CSS

Índice

[Introducción 3](#_Toc61941515)

[Historia 3](#_Toc61941516)

[Funcionamiento 5](#_Toc61941517)

[Tabla de ejercicios 5](#_Toc61941518)

[Conceptos básicos de CSS 6](#_Toc61941519)

[Tamaño y colores 7](#_Toc61941520)

[Unidades absolutas 7](#_Toc61941521)

[Unidades relativas 7](#_Toc61941522)

[Colores 8](#_Toc61941523)

[Selectores, herencia y cascada 9](#_Toc61941524)

[Selectores 9](#_Toc61941525)

[Selectores simples 9](#_Toc61941526)

[Selector de etiqueta 9](#_Toc61941527)

[Selector descendiente 10](#_Toc61941528)

[Selector de clase 11](#_Toc61941529)

[Selector de identificador de elemento 11](#_Toc61941530)

[Selector universal 11](#_Toc61941531)

[Selector de atributo 13](#_Toc61941532)

[Selector de hijos 13](#_Toc61941533)

[Selectores adyacentes 13](#_Toc61941534)

# Introducción

Las hojas de estilo en cascada es un lenguaje de estilos que define el aspecto, la presentación y la posición que tendrán los diferentes elementos que componen una pagina Web. Principales ventajas que nos aportan las hojas de estilos:

* Nos permite obtener una completa separación entre el contenido y la presentación de los datos
* Posibilidad de modificar el diseño de una página Web por interacción del usuario.
* Con una hoja de estilos bien diseñada, conseguiremos una mejora en la velocidad de carga de nuestra pagina Web al economizar líneas de código.
* Al usar hojas de estilos, indirectamente estamos evitando el uso de ciertos elementos que no cumplen con las normas de accesibilidad, como, por ejemplo, el uso de tablas como base del diseño de una página Web.

## Historia

El fundador de este lenguaje fue **Håkon Wium Lie**. Hoy en día, son 4 las versiones disponibles de CSS, por noma general, cada versión ha ido aumentando las prestaciones y el alcance de las versiones anteriores.

|  |  |
| --- | --- |
| CSS1 | Ya contaba con el visto bueno del W3C. Con esta primera versión ya se podían configurar aspectos como las propiedades de texto tales como color, tamaño y grosor, propiedades de los párrafos o bloques de texto y propiedades de las capas. |
| CSS2 | Fue desarrollada íntegramente por el W3C y aumentaba sus funcionalidades como posicionamiento absoluto y relativo de las capas, propiedad *z-index* de las capas y propiedades de sombra y texto bidireccional. |
| CSS 2.1 | Esta versión trata de solucionar errores de las versiones anteriores y eliminar ciertos mecanismos que, con el paso del tiempo, se habían vuelto obsoletos. |
| CSS3 | CSS3 es modular, ya que la especificación de estilos fue creciendo tanto, que hubo que desarrollarla modularmente. De hecho, es una especificación que se viene desarrollando desde 1999.  Cada módulo de CCS3 tiene su propia madurez, así que la estandarización de los módulos es propio. Podemos tener un módulo que sea ya estándar y otro módulo que esté todavía en un proceso de estandarización. De hecho, se estima que no va a haber una versión de CSS4, sino que habrá módulos en su versión 4 y módulos que todavía no habrán progresado.  Entre las cosas que podemos hacer con CSS3 destacamos algunas:   * Permite hacer muy fácilmente cosas como los bordes redondeados, que con versiones anteriores de CSS3 eran bastante difíciles de conseguir. * Permite dar gradientes de colores, es decir, si queremos dar a una etiqueta o a una parte de la página web un color de fondo, ya no nos vemos limitados a darle un único color, sino que podemos establecer un gradiente, una transición de un color a otro. * Se pueden transformaciones en muchas de las propiedades de los elementos de la página web e incluso animaciones utilizando solo CSS3. * Permite maquetar de manera mucho más fácil utilizando contenedores Flex y Grid. * Permite utilizar las media-queries, que permiten elegir una u otra hoja de estilos dependiendo de las propiedades en la pantalla que tengamos, para que la página web cambie su presentación si se muestra en una pantalla grande, en un móvil, en una tableta e incluso en la televisión. |

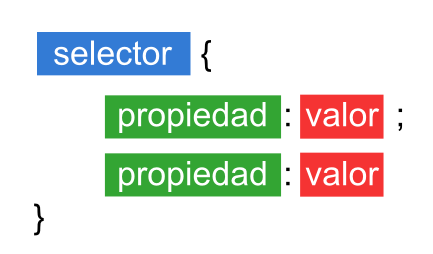
## Funcionamiento

El uso de CSS en un documento HTML nos permite básicamente aplicar un diseño o estilo a un elemento o grupo de elementos sin necesidad de utilizar los atributos propios de cada etiqueta.

## Tabla de ejercicios

|  |  |
| --- | --- |
| **Ejercicio 9** | Estructura |
| **Ejercicio 10** | Propiedad Input |
| **Ejercicio 11** | Formulario |
| **Ejercicio 12** | Multimedia |

# Conceptos básicos de CSS

Una hoja de estilos esta compuesta por una serie de reglas que, dependiendo del ámbito en el que se hayan definido, se aplicaran a unos elementos HTML u otros. Las partes que conforman una regla CSS son:

* Selector: Termino que especifica el elemento o conjunto de elementos a los que se le aplicara el formato definido en la regla CSS. Puede estar formado por multitud de variables: nombres de un elemento CSS, identificado de un elemento, clase de un elemento, etc.
* Propiedad: Nombre de la propiedad que estamos configurando, por ejemplo, con *Font-size* estaremos definiendo el tamaño del texto.
* Valor: Valor escogido para la propiedad seleccionada, este valor debe ir definido tal cual, sin ningún tipo de comillas y finalizar con un punto y coma.

## Tamaño y colores

Las unidades de medida usadas en las hojas de estilos modifican en según que elementos propiedades como altura, tamaño de márgenes, tamaño del relleno y tamaño de fuente. El valor usado como medida siempre será un numero entero o decimal seguido de la unidad de medida especificada de cada caso.

Hay ciertas propiedades que admiten valores negativos en sus unidades de medida, sea cual sea su valor. Es importante no olvidarse nunca de indicar la medida, ya que, en caso contrario, la declaración se ignoraría por completo.

La especificación CSS divide las unidades de medida en dos tipos:

### Unidades absolutas

Las medidas absolutas son medidas cuyo valor real no ha sido calculado tomando como referencia el valor de otro elemento, el valor de una unidad absoluta no depende de otro valor de referencia.

### Unidades relativas

El valor de las unidades relativas siempre estará calculado tomando como referencia el valor de otro elemento. Actualmente, es el mecanismo de medida mas utilizado en CSS, ya que la flexibilidad que aportan los valores relativos hace que los elementos se adapten perfectamente a cualquier escenario.

### Colores

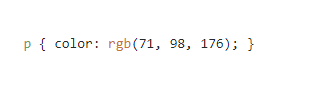
El estándar CSS especifica cinco formas distintas de definir el color en las hojas de estilos: código RGB hexadecimal, código RGB decimal, código RGB porcentual, colores del sistema y colores predefinidos. En líneas generales, un modelo de colores RGB se basa en la mezcla de los tres colores básicos de la luz: rojo, verde y azul. La intensidad de cada uno de estos colores viene representada por un código numérico.

Dicho esto, los códigos RGB porcentual y colores del sistema están obsoletos entonces se explicará los otros tipos:

* RGB hexadecimal: Es la representación en hexadecimal de los códigos de colores RGB, el valor definido en la propiedad CSS debe ir precedido del símbolo #.



* RGB decimal: Es la representación decimal de los códigos de colores RGB.



* Colores predefinidos: El estándar CSS tiene definidas ciertas palabras claves para representar los colores mas comunes de la paleta, estas palabras claves no son más que el nombre del color en inglés.



# Selectores, herencia y cascada

Uno de los aspectos mas importantes a la hora de trabajar con hojas de estilos es tener bien claro el mecanismo que nos permite seleccionar el elemento o grupo de elementos a los que vamos a aplicarle alguna regla CSS.

## Selectores

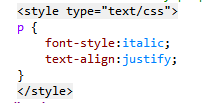
El selector es el primer término o patrón que precede a una regla CSS y es el encargado de especificar el elemento o elementos de un documento HTML al que se aplicara dicha regla. El estándar CSS define dos grandes grupos de selectores: selectores simples y selectores avanzados.

### Selectores simples

Son aquellos selectore3s que están formados por elementos y atributos propios del lenguaje HTML, estos selectores se pueden usar de forma única o bien combinados con otros.

### Selector de etiqueta

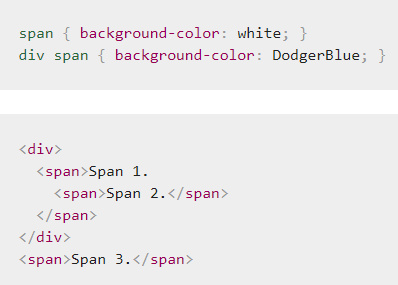
El valor de este selector es el nombre de cualquier etiqueta o elemento de un documento HTML, exceptuando aquellas etiquetas que forman parte de la cabecera del documento y que tienen implicación en el contenido de la página.



### Selector descendiente

Tiene como finalidad seleccionar elementos que se encuentran definidos dentro de otros elementos. La declaración del selector consiste en listar los elementos de mayor a menos profundidad separados por espacios. El resultado de aplicar esta regla CSS será que únicamente los elementos que se encuentren encerrados dentro de una capa tendrán un tamaño de letra,

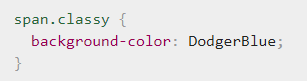
La regla definida mediante un selector descendente únicamente se aplicará al ultimo elemento definido en el selector.

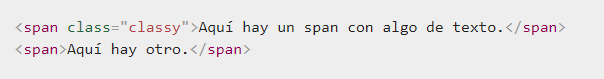


### Selector de clase

Este tipo de selector nos permite aplicar una determinada regla CSS a cualquier elemento HTML que tenga definido un atributo class del mismo valor que el indicado en el propio selector.

Podemos utilizar el selector de clase en combinación con el nombre de un elemento y de esta forma aumentar el nivel de concreción.

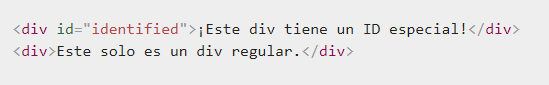




### Selector de identificador de elemento

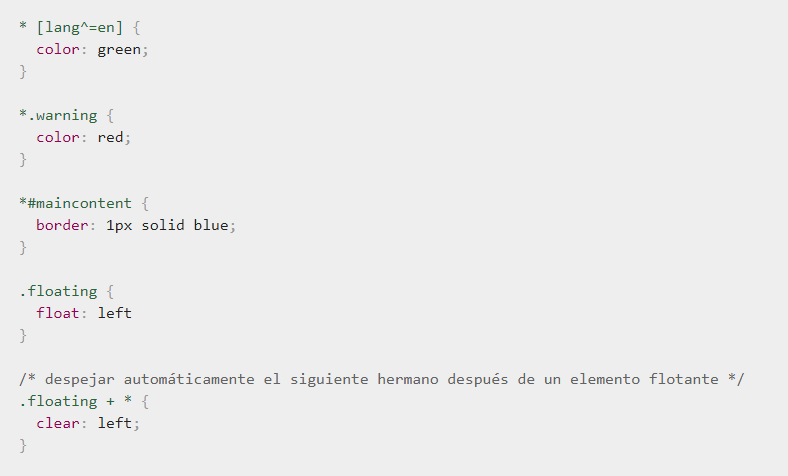
Este selector nos permite aplicar una regla CSS al elemento HTML que tenga definido el mismo atributo id que el declarado en el selector de la regla. Debido a que en un documento HTML no pueden existir dos elementos CSS, su naturaleza solamente le permite seleccionar un único elemento.

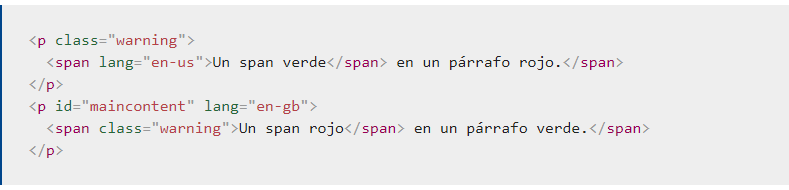




### Selector universal

Esta representado por un asterisco y tiene como función seleccionar todos los elementos del documento HTML.





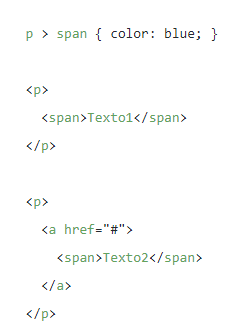
### Selector de atributo

Los selectores de atributos nos permiten seleccionar los elementos HTML en función del valor o contenido de sus atributos. Seis son los tipos de selectores de atributos que existen:

* [att=val]: Selecciona los elementos que tengan definido el atributo att con valor val.
* [att~=val]: Selecciona los elementos que tengan definido el atributo att y que al menos uno de los valores del atributo sea val.
* [att|=val]: Selecciona los objetos que tengan declarado un atributo att y cuyo valor sea una serie de palabras separadas por guiones que obligatoriamente deben comenzar por val.
* [att$=val]: Selector de elementos que tengan definido un atributo att con un valor que finalice con val.
* [att^=val]: Selecciona objetos HTML que tengan un atributo att con un valor que comience por val.
* [att\*=val]: Selecciona los elementos HTML que tengan definido un atributo att cuyo valor contenga el termino val en cualquier posición.

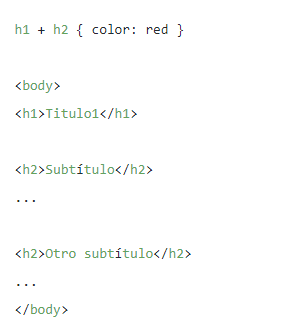
### Selector de hijos

Tiene un funcionamiento semejante al selector descendente, con la diferencia de que el selector de hijos sol escoge aquellos elementos que son descendientes directos de otros elementos. La forma de definir este tipo de selectores es separando al hijo del padre con el carácter *>*.



### Selectores adyacentes

El selector adyacente posibilita seleccionar aquellos elementos que en el código HTML están declarados inmediatamente después de otros elementos.



# Herencia y cascada

Los mecanismos de herencia y cascada son, junto a los selectores, los tres pilares básicos en los que se fundamentan el diseño Web con CSS.

## Herencia

La herencia es una característica de las hojas de estilos por la cual las propiedades de las reglas de estilo asignadas a un elemento HTML son heredadas de forma automática por los elementos descendientes de este.

Es importante destacar que no todas las propiedades CSS son heredables, para saber cuales lo son y cuales no, tendremos dos opciones: la conocida como “prueba y error” o bien consultar la especificación CSS.

La herencia de una propiedad se anulará siempre a favor de la regla CSS que sea mas especifica o concreta, siempre y cuando esta regla tenga definida la misma propiedad.

## Cascada

El conocido mecanismo de cascada, que es el que decide el resultado final en caso de colisión entre reglas CSS. Básicamente, estos conflictos suelen producirse por dos razones fundamentales:

* **Herencia múltiple**: Un elemento hereda la misma propiedad desde varios elementos padres o ascendentes. En este caso, ganaría el estilo aplicado por el padre más cercano.
* **Conflicto de selectores**: Surge de intentar modificar la misma propiedad de un elemento desde diferentes reglas CSS formadas por diferentes selectores. Los conflictos de selectores se resuelven basándose en unos pesos o niveles de especificidad que los navegadores asignan a cada selector, el selector con mayor peso será el que haga predominar la propiedad de su regla CSS.

# Posicionamiento CSS

El correcto funcionamiento en la pantalla de los diferentes elementos que conforman una página web es sin dudad alguna la parte mas complicada del diseño Web con hojas de estilos. Estos mecanismos o funcionalidades se dividen en tres grupos: modelo de cajas en CSS, el comportamiento de los elementos y el posicionamiento de los elementos.

## El modelo de cajas

Se conoce como modelo de cajas al tratamiento que el estándar CSS le da a todos los elementos de un documento HTML. Este tratamiento consiste en interpretar cada elemento como una caja rectangular y dentro de cada caja su contenido. Por defecto, la estructura de estas cajas es transparente al usuario, únicamente es visible el contenido de cada etiqueta o elemento. Las hojas de estilos disponen de varias propiedades que nos permiten modificar y personalizar las cajas dependiendo de nuestras necesidades. Estas propiedades son:

* *Margin*: Tamaño del margen externo de la caja, es la separación existente entre la caja actual y el resto de las cajas.
* *Padding*: Relleno o margen interno, espacio existente entre el borde interno de la caja y su contenido.
* *Background*-color: Color de fondo de la caja.
* *Background*-imagen: URL de la imagen que se mostrara como fondo de la caja.
* *Border*-width: Define el tamaño del grosor del borde de la caja.
* *Border*-style: Define el estilo de la línea o borde de la caja.
* *Borde*-color: Establece el color del borde de la caja.
* *Overflow*: Especifica de que forma se comporta una caja cuando su contenido excede el tamaño de la propia caja. Tiene cuatro valores posibles:
  + *Visible*: Si el contenido de una caja supera el tamaño de la caja, el contenido se mostrará como tal, superando el limite de la caja. Es el valor que tiene por defecto cualquier elemento.
  + *Hidden*: Solamente se mostrará el contenido que no supere el tamaño de la cja.
  + *Scroll*: El navegador mostrara una barra de desplazamiento para acceder al contenido que supere el límite de la caja.
  + *Auto*: Su funcionamiento depende principalmente del navegador, aun así, normalmente la caja se mostrará como tal mientras el contenido no supere el tamaño de esta y aparecerán los botones de desplazamiento de forma automática en cuanto el contenido del elemento supere las dimensiones de la caja.

En la especificación CSS, existen una serie de propiedades conocida como *shorthands*, muy útiles para englobar dentro de una misma propiedad el valor de varias propiedades individuales de un elemento.

## Comportamiento de los elementos

Según la especificación del W3C, los elementos HTML pueden ser clasificadas basándose en su estructura en tres grupos principalmente: básicos. De cabecera y de cuerpo. Los elementos básicos serian aquellos que representan la estructura y cimientos del documento HTML:<html>, <head> y <body>. Los elementos de cabecera son los que recogen información acerca del documento: <title>, <meta> y <link> entre otros. Los elementos de cuerpo son los que estructuran y dan contenido al documento. Dentro de este ultimo grupo y dependiendo del comportamiento de cada elemento según su contenido, existe otra clasificación en dos grandes grupos: elementos de bloque (block) y elementos en línea (inline).

### Elementos de bloque

Los elementos de bloque son aquellos que por defecto ocupan todo el ancho de la pagina y siempre comienzan en una nueva línea. Características:

* Siempre comienzan en una nueva línea.
* Ocupan todo el ancho de la pagina independientemente del contenido que posean.
* Pueden contener otros elementos de tipo bloque o en línea.
* La altura varía en función del contenido.
* La anchura y la altura se pueden modificar con el uso de CSS.

Los elementos de bloque más comúnmente usados en HTML son <div>, <fieldset>, <form>, <h1-h6>, <ol>, <p>, <table> y <ul>.

### Elementos en línea

Los elementos en línea únicamente ocuparan el espacio necesario para mostrar su contenido. Características:

* Solo ocupan el ancho necesario para mostrar su contenido.
* No tienen por que comenzar en una nueva línea.
* Pueden permanecer en la misma línea junto con otros documentos.
* Únicamente pueden contener otros elementos en línea.
* Su anchura y su altura dependen del contenido, no son configurables con reglas CSS.

Los elementos en línea son <a>, <br>, <cite>, <img>, <input>, <label>, <select>, <span>, <strong> y <textarea>.

## Posicionamiento de los elementos

El posicionamiento inicial de los elementos en la pantalla esta directamente relacionado con el grupo al que pertenezca cada elemento, block o inline. Atendiendo a esta clasificación, el navegador será el que se ocupe de hacer la distribución de los elementos de forma automática.

Para modificar o personalizar esta distribución inicial mediante hojas de estilos, tendremos que hacer uso de las siguientes propiedades: *position*, *float*, *top*, *left*, *right* y *bottom*.

* *Position*: Define el posicionamiento de los elementos en una pagina web. Dependiendo de este posicionamiento, los valores de top, left, right y bottom funcionaran de una forma u otra. Cinco son los valores: *static, relative, absolute, fixed e inherit.*
* *Float*: Define un tipo de posicionamiento especial que nada tiene que ver con los anteriores, pero muy útil a la hora de desplazar elementos a los externos de la pantalla. Los valores son: *left, right, none e inherit*.

### Posicionamiento estático o normal

Es el posicionamiento por defecto. La situación de los elementos en pantalla será la que defina el flujo del documento y el tipo de elemento, block o inline. Normalmente, si los elementos son de tipo bloque, estos se mostrarán uno debajo de otro, siguiendo el orden en el que fueron definidos dentro del documento HTML. En este posicionamiento solo usaremos las propiedades heigth y Width para asigna la altura y la anchura respectivamente de los elementos.

### Posicionamiento relativo

En el posicionamiento relativo, el desplazamiento experimentado por un elemento se realizará en función de su posición original. Esta es la posición normal o estática que tiene por defecto un elemento HTML. La forma de definir este posicionamiento en una regla CSS es position: relative.

### Posicionamiento absoluto

En el posicionamiento absoluto, el desplazamiento efectuado por un elemento se lleva a cabo tomando como referencia el primer elemento padre o contenedor que no este posicionado de forma de estática. En el caso que no exista un elemento padre con tales características, el posicionamiento se realizara basándose en la propia ventana del navegador.

Al contrario que el posicionamiento relativo, el elemento abandona el flujo original de la pagina y el resto de los elementos modifican su posición para ocupar el sitio abandonado.

### Posicionamiento fijo o fixed

Se trata de un posicionamiento especial de posicionamiento absoluto, en el cual el desplazamiento efectuado por un elemento se llevará siempre a cabo tomando como referencia la ventana del navegador. En el posicionamiento fijo, el elemento contenedor siempre será la propia ventana del navegador, además, la ubicación de una caja posicionada de forma fija es inamovible, su posición no variara, aunque lo haga la ventana del navegador.

### Posicionamiento inherit

Al definir un elemento con posicionamiento inherit le estamos indicando al navegador que herede esa propiedad de su contenedor más inmediato. Se trata de un tipo de posicionamiento aún menos utilizado que el anterior y que no está soportado por versiones de Internet Explorer inferiores a 9.

### Elementos flotantes

Cuando se define un elemento con posición flotante, este elemento deja de pertenecer al flujo normal de la pagina y se desplaza (flota) hacia el extremo izquierdo o derecho del documento, según le hayamos especificado en la propiedad float. Estos son los posibles valores de la propiedad float:

* Left: Desplaza de forma flotante un elemento hacia el lugar mas a la izquierda posible.
* Right: Desplaza un elemento hacia el lugar mas a la derecha posible.
* None: El elemento no flota hacia ningún lado y se mantiene en su posición original.
* Inherit: Hereda el valor de esta propiedad de su padre o contenedor más cercano.

Los elementos posicionados de forma flotante pueden solapar al resto de elementos que no se encuentren posicionados de la misma forma. Esto ocurre así porque, al abandonar el flujo normal de la página, los elementos flotantes no tienen en cuenta el resto de los elementos, de ahí el solapamiento del que estamos hablando.

#### La propiedad clear

Su función es la de indicar que lado de la caja de un elemento no puede ser adyacente a una caja flotante. Es decir, impide que otros elementos fluyan alrededor de la caja posicionada flotante, se trataría de forzar a los elementos a ubicarse debajo de cualquier elemento flotante. Sus propiedades son:

* *Left*: Este valor provoca que el elemento en cuestión efectúe un desplazamiento hacia abajo hasta encontrar una línea en la que no exista ningún elemento flotante en su lado izquierdo.
* *Right*: El efecto es similar al anterior, salvo que en esta ocasión lo que se busca es una línea donde no exista ninguna caja flotante en su lado derecho.
* *Both*: El elemento desciende hasta encontrar un lugar donde no exista ningún elemento flotante en sus lados derecho e izquierdo.
* *Inherit*: Toma el valor de esta propiedad de su padre o contenedor más inmediato.