres columnas de igual tamaño. El espacio entre celdas se declara con la propiedad ***grid-gap***:

- ***grid-gap: .5em 1em;***

Una cuadrícula grid tiene la siguiente estructura:

- ***Grid line*** → Una línea que puede ser vertical u horizontal y se sitúa a ambos lados de una fila o columna.
- ***Grid track*** → cualquiera de las filas o columnas que componen la cuadrícula de un extremo al otro.
- ***Grid cell*** → similar a la celda de una tabla, es la unidad básica del elemento grid.
- ***Grid area*** → un grupo de celdas que forman un área rectangular.

2. CSS3



2. CSS3

Veamos las características de grid con ejemplos:

- Ubicación de los elementos grid
- Elementos que ocupan varias filas/columnas
- Ordenar los elementos por columnas
- Establecer el orden de los elementos

Guia completa en CSS -Tricks

Una referencia visual para grid (grid cheat sheet)

Combinando grid y flexbox

Módulo 2 - Tema 2
HTML5 & CSS3: Intermediate

Antonio Cebrián