6. CSS3.

Las hojas de estilo en cascada (CSS, Cascading Style Sheets) son un lenguaje para definir el aspecto, la presentación y la posición que tendrán los diferentes elementos que componen una página web.
La importancia de las hojas de estilo ha ido creciendo hasta convertirse en un elemento esencial para el diseño web.

CSS3 es la última evolución del lenguaje de las Hojas de Estilo en Cascada (Cascading Style Sheets), y pretende ampliar la versión CSS2.1. Trae consigo muchas novedades altamente esperadas, como las esquinas redondeadas, sombras, gradientes , transiciones o animaciones, y nuevos layouts como multi-columnas, cajas flexibles o maquetas de diseño en cuadrícula (grid layouts).

Evolución de las hojas de estilo: Las hojas de estilo surgen alrededor de 1970 pero su auge se produce con Internet.
Durante muchos años la ausencia de un estándar provocó que existieran diferentes formas de establecer reglas de estilo y por tanto diferencias de visualización entre navegadores.
Al igual que HTML, el encargado de realizar una especificación oficial es el W3C.

CSS1
La primera versión se publica en 1996 partiendo del trabajo de Bert Bos.
CSS1 incluye formatos de texto, párrafo, márgenes, listas, tamaños de imagen, etc.
Esta primera versión tenía bastantes problemas y la W3C decidió crear un grupo de trabajo independiente para CSS.

CSS2
La segunda versión se publica en 1998.
CSS2 amplía la versión anterior incluyendo posicionamiento (capas), y tipos de medios que permiten establecer diferentes estilos en función del dispositivo (pantallas, impresoras, lectores digitales, etc).
Posteriormente es revisada con CSS 2.1 que es la versión más utilizada y la de mayor compatibilidad en la actualidad.

CSS3
Se lleva trabajando en la tercera versión desde 1998 y es ahora cuando poco a poco se va convirtiendo en el nuevo estándar.
En CSS3 la especificación se ha separado en módulos que aportan nuevas funciones a CSS 2.1, manteniendo retrocompatibilidad.
De los más de 50 módulos, sólo 3 (Selectores, Color y Espacios de nombres) son recomendaciones oficiales, el resto se encuentran en desarrollo.

Las partes experimentales son particulares para cada navegador y deberían ser evitadas en entornos de producción, o usadas con extrema precaución, ya que tanto la sintaxis como la semántica pueden cambiar en el futuro.

Hojas de Estilo en Cascada (del inglés Cascading Style Sheets) o CSS es el lenguaje utilizado para describir la presentación de documentos HTML o XML, esto incluye varios lenguajes basados en XML como son XHTML o SVG. CSS describe como debe ser renderizado el elemento estructurado en pantalla, en papel, hablado o en otros medios.

Las hojas de estilo CSS son un conjunto reglas que enumeran en un fichero .css y que describen el aspecto que deben tener los diferentes elementos HTML de una página.

> *Lo interesante de esto es que funcionan con una filosofía de patrones o plantillas, es decir, no es necesario especificar cada uno de los elementos, sino que se pueden definir reglas como estas dos:*
>
> *"Los títulos de nivel 1 y 2 han de ser de color negro y un tamaño de fuente de 16 y 14 pixeles respetivamente."*
> *"El texto de los párrafos están alineados a la izquierda, tienen un tamaño de fuente de 12 pixeles y un color gris oscuro."*
> *A modo de comparación, si dominas el uso de estilos en Microsoft Word, verás que esto se parece mucho al concepto de estilo en Word, aunque las CSS son infinitamente más potentes que Word en todas sus posibilidades.*

CSS (Hojas de Estilo en Cascada) es el código que usas para dar estilo a tu página web. CSS es uno de los lenguajes base de la Open Web y posee una especificación estandarizada por parte del W3C. Desarrollado en niveles, CSS1 es ahora obsoleto, CSS2.1 es una recomendación y CSS3, ahora dividido en módulos más pequeños, está progresando en camino al estándar.

▪ Ventajas de las hojas de estilo:
La principal ventaja es la separación entre presentación y contenido que se traduce en:

- Disminuye el código a escribir.
- Facilita la generación de código y su mantenimiento.
- Mejora la legibilidad de los documentos.

Vincular los estilos CSS a páginas HTML

Como HTML, CSS no es realmente un lenguaje de programación. Es un lenguaje de hojas de estilo, es decir, te permite aplicar estilos de manera selectiva a elementos en documentos HTML.

Por ejemplo, para seleccionar todos los elementos de párrafo en una página HTML y volver el texto dentro de ellos de color rojo, has de escribir este CSS:

```
p {
  color: red;
}
```

Pero aun debemos aplicar el CSS a tu documento HTML, de otra manera el estilo CSS no cambiará cómo tu navegador muestra el documento HTML.
En el archivo html index.html se debe indicar, dentro del head (es decir, entre las etiquetas <head> y </head>)

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

Para **vincular los estilos CSS a páginas HTML** existen básicamente 3 opciones:

- Etiqueta HTML concreta.
- Elemento style en la cabecera (head)
- Archivo externo.

1. Hojas de estilo CSS inline (Etiqueta HTML concreta)

<etiqueta style=”propiedad: valor; propiedad: valor; …..”>

El estilo se incluye dentro de la etiqueta del elemento a tratar, añadiendo como “atributo” style=”estilos a aplicar”.
Dado que se incluye como regla de etiqueta no hay que especificar selector, ya que elemento sobre el que se aplica ya está identificado.
Esta regla tiene prioridad sobre el resto de vinculaciones de estilo.

La primera opción consiste en usar el atributo “style” en un elemento HTML tal como se puede ver aquí:

```
<h1 style=”font-family:Verdana; color:red”>Título de nivel 1</h1>
```

En este caso, no hay selector puesto que no hay elemento que seleccionar, es el que es. Por tanto, se especifican simplemente las propiedades a aplicar al elemento en cuestión. Este tipo de reglas tienen mayor prioridad de todas las reglas.

Se suele recomendar encarecidamente evitar esta manera de vincular estilos CSS puesto que al tener que aplicarse a cada elemento individualmente, implican un esfuerzo y mantenimiento disparatado.

No obstante, hay casos de uso muy concretos en los que son prácticas. Por ejemplo, en un widget sencillo de WordPress que permita configurar directamente ciertas propiedades CSS desde su interfaz de configuración puede ser una buena opción implementar estas propiedades con esta técnica.

2. Hojas de estilo CSS internas (Elemento style en la cabecera (head))

```
<head>
    <style type="text/css">
        @media=”all” {
            selector { propiedad: valor; propiedad: valor; … }
        }
    </style>
</head>
```

La segunda opción es escribir las reglas dentro de un elemento <style> dentro de esta etiqueta que, a su vez, se encontraría dentro del elemento <head> de la página HTML.

En este caso el código CSS se inserta dentro de la cabecera del documento (head) por medio de la etiqueta style y afecta a todos los elementos del selector indicado que aparezcan en el documento.

El elemento style utiliza dos **atributos**:

- type: siempre con el valor "text/css".
  Si apareciera otro lenguaje se podría indicar.
- media: tipo de dispositivo al que se aplican los estilos:
  all (todos, opción por defecto)
  screen (pantallas color)
  tty (pantallas texto)
  tv, projection, handhell (dispositivos manuales)
  print (impresora),
  braille

Obviamente la desventaja es que en este caso las reglas sólo son visibles para esa página concreta en las que se han declarado y no se pueden usar para otras páginas.

3. Hojas de estilo CSS externas (Archivo externo)

```
<head>
    <link href="estilos/style.css" rel="stylesheet" type="text/css">
</head>
```

Esta opción es la más usada y consiste en crear uno o varios ficheros con extensión ".css" en los cuales se declaran las reglas CSS.

Para utilizar un archivo css externo utilizaremos la etiqueta link con los **atributos**:

- href: ruta del archivo css (url).
- rel: con el valor "stylesheet".
- type: siempre con el valor "text/css".
  Si apareciera otro lenguaje se podría indicar.
- media: tipo de dispositivo al que se aplican los estilos:
  all (todos, opción por defecto)
  screen (pantallas color)
  tty (pantallas texto)
  tv, projection, handhell (dispositivos manuales)
  print (impresora),
  braille

La manera de vincular estas reglas a una página HTML consiste en referenciar el fichero .css desde esa página. De este modo se puede crear un fichero global para una web entera y usarlo en todas las páginas que se quiera.
En este caso, la declaración también se incluye dentro del elemento <head> y tiene el siguiente aspecto:

```
<link rel="stylesheet" type="text/css" href="/css/estilos-1.css" />
<link rel="stylesheet" type="text/css" href="/css/estilos-2.css" />
```

Aquí hay que tener también en cuenta que en el caso de usar varios ficheros el orden de inclusión marca precedencia en el orden inverso. En el caso de existir un conflicto por haber diferentes reglas con igual selector en varios ficheros, las reglas de los ficheros siguientes sobreescriben los anteriores.
Es decir, si en estilos-1.css tenemos una regla como:

```
p a {
        text-decoration: underline;
}
```

Y luego en estilos-2.css otra que dice:

```
p a {
        text-decoration: none;
        color: blue;
}
```

La propiedad "text-decoration" genera un conflicto que se resuelve dando prioridad a la propiedad de estilos-2.css.

▪ **Preferencias de Estilos**:
Cuando en un mismo documento se aplican estilos de varias formas será más preferente la regla cuanto más restrictiva sea, es decir:

1) Estilos definidos en una etiqueta concreta mediante el atributo style.
2) Estilos que proceden de la etiqueta style de la cabecera del documento.
3) Estilos definidos en un archivo externo enlazados con la etiqueta link.

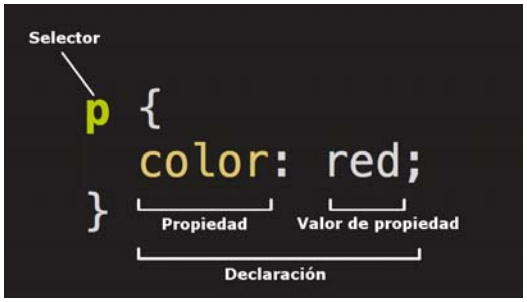
- Dentro de cada una de las formas, en caso de repetición se aplicará la última regla.
- No obstante podemos establecer una regla como prioritaria con el atributo !important.

```
Selector {propiedad: valor !important; ….}
```

Una cosa que notarás sobre la escritura de CSS es que trata mucho sobre cajas (ajustando su tamaño, color, posición, etc). Puedes pensar en la mayoría de los elementos HTML de tu página como cajas apiladas una sobre la otra.

Conceptos Básicos CSS:

Anatomía de una regla CSS:



La sintaxis de CSS es muy sencilla, su dificultad radica en que es muy extenso.

La estructura completa es llamada regla predeterminada (pero a menudo "regla" para abreviar). Nota también los nombres de las partes individuales:

▪ Selector

El elemento HTML en el que comienza la regla. Esta selecciona el(los) elemento(s) a dar estilo (en este caso, los elementos p).
Permite indicar a qué elemento o conjunto de elementos se les aplica el formato
Para dar estilo a un elemento diferente, solo cambia el selector.

▪ Declaración

Cada regla está compuesta de dos elementos: propiedad:valor de propiedad.
Una sola regla como color: red; especifica a cuál de las propiedades del elemento quieres dar estilo.

▪ Propiedades

Maneras en las cuales puedes dar estilo a un elemento HTML. (por ejemplo, color es una propiedad del elemento p.)
Característica de formato a modificar, debe ser un nombre reconocido por CSS.
En CSS, seleccionas que propiedad quieres afectar en tu regla.

▪ Valor de la propiedad

A la derecha de la propiedad, después de los dos puntos (:), tenemos el valor de la propiedad, para elegir una de las muchas posibles apariencias para una propiedad determinada (por ejemplo valor red para color).
Nuevo valor para la propiedad indicada.

Nota: Podemos incluir comentarios (significado o la finalidad del estilo) entre los símbolos /* y */.

▪ Otras partes importantes de la sintaxis:

– Cada una de las reglas (aparte del selector) deben estar encapsulada entre corchetes ({}).

- Dentro de cada declaración, debes usar los dos puntos (:) para separar la propiedad de su valor.
- Dentro de cada regla, debes usar el punto y coma (;) para separar una declaración de la siguiente.

Así que para modificar varios valores de propiedad a la vez, solo necesitas escribirlos separados por punto y coma (;).

Seleccionando varios elementos

También puedes seleccionar varios elementos y aplicar una sola regla a todos ellos. Incluye varios selectores separados por comas (,).

Por ejemplo:

```
p,li,h1 {
  color: red;
}
```

Ejemplo:

Se pueden establecer varias reglas para un mismo selector:

```
h1 { font-family: Verdana; }
h1 { color: red; }
```

Y también agrupar reglas de varios selectores cuando sean compartidas:

```
h1, h2 { font-family: Verdana; color: red; }
```

Diferentes tipos de selector

Existen muchos tipos diferentes de selectores. Antes, solo vimos los selectores de elementos, los cuales seleccionan todos los elementos de un tipo dado en los documentos HTML. Sin embargo podemos hacer selecciones más específicas que esas.

Selectores más comunes:

Nombre del selector	Qué selecciona	Ejemplo
Selector de elemento (llamado algunas veces selector de etiqueta o tipo)	Todos los elementos HTML del tipo especificado.	P Selecciona <p>
Selector de identificación (ID)	El elemento en la página con el ID especificado (en una página HTML dada, solo se permite un único elemento por ID).	#my-id Selecciona <p id="my-id"> y <a id="my-id">
Selector de Clase	Los elementos en la página con la clase especificada (una clase puede aparecer varias veces en una página).	.my-class Selecciona <p class="my-class"> y <a class="my-class">
Selector de atributo	Los elementos en una página con el atributo especificado.	img[src] Selecciona <img src="myimage.png"> pero no <img>

Selector de Pseudo-clase	Los elementos especificados, pero solo cuando esté en el estado especificado, por ejemplo cuando el puntero esté sobre él.	a:hover Selecciona <a>, pero solo cuando el puntero esté sobre el enlace.

La cascada: herencia, sobreescritura y conflictos de estilos.

El nombre de hojas de estilo en "cascada" no es casual. Expresa que los estilos que especifican con reglas se pueden heredar de una manera jerárquica.

Es decir, veamos esta regla:

```
body {
        font-family: Arial;
}
```

Aquí estamos diciendo que la etiqueta <body> que es la que envuelve el contenido de cualquier página web tenga un tipo de letra "Arial". Esto no tendría mucho sentido si no fuera porque gracias a la capacidad de herencia de las reglas de este modo por defecto cualquier elemento hijo como un título o un párrafo va a "heredar" ese estilo, salvo que especifique lo contrario como, por ejemplo, en esta regla:

```
p {
        font-family: Verdana;
}
```

En esta última regla aplica el principio de "especificidad". En principio, se plantearía un conflicto entre la regla general de <body> con lo que dice la regla de <p>, pero se resuelve fácilmente puesto que se aplica la regla más específica y referirse a un párrafo es más específico que referirse a "los elementos hijos que pueden dentro de <body>".

Igualmente, una regla con un selector "p a" (enlace dentro de un párrafo) tendría precedencia sobre una regla con un selector "a" que se refiere a un enlace a secas.

▪ Herencia de estilos:

En CSS tenemos etiquetas que contienen a otras etiquetas. En esos casos a las etiquetas interiores se les aplican todas las características de formato de las etiquetas que las contienen (heredan su estilo).

Por ejemplo:

```
p { font-size: 12px; color:blue; }
em { font-size: 14px; }

<p>Miguel de Cervantes: <em>El Quijote</em></p>
```

(El Quijote aparecerá en color azul y tamaño 14px.)

Propiedades Básicas.

1. **Maquetación básica**:

width	Ancho de un elemento.
height	Alto de un elemento.
vertical-align	Alineamiento vertical dentro de un elemento.
margin	Espacio que se añade entre el elemento y sus vecinos. Se puede diferencia por lado (arriba, abajo, izquierda, derecha).
padding	Relleno interior que se añade en los bordes del elemento. A diferencia de margin, cuenta para el tamaño del elemento.
float	Mueve el elemento todo lo posible hacia el lado indicado. Esta propiedad se usa en el posicionamiento flotante de CSS. El tema del posicionamiento en CSS no es trivial y conviene estudiar cómo funciona antes de usar esta propiedad.

2. **Fuentes y texto:**

font-family	Tipo de letra. (Ver tipografías)
font-size	Tamaño de letra. (*N*px tamaño en pixeles)
font-weight	Peso (normal, negrita, …).
font-style	Estilo (normal, cursiva, …).
text-decoration	"Decoraciones" como subrayado, tachado, etc.
text-align	Alineación del texto (izquierda, derecha, etc.). (center, left, right)
text-transform	Mostrar un texto en mayúsculas, minúsculas o la primera letra de cada palabra en mayúsculas.

3. **Color y fondos:**

color	Color del elemento. Se puede especificar en diferentes formatos como palabras predefinidas (red, green, etc.) RGB o como valor hexadecimal.
background-color	Color del fondo del elemento.
background-image	Permite especificar una imagen de fondo.
background-repeat	Permite usar una imagen a modo de mosaico en diferentes modalidades.
box-shadow	Crear un efecto de sombra para un elemento.

4. Listas:

list-style-image	Usar la imagen especificada como viñeta para la lista.
list-style-type	Diferentes estilos de viñetas y estilos de numeración para elementos de lista.

5. Bordes:

border	Añade un borde a un elemento y establece algunas propiedades (grosor, estilo de línea, etc.)
border-color	Color del borde.
border-style	Diferentes estilos para el borde (sólido, puntos, etc.)
border-radius	Permite crear esquinas redondeadas para un elemento.