



Diseño de interfaces web

Unidad 2: Uso de estilos: creación de interfaces web utilizando estilos

ÍNDICE

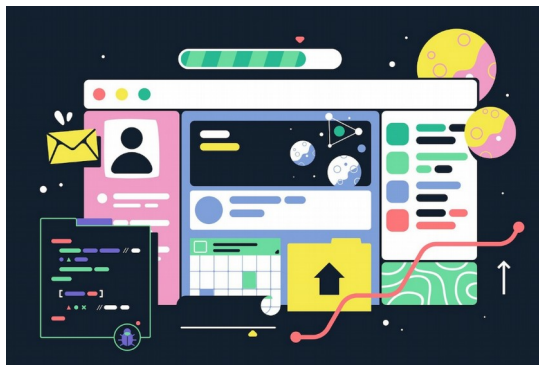
INTRODUCCIÓN.....	3
OBJETIVOS / CAPACIDADES.....	4
PROYECTO DE LA UNIDAD.....	5
1. INTRODUCCIÓN A HOJAS DE ESTILO EN CASCADA (CSS, CASCADING STYLE SHEET).....	6
2. ESTILOS EN LÍNEA BASADOS EN ETIQUETAS, CLASES E IDENTIFICADORES.....	8
3. SELECTORES. AGRUPACIÓN Y ANIDAMIENTO DE SELECTORES. NOTACIÓN. SINTAXIS. ATRIBUTOS.....	10
4. MODELO DE CAJAS. MÁRGENES, RELLENO Y BORDES.....	14
Cuestionario.....	17
5. POSICIONAMIENTO DE ESTILOS. MAQUETACIÓN.....	18
6. ELEMENTOS: COLORES DE FONDO, TEXTOS, ENLACES, LISTAS, TABLAS, CAJAS, POSICIONAMIENTO, VISIBILIDAD, ALINEAMIENTO, IMÁGENES.....	21
7. CREAR Y VINCULAR HOJAS DE ESTILO. ELEMENTOS DE LAS HOJAS DE ESTILO.....	23
8. CREAR Y VINCULAR HOJAS DE ESTILO EN CASCADA EXTERNA.....	26
Cuestionario.....	29
9. PSEUDÓNIMOS.....	30
10. HOJAS DE ESTILO AUDITIVAS.....	35
Cuestionario.....	38
RESUMEN.....	39
RECURSOS PARA AMPLIAR.....	40
BIBLIOGRAFÍA.....	41
GLOSARIO.....	42

INTRODUCCIÓN

Durante este módulo nos centraremos en los **estilos** que podemos aplicar a las páginas web.

Veremos una pequeña introducción de que son las **hojas de estilo en cascada**, o **CSS**, para después ampliar los conocimientos.

Mostraremos las diferentes maneras que tenemos de dotar de estos **estilos** a una página web, haciendo hincapié en sus **diferencias, ventajas e inconvenientes**.



Finalmente, veremos lo que son las **hojas de estilo**, y como podemos usarlas de manera **interna** o **externa** a nuestra página web, con lo que ello comporta.

Durante toda la unidad haremos uso de **código HTML**, el cuál estudiaremos en otros módulos del ciclo, por lo que se recomienda haberlos cursado previamente, o al menos hacerlos en paralelo, para entender mejor el contenido.



OBJETIVOS / CAPACIDADES

En esta unidad de aprendizaje, las capacidades que más se van a trabajar son:

- ✓ Utilizar lenguajes de marcas y estándares web, asumiendo el manual de estilo, para desarrollar interfaces en aplicaciones web.



PROYECTO DE LA UNIDAD

Antes de empezar a trabajar el contenido, te presentamos la **actividad** que está relacionada con esta unidad de aprendizaje. Se trata de un **caso práctico** basado en una **situación real** con la que te puedes encontrar en tu puesto de trabajo. Con esta actividad se evaluará la puesta en práctica de los **criterios de evaluación** vinculados al resultado de aprendizaje que se trabaja en esta unidad. Para realizarla deberás hacer lo siguiente: lee el enunciado que te presentamos a continuación, dirígete al área general del módulo profesional, concretamente a la actividad de evaluación que se encuentra dentro de esta unidad, allí encontrarás todos los detalles sobre fecha y forma de entrega, objetivos... A lo largo de la unidad irás adquiriendo los conocimientos necesarios para ir elaborando este proyecto.

Enunciado:

Creando el currículum perfecto.

Finalmente, tu profesor ha decidido que tienes el nivel suficiente para realizar la entrevista en la empresa de prácticas. Pero, aparte de las actividades que has ido realizando, la empresa pide un **currículum** para la entrevista, por lo que necesitas trabajar en ello.

Has encontrado por internet un **diseño** que te gusta ([UD2_eval.png](#)), y te gustaría recrearlo, pero por desgracia sólo has podido conseguir el **código HTML** ([UD2_eval.html](#)) y la **imagen** ([UD2_eval.jpg](#)). Así que necesitas **crear los estilos para maquetarlo** como en el diseño.

Se pide **crear el archivo CSS** correspondiente, y **validarlo** mediante el sistema validador de **W3C** (<http://www.css-validator.org/validator.html.es>). Además, es importante que **modifiques el HTML para que contenga tus datos**.

Para tener en cuenta a las personas con deficiencias visuales, es importante que los **títulos** se lean con un **volumen superior**.

1. INTRODUCCIÓN A HOJAS DE ESTILO EN CASCADA (CSS, CASCADING STYLE SHEET)

A lo largo de esta unidad nos centraremos en las **hojas de estilo en cascada**, conocidas comúnmente como **CSS**. Pero antes veamos exactamente de qué se tratan.



*En otros módulos del ciclo vemos cómo podemos utilizar el **lenguaje de marcas HTML para crear páginas web**, especificando su estructura y diseño. Además, las podemos hacer dinámicas complementando el código con JavaScript, a nivel de cliente.*

Ahora veremos como **CSS** nos puede ayudar y mucho en la parte del **diseño**, no sólo facilitando el proceso, si no haciendo que éste pueda ser **modular**.

Modular hace referencia a que el trabajo se pueda realizar por **partes**, habiendo desarrolladores **HTML** que se encarguen de la **estructura**, desarrolladores que se encarguen del **código JavaScript** y **diseñadores** que se encarguen del **CSS**.



*Las **hojas de estilos en cascada o CSS** corresponden a un conjunto de **reglas** usadas para **definir y crear la presentación** de un documento escrito en HTML.*

Así pues, podemos decir que las **principales ventajas** de la utilización de las **hojas de estilo** son:

- ➔ Permiten separar la **estructura** de las **páginas y su contenido** (en HTML), del **formato del texto** y de la **página** (en CSS).
- ➔ Aporta muchas más posibilidades de **formato y presentación** de una página web.

- ➔ Permite **unificar** el **diseño** de las páginas web del sitio web, para así **definir un estilo una sola vez** y poder aplicarlo tantas veces como se desee.
- ➔ Posibilita la **reutilización** de las **mismas hojas de estilo** para distintos documentos HTML.
- ➔ Son un **estándar de W3C**, que ofrece **compatibilidad** con versión anterior y con los distintos navegadores.

2. ESTILOS EN LÍNEA BASADOS EN ETIQUETAS, CLASES E IDENTIFICADORES

Antes de entrar propiamente en las **hojas de estilo CSS**, las cuales normalmente se crean en archivos separados del HTML, vamos a ver qué son los **estilos en línea basados en etiquetas**, como crearlos y utilizarlos.

Para ello, la idea es utilizar el **atributo style**, que tienen la inmensa mayoría de etiquetas HTML, para poder definir el estilo deseado. Veamos a continuación las **ideas básicas** sobre los **estilos en línea**:



→ Concepto.



DESTACADO

Se denominan **estilos en línea** porque el estilo **CSS** a utilizar se **define de manera conjunta** con la **etiqueta HTML**, en la misma línea.

→ Sintaxis.

Su **sintaxis** será la siguiente:

```
<etiquetaHTML style = "propiedadCSS : valor;" > ... </ etiquetaHTML >
```

Por **ejemplo**, si lo que queremos es crear un **párrafo que tenga el texto de color azul**, haríamos:

```
<p style ="color:blue;"> Párrafo </p>
```



TOMA NOTA

Para conseguir el **estilo** que queremos en cada etiqueta es esencial **dominar la sintaxis del lenguaje CSS**, lo que nos permitirá conocer todas sus posibilidades. Lo iremos viendo en los siguientes capítulos.

Veamos a continuación otro **ejemplo**:

Ejemplo: asignación del color verde a una lista de links

Imaginemos que queremos conseguir que una **lista de links tenga el texto en color verde**. Siguiendo con lo visto, podríamos hacer algo como:

```
<ul style="color: green;">
  <li> <a href="#">Link 1</a></li>
  <li> <a href="#">Link 2</a></li>
  <li> <a href="#">Link 3</a></li>
</ul>
```

Teóricamente estamos **asignando el color verde a toda la lista**, y por lo tanto a cada elemento ``. Pero si ejecutamos el código veremos que esto no es así.

La explicación es sencilla. Los **links**, por defecto, tienen un **estilo propio**: color azul cuando no lo hemos visitado y morado cuando ya lo hemos visitado.

Así que en este caso **prevalece el estilo del link `<a>` respecto al de la lista ``**. Si queremos que cada link se vea en verde, tendríamos que **especificarlo dentro de la etiqueta de cada link**. Por ejemplo:

```
<li> <a style="color: green;" href="#">Link 1</a></li>
```

Así, independientemente de si el link ha sido visitado o no, **siempre sería verde**. Estos comportamientos los entenderemos mucho mejor cuando veamos el sistema de cajas en capítulos posteriores.

3. SELECTORES. AGRUPACIÓN Y ANIDAMIENTO DE SELECTORES. NOTACIÓN. SINTAXIS. ATRIBUTOS



Los **estilos en línea** están bien para empezar o para trabajos que no son demasiado extensos, pero cuando se trata de **proyectos grandes**, no funcionan tan bien.

Comencemos viendo un **ejemplo** en el que los **estilos en línea** pueden no ser la herramienta que necesitamos:

Ejemplo: posibles problemas en el uso de estilos de línea

Imaginemos que tenemos una página en la cual **todos los títulos** `<h1>` han de tener un **mismo estilo**. Para ponerlo fácil, sigamos con el color del texto en verde.

Deberíamos hacer algo así:

```
<h1 style="color: green;"> Título 1 </h1>
```

Parece algo sencillo y sin complicaciones, pero **¿cuál es el problema?** Que cada vez que tengamos que definir un nuevo título `<h1>` tendremos que indicarle que lo queremos en color verde.

Y no solo eso, sino que la situación se nos puede **complicar** aún más. Imagina que nuestro cliente ya no quiere los títulos en verde, si no que ahora los quiere en azul. En este caso, deberíamos **recorrer toda nuestra** página buscando los títulos y haciendo lo siguiente:

```
<h1 style="color: blue;"> Título 1 </h1>
```

Esto, aparte de una **pérdida de tiempo**, puede ser peligroso si nos **saltamos algún título** que permanecerá en verde cuando ya hemos cambiado todos a azul.

¿Te recuerda a algo este problema? Seguramente si dominas los procesadores de texto te darás cuenta de que es algo similar a lo que ocurre con los estilos.

Así que, para **evitar problemas** como el visto en el ejemplo, en CSS tenemos los llamados **selectores**.

La **función de los selectores** es indicar a qué elementos de nuestra página se aplicarán los estilos definidos. Teniendo esto en cuenta, veamos los **diferentes tipos de selector** :

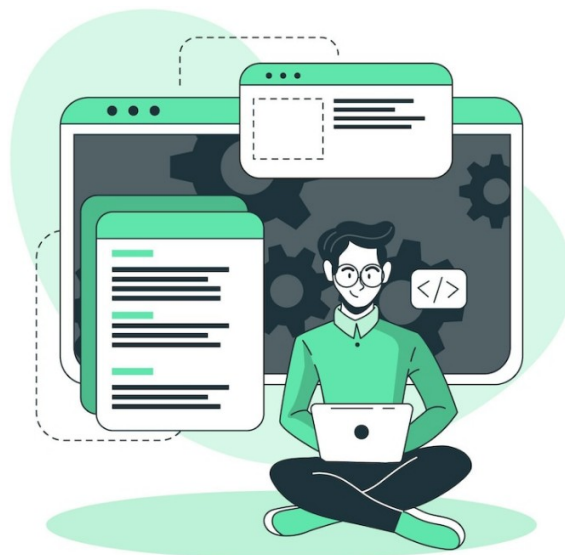
➔ ***** (**selector universal**): permite aplicar un estilo a **cualquier tipo**.

➔ **nombreElemento** (**selector de tipos**): permite aplicar un estilo a un **elemento HTML concreto**.

➔ **.nombreClase** (**selector de clase**): permite aplicar un estilo a los **elementos que utilicen esta clase**.

➔ **#idNombre** (**selector ID**): permite aplicar un estilo a un **elemento que tenga un código de identificación único**.

➔ **selector[atributo]** (**selector de atributo**): permite aplicar un estilo a un **elemento que utilice un atributo específico**.



Así pues, podríamos usar un **selector de tipos** para indicar que todos los títulos `<h1>` sean **verdes**. Veamos cómo podríamos conseguirlo en **dos simples pasos**:

➔ **Paso 1.**

```
h1 {  
  color: green;  
}
```

➔ **Paso 2.**

Con algo tan sencillo, ahora todos nuestros títulos serían verdes **sin necesidad de especificarlo en el propio título**.

```
<h1> Título 1 </h1>  
<h1> Título 2 </h1>
```

Te proponemos visualizar el siguiente **vídeo** para conocer más a fondo los **selectores**:

[Vídeo: ¿cómo usar los selectores?](#)



Visualiza este vídeo en el que hablamos sobre los selectores y cómo usarlos.

Actividad de aprendizaje 1: modificar un código HTML para añadir estilos en línea

Tu profesor está tan contento con los progresos en la primera unidad que te ha propuesto empezar a realizar prácticas en una empresa, pero para ello te pide una serie de trabajos, que podréis enseñar cuando vayáis a hablar con ellos para presentarte.

El primero consiste en **modificar un código HTML dado para añadir una serie de estilos en línea**.

La página resultante debería ser **parecida a la imagen adjunta**:

Lorem ipsum dolor sit amet

Nulla pretium. Sed tempus nunc vitae neque. **Suspendisse gravida**, metus a scelerisque sollicitudin, lacus velit ultricies nisl, nonummy tempus neque diam quis felis. **Etiam sagittis tortor** sed arcu sagittis tristique.

Aliquam tincidunt, sem eget volutpat porta

Vivamus velit dui, placerat vel, feugiat in, ornare et, urna. **Aenean turpis metus**, aliquam non, tristique in, pretium varius, sapien. Proin vitae nisi. Suspendisse **porttitor purus ac** elit. Suspendisse eleifend odio at dui. In in elit sed metus pretium elementum.

Titul de la taula

	Titul columna 1	Titul columna 2
Titul fila 1	Donec purus ipsum	Curabitur <i>blandit</i>
Titul fila 2	Donec purus ipsum	Curabitur blandit
	Titul columna 1	Titul columna 2

Donec purus ipsum, posuere id, venenatis at, placerat ac, lorem. Curabitur blandit, eros sed gravida aliquet, risus justo porta lorem, ut mollis lectus tortor in orci. Pellentesque nec augue.

Fusce nec felis eu diam pretium adipiscing. **Nunc elit elit**, **vehicula vulputate**, venenatis in, posuere id, lorem. Etiam sagittis, tellus in ultrices accumsan, diam nisi feugiat ante, eu congue magna mi non nisl.

Vivamus ultrices aliquet augue. Donec arcu pede, pretium vitae, rutrum aliquet, tincidunt blandit, pede. Aliquam in nisi. Suspendisse volutpat. Nulla facilisi. Ut ullamcorper nisi quis mi.

Entrega el **archivo .html** en el espacio habilitado para ello.

4. MODELO DE CAJAS. MÁRGENES, RELLENO Y BORDES

Anteriormente mencionamos el modelo de cajas, el cual retomamos a continuación para ver de qué se trata y cómo utilizarlo.



Los **navegadores**, de forma automática, crean y colocan cada elemento HTML en una **estructura** que tiene formato de **caja**.

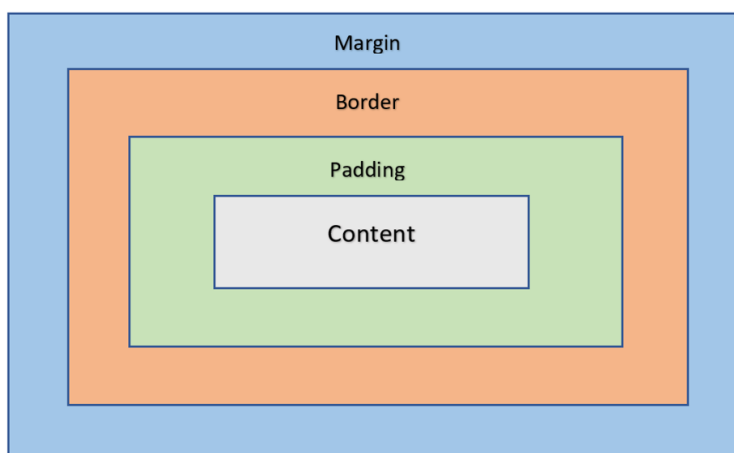
Así, por defecto, la caja de los elementos HTML de tipo **block** ocupa **todo el ancho de la ventana**, y en cambio, la de los elementos **in-line** sólo cogen la **anchura necesaria**.

En principio, la caja de cada elemento **no es visible**, puesto que no muestra **ningún color de fondo ni borde**. Sin embargo, esto se puede modificar con CSS para cambiar sus **dimensiones, colores, borde y posición**.



Pueden modificarse las propiedades de la caja de cualquier elemento HTML, pero generalmente se utiliza el elemento `<div>` para organizar y dar estructura al diseño de las páginas web.

Las **partes que componen una caja** en su orden de visualización desde el punto de vista del usuario son:



➔ **Contenido (content)**: se refiere al **contenido del elemento**; por ejemplo, el texto, una imagen, etc.

- ➔ **Relleno (padding): espacio libre** opcional existente entre el contenido y el borde.
- ➔ **Borde (border): línea** que envuelve el contenido y el relleno.
- ➔ **Margen (margin): separación opcional** existente entre la caja y el resto de cajas adyacentes.

Veamos a continuación un **ejemplo** para conocer más a fondo estos **elementos**:

Ejemplo: creación de un estilo para caja empleando colores, bordes, margen y relleno

En el siguiente **ejemplo** veremos cómo **crear un estilo para una caja** de color amarillo, borde sólido y margen y relleno de 10 píxeles:

```
.estilo1 {  
  background-color: #FBD603;  
  border-style: solid;  
  margin: 10px;  
  padding: 10px;  
}
```

También podemos especificar el **tamaño** de las cajas. En el ejemplo, emplearemos una caja de 300x100 píxeles, con un relleno de 50 píxeles y un borde sólido rojo y de 1 píxel.

```
.estilo2 {  
  width: 300px;  
  height: 100px;  
  padding: 50px;  
  border: 1px solid red;  
}
```

Tras este ejemplo, te proponemos la visualización del siguiente **vídeo**:

Vídeo: explicación del modelo de caja



Visualiza el siguiente vídeo sobre el modelo de caja:
<https://www.youtube.com/embed/O8PGtkO3sjE>

Cuestionario



Lee el enunciado e indica la opción correcta:

¿Qué organización regula la estructura de CSS?

- a. WWW.
- b. W3C.
- c. W3Schools.



Lee el enunciado e indica la opción correcta:

En una caja ¿cómo regulamos la distancia hacia el elemento de encima?

- a. bottom.
- b. left
- c. top.



Lee el enunciado e indica la opción correcta:

¿Cuál es el selector universal?

- a. *
- b. .
- c. #

5. POSICIONAMIENTO DE ESTILOS. MAQUETACIÓN

Siguiendo con el **modelo de cajas**, vamos a ver un concepto asociado a ellas como es el **posicionamiento**.



DESTACADO

*El posicionamiento nos permite **definir la maqueta** de una página web.*

Los navegadores **crean y posicionan de forma automática** todas las **cajas** de los elementos que forman cada página HTML, siguiendo el flujo normal de ésta.

Sin embargo, **CSS permite modificar la posición** en la que se muestra la caja, para así poder conseguir un diseño determinado.

Podemos usar la propiedad **position** de CSS para indicar el **posicionamiento** de cada elemento. En concreto existen **cuatro alternativas**, como son:

- Static.
- Relative.
- Absolute.
- Fixed.

Profundicemos a continuación en las **ideas clave** relacionadas con estas **cuatro alternativas**:

➔ Posicionamiento estático o normal (static).

Es el **comportamiento por defecto**. A la hora de posicionar, sólo se tiene en cuenta si el elemento es en **bloque** o en **línea**, las propiedades **width** y **height** y su **contenido**.



Los contenidos en bloque se muestran uno debajo del otro, y los contenidos en línea se muestran uno detrás del otro.

→ Posicionamiento relativo (**relative**).

Consiste en desplazar una caja respecto a su **posición normal**, mediante las propiedades top, bottom, left y right.

```
.cajaRelativa {  
  position: relative;  
  top: 10px;  
  left: 5px;  
  width: 50px;  
  height: 20px;  
}
```

→ Posicionamiento absoluto (**absolute**).

Se establece el **posicionamiento exacto** respecto al elemento contenedor, también mediante las propiedades top, bottom, left y right.

Las cajas posicionadas de forma absoluta salen del **flujo normal** de la página, lo que provoca que el resto de elementos se muevan.

```
.cajaAbsoluta {  
  position: absolute;  
  top: 70px;  
  left: 100px;  
  width: 50px;  
  height: 20px;  
}
```

→ Posicionamiento fijo (**fixed**).

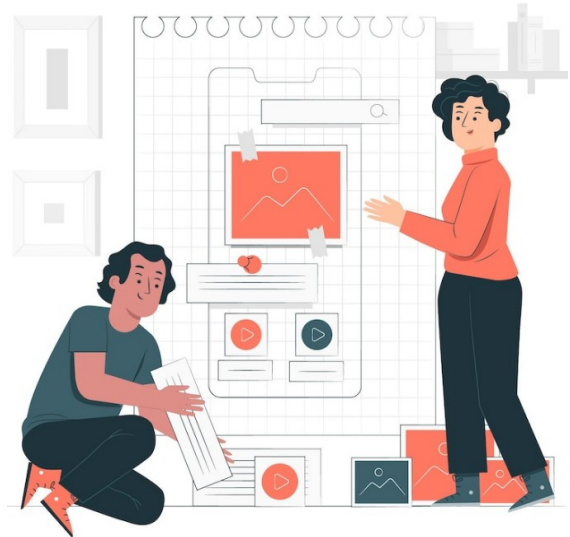
La caja se convierte en un elemento **inamovible**. Es parecido al absoluto, pero el elemento permanece en el mismo sitio, aunque el usuario se mueva por la página.

```
.cajaFija {  
  position: fixed;  
  top: 180px;  
  left: 100px;  
  width: 50px;  
  height: 20px;  
}
```

6. ELEMENTOS: COLORES DE FONDO, TEXTOS, ENLACES, LISTAS, TABLAS, CAJAS, POSICIONAMIENTO, VISIBILIDAD, ALINEAMIENTO, IMÁGENES



Hemos visto ya algunas maneras de aplicar estilos CSS en una página HTML, ya sea en línea o mediante selectores, y también hemos visto algunas de las cosas que podemos modificar, como el color de texto, el color de fondo o los bordes.



Centrémonos ahora en conocer otros **elementos** que podemos modificar mediante **estilos de CSS**:

→ Colores de fondo.

Permite modificar el **color de fondo** de un elemento o caja. Podemos seleccionar el color de entre los **predefinidos del sistema** o mediante **códigos hexadecimales**.

```
background-color: yellow;  
background-color: #FBD603;
```

→ Textos.

Podemos especificar desde los colores del **texto** o de **fondo** hasta el **tipo o tamaño de letra**. Por ejemplo:

```
color: red;  
font: 15px Arial, sans-serif;
```

→ Enlaces.

Tal y como hemos visto en un capítulo anterior, también podemos aplicar estilos a los **enlaces** `<a>`, para especificar cómo será el texto de estos, por ejemplo:

```
a:link {  
  color: red;  
}
```

→ Listas y tablas.

Tanto en listas como en tablas podemos definir un **estado para todo el conjunto** (si lo ponemos en el elemento principal, por ejemplo `<table>`), o para **cada elemento concreto** (por ejemplo `<tr>`), para una fila determinada:

```
<tr class="estilo1">
```

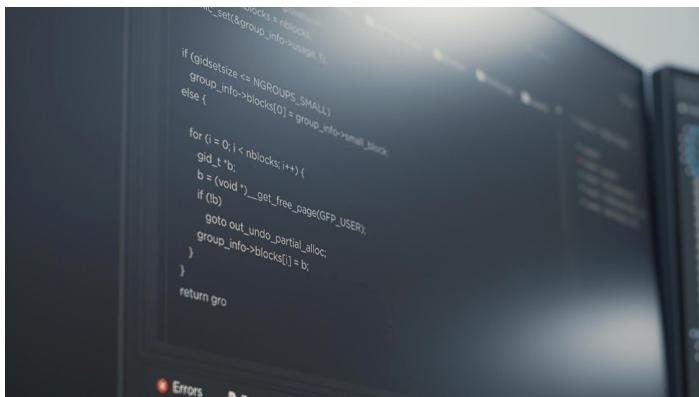
→ Imágenes.

Mediante CSS podemos **modificar aspectos de las imágenes** como el borde, el radio del borde, hacerlas responsive, centrarlas, etc.

```
img {  
  border-radius: 8px;  
}
```


7. CREAR Y VINCULAR HOJAS DE ESTILO. ELEMENTOS DE LAS HOJAS DE ESTILO

Ya conocemos cómo definir los **estilos en línea**, los cuales permiten aplicar diferentes estilos en el propio documento HTML, a la vez que **definimos cada elemento**. Este método tiene sus **ventajas** y **desventajas**, las cuales hemos ido viendo en capítulos anteriores. Sin embargo, es momento de conocer más a fondo las denominadas **hojas de estilo**:



→ Concepto.



DESTACADO

*El método más común para definir los **estilos CSS** es mediante lo que se denominan las **hojas de estilo**.*

*Una **hoja de estilos** consiste en una serie de **reglas** que describen cómo se mostrará el **contenido** de la página web.*

→ Opciones.

A la hora de aplicar los estilos a una página web, tenemos **3 grandes posibilidades**:

- ✓ **Estilos en línea**: son los que ya conocemos.
- ✓ **Hojas de estilos incrustadas**: son las que veremos en este capítulo. Las cuales veremos más detenidamente a continuación.
- ✓ **Hojas de estilos externas**: aquellas que veremos en el siguiente capítulo.

→ Hojas de estilos incrustadas.

Centrándonos, tal y como comentamos, en las **hojas de estilo incrustadas**, podemos decir que éstas también se definen en el propio documento HTML, como los estilos en línea, pero la principal diferencia es que **se agrupan todas en un mismo sitio**.



Para las **hojas de estilo incrustadas** crearemos una **etiqueta** `<style>`, generalmente en la cabecera del HTML, pero podríamos ponerla en cualquier sitio. Dentro de dicha etiqueta podemos crear las diferentes reglas usando, por ejemplo, los **selectores** vistos anteriormente.

Veamos un **ejemplo** sobre cómo **crear un estilo para títulos**:

Ejemplo: crear un estilo para títulos

Si queremos crear un **estilo para los títulos** bastaría con poner el siguiente código en la etiqueta `<head>` del documento HTML:

```
<style>
  h1 {
    color: green;
  }
</style>
```

De igual manera, podríamos **crear todos los estilos necesarios**, siempre dentro de esa etiqueta `<style>`.

Actividad de aprendizaje 2: uso de estilos CSS incrustados

Parece que ya dominas los **estilos en línea**, por lo que tu profesor ha decidido que **colabores** con otro alumno para la siguiente actividad.

En esta ocasión se trata de hacer lo mismo que en la primera actividad, pero esta vez sin usar estilos en línea, sino empleando los **estilos CSS incrustados**.

Es decir, a partir del **documento** [AA2.html](#), tenéis que conseguir, con **estilos CSS**, un resultado parecido a la imagen que veremos a continuación:

Lorem ipsum dolor sit amet

Nulla pretium. Sed tempus nunc vitae neque. **Suspendisse gravida**, metus a scelerisque sollicitudin, lacus velit ultricies nisl, nonummy tempus neque diam quis felis. **Eti**am **sagittis tortor** sed arcu sagittis tristique.

Aliquam tincidunt, sem eget volutpat porta

Vivamus velit dui, placerat vel, feugiat in, ornare et, urna. **Aenean turpis metus**, **aliquam non**, **tristique in**, pretium varius, sapien. Proin vitae nisi. Suspendisse **porttitor purus ac elit**. Suspendisse eleifend odio at dui. In in elit sed metus pretium elementum.

Titul de la taula

	Titul columna 1	Titul columna 2
Titul fila 1	Donec purus ipsum	Curabitur blandit
Titul fila 2	Donec purus ipsum	Curabitur blandit
	Titul columna 1	Titul columna 2

Donec purus ipsum, posuere id, venenatis at, placerat ac, lorem. Curabitur blandit, eros sed gravida aliquet, risus justo porta lorem, ut mollis lectus tortor in orci. Pellentesque nec augue.

Fusce nec felis eu diam pretium adipiscing. **Nunc elit elit**, **vehicula vulputate**, venenatis in, posuere id, lorem. Etiam sagittis, tellus in ultrices accumsan, diam nisi feugiat ante, eu congue magna mi non nisl.

Vivamus ultrices aliquet augue. **Donec arcu pede**, **pretium vitae**, rutrum aliquet, tincidunt blandit, pede. Aliquam in nisi. Suspendisse volutpat. Nulla facilisi. Ut ullamcorper nisi quis mi.

Podéis aprovechar para discutir cómo creasteis los estilos en línea cada uno de vosotros y adaptar el que creáis más conveniente para esta actividad.

Entrega los archivos en el espacio habilitado para ello.

8. CREAR Y VINCULAR HOJAS DE ESTILO EN CASCADA EXTERNA

La última opción que nos queda por ver en cuanto a las hojas de estilo son las denominadas **hojas de estilo externas**.



DESTACADO

Una **hoja de estilo externa** es un documento independiente del documento HTML, en el cual **definimos todos los estilos** de la página web.

Esto nos aporta una serie de **ventajas** respecto a los otros dos métodos vistos anteriormente, como son:

- ➔ **Modularidad**: permite **separar** de mejor manera la **estructura** de la página web de su **diseño**.
- ➔ **Trabajo en equipo**: debido a la **modularidad**, el trabajo en equipo se facilita, ya que cada miembro puede **trabajar en un apartado diferente** de forma más o menos **independiente**.
- ➔ **Reutilización de código**: podemos aprovechar **una misma hoja de estilos** para varias páginas o incluso varios proyectos diferentes. Además, podemos utilizar todos los estilos o bien sólo coger algunos de ellos.

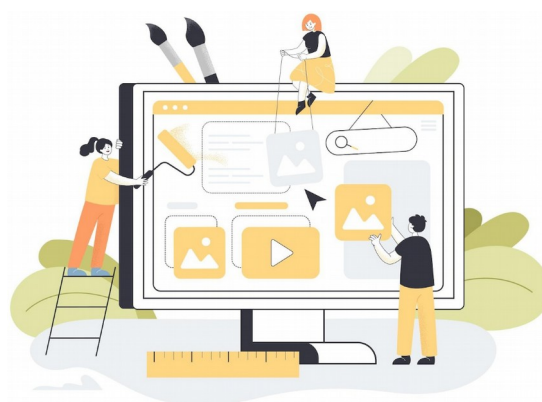
Estas hojas de estilo llevan normalmente la **extensión .css**, aunque podrían trabajar con cualquiera siempre y cuando se **referencie** bien desde el documento HTML.

En este sentido, existen **dos maneras** de asociar estas hojas de estilo al documento HTML:

- Importación.
- Vinculación.

Profundicemos en el conocimiento de estas **dos vías**:

- ➔ **Importar**.



Se trata de añadir una serie de **reglas** en un CSS externo a una etiqueta `<style>`, mediante la sentencia **@import**, indicando la dirección del archivo CSS, que puede ser **local del proyecto** o una **dirección web global**. Esto puede ser:

```
<style>
  @import url(dirección web o local del archivo css);
  p {
    background-color: yellow;
    color: black;
  }
</style>
```

Esta importación haría que **todos los estilos estuvieran en el mismo lugar**, algunos definidos directamente y otros importados. Sin embargo, no vemos claramente la definición de todos.

➔ Vinculación.

Para solucionar esto existe el otro método de asociación, la **vinculación**, en la cual se trata de **asociar todo el documento CSS** que contiene las **reglas**.

De esta manera, **no mezclamos reglas internas y externas**, si no que todas están en el documento CSS.

```
<link href="misEstilos.css" rel="stylesheet" />
```

Para la vinculación usamos la etiqueta `<link>`, indicando la referencia a la hoja de estilos con el atributo **href**, y la relación del documento vinculado con el atributo **rel**.

Te proponemos visualizar el siguiente vídeo para **ampliar la información**:

[Vídeo: asociación entre archivos CSS y HTML](#)



Visualiza este vídeo sobre la asociación entre archivos CSS y HTML.

<https://www.youtube.com/embed/7IaqdnU44rQ>

Cuestionario



Lee el enunciado e indica la opción correcta:

En una hoja de estilos incrustada...

- a. Los estilos se definen en un documento externo.
- b. Los estilos se definen juntos dentro del documento HTML.
- c. Los estilos se definen dentro de cada elemento.



Lee el enunciado e indica la opción correcta:

A la hora de vincular una hoja de estilos externa se usa la etiqueta...

- a. import.
- b. add.
- c. link.



Lee el enunciado e indica la opción correcta:

¿Qué extensión se recomienda que tengan las hojas de estilo?

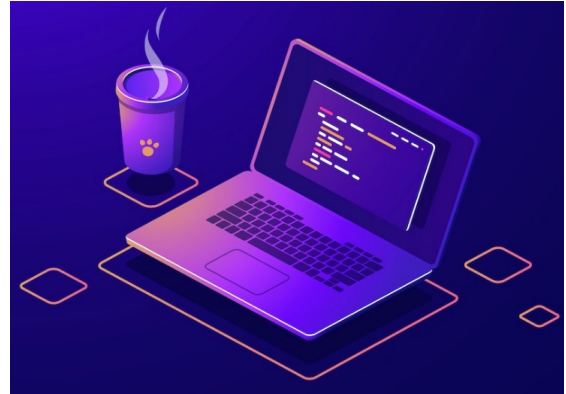
- a. css.
- b. ssh.
- c. txt.

9. PSEUDÓNIMOS



DESTACADO

Los **pseudónimos** o **pseudoselectores** nos permiten **definir** de manera más precisa los **estilos de los selectores**.



Existen dos tipos de **pseudónimos** o **pseudoselectores**:

- Pseudoclases.
- Pseudoelementos.

→ Pseudoclases.

Las pseudoclases **se añaden al selector** de forma que se pueda **aplicar un formato** determinado al **elemento seleccionado**.

Se definen asociando la particularidad al selector, mediante los **dos puntos**.

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

Así, por ejemplo, en un vínculo, podríamos **definir el color para cuatro posibles estados**:

```

/* No visitado */
a:link {
    color: red;
}

/* Visitado */
a:visited {
    color: green;
}

/* Ratón por encima */
a:hover {
    color: hotpink;
}

/* Seleccionado */
a:active {
    color: blue;
}

```

De igual manera podemos usar, por ejemplo, la pseudoclase **"focus"** para definir el estado de un elemento cuando recibe el **foco**.

```

input:focus {
    border: red 2px dotted;
    padding: 2px;
}

```



Existen otras muchas pseudoclases, las cuales puedes consultar en la documentación oficial que encontrarás dentro de los recursos para ampliar.

➔ **Pseudoelementos.**

A diferencia de las pseudoclasas, los **pseudoelementos** no describen el estado de un elemento, sino que se **añaden a un selector para definir estilos en una parte concreta** del documento HTML. Se definen de la siguiente manera:

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

La principal diferencia es que se usan **dos veces los dos puntos de manera consecutiva**.

También puedes consultar la lista en los recursos para ampliar, pero veremos algunos **ejemplos**:

```
/* Primera línea del elemento */  
div::first-line {  
    color: red;  
}  
  
/* Antes de un título añadimos "Título " */  
h1::before {  
    content: "Título ";  
}  
  
/* Después de un párrafo añadimos un punto */  
p::after {  
    content: ".";  
}  
  
/* Transformamos la primera letra a mayúscula */  
p::first-letter {  
    text-transform: uppercase;  
}
```

Te proponemos visualizar el siguiente **vídeo** para conocer más a fondo las **pseudoclases** y los **pseudoelementos**:

Vídeo: pseudoclases y pseudoelementos



Visualiza este vídeo en el que hablamos sobre las pseudoclases y los pseudoelementos.

Actividad de aprendizaje 3: uso de hojas de estilo externas

Ya hemos trabajado los estilos en línea y las hojas de estilo incrustadas, así que para que realmente acabes de dominar el tema tu profesor te pide que realizar la actividad anterior, pero esta vez con una **hoja de estilos externa**, llamada **estilos.css**.

Es decir, a partir del **documento** [AA2.html](#), tienes que conseguir, con **estilos CSS**, un resultado parecido a la imagen que veremos a continuación, pero dichos estilos quedarán en un **documento externo**, llamado **estilos.css**.

Lorem ipsum dolor sit amet

Nulla pretium. Sed tempus nunc vitae neque. **Suspendisse gravida**, metus a scelerisque sollicitudin, lacus velit ultricies nisl, nonummy tempus neque diam quis felis. **Etiam sagittis tortor** sed arcu sagittis tristique.

Aliquam tincidunt, sem eget volutpat porta

Vivamus velit dui, placerat vel, feugiat in, ornare et, urna. **Aenean turpis metus**, **aliquam non**, **tristique in**, pretium varius, sapien. Proin vitae nisi. Suspendisse **porttitor purus ac** elit. Suspendisse eleifend odio at dui. In in elit sed metus pretium elementum.

Titul de la taula

	Titul columna 1	Titul columna 2
Titul fila 1	Donec purus ipsum	Curabitur blandit
Titul fila 2	Donec purus ipsum	Curabitur blandit
	Titul columna 1	Titul columna 2

Donec purus ipsum, posuere id, venenatis at, placerat ac, lorem. Curabitur blandit, eros sed gravida aliquet, risus justo porta lorem, ut mollis lectus tortor in orci. Pellentesque nec augue.

Fusce nec felis eu diam pretium adipiscing. **Nunc elit elit**, **vehicula vulputate**, venenatis in, posuere id, lorem. Etiam sagittis, tellus in ultrices accumsan, diam nisi feugiat ante, eu congue magna mi non nisl.

Vivamus ultrices aliquet augue. **Donec arcu pede**, **pretium vitae**, rutrum aliquet, tincidunt blandit, pede. Aliquam in nisi. Suspendisse volutpat. Nulla facilisi. Ut ullamcorper nisi quis mi.

Recuerda que tendrás que **modificar** también el **archivo .html** para vincular el nuevo **archivo .css** creado.

Entrega los **archivos** en el **espacio habilitado para ello**.

10. HOJAS DE ESTILO AUDITIVAS

El **contenido principal** de casi cualquier soporte informático, y en particular de las páginas web, es normalmente **visual**: texto, imágenes, vídeos...

Este hecho, como podemos deducir, **dificulta** enormemente la **navegación** a personas con **deficiencias visuales**. En este sentido, las **hojas de estilo auditivas** nacieron con el propósito de ayudar a mitigar estos problemas.

Conozcamos mejor este elemento:



→ Concepto.



DESTACADO

Las **hojas de estilo auditivas** proporcionan información a las personas con **deficiencias visuales**, las cuales usan **navegadores de voz** para obtener una experiencia lo más parecida posible al comportamiento normal.

Cuando un usuario con deficiencias visuales hace uso de este tipo de navegadores se combina la **síntesis de voz** (o TTS) con los **iconos auditivos** (o "audio cues").

→ ¿Para qué sirven?

Lo que se puede hacer mediante las hojas de estilo auditivas es, por ejemplo, controlar el **tono de la voz sintetizada**, así como el **volumen** u otras características.

Esto hace que el usuario no esté escuchando un texto hablado de manera **plana**, sino que disponga de una serie de **matices** que hagan más **realista** la experiencia.



Estas propiedades CSS se pueden combinar con las **propiedades visuales** o bien podemos definir las de manera **totalmente alternativa**.

→ Modular la voz sintetizada.

Podemos usar **tres propiedades** principales para modular la voz sintetizada:

- ✓ Rest.
- ✓ Cue.
- ✓ Pause.

Estas tres propiedades permiten realizar ciertas **pausas en el texto** y se asemejan a lo que podríamos obtener de manera **visual** con el **padding**, **border** o **margin**, respectivamente.



Además, existen otras propiedades, que nos permiten, por ejemplo, **subir o bajar el volumen de la voz**, hacer que algún trozo de texto no se lea o balancear el volumen hacia derecha o izquierda.

Actividad de aprendizaje 4: márgenes y rellenos de cajas en CCS

La siguiente y última actividad que el profesor quiere mostrar a la empresa trata sobre los **márgenes y rellenos de cajas en CSS**.

A partir de los códigos [HTML](#) y [CSS](#) proporcionados, se han de determinar las reglas CSS necesarias para añadir los siguientes **márgenes y rellenos**:

- El elemento #cabecera debe tener un relleno de 1em en todos los lados.
- El elemento #menu debe tener un relleno de 0.5em a todos los lados y un margen inferior de 0.5em.

- El resto de elementos (#noticias, #publicidad, #principal, #secundario) deben tener 0.5em de relleno a todos sus lados, excepto el elemento #pie, que sólo debe tener un relleno de 1em.
- Los elementos .artículo deben mostrar una separación entre ellos de 1em.
- Las imágenes de los artículos muestran un margen de 0.5em en todos sus lados.
- El elemento #publicidad está separado 1em de su elemento superior.
- El elemento #pie debe tener un margen superior de 1em.

Si habéis aplicado correctamente los estilos, la web tendría el aspecto mostrado en la siguiente imagen:



Colaborad entre vosotros para **crear todos los estilos** de la página web.

Entrega los **archivos** en el **espacio habilitado para ello**.

Cuestionario



Lee el enunciado e indica la opción correcta:

¿Qué pseudoclase se usa para modificar un estilo cuando pasa el ratón por encima de un vínculo?

- a. visited.
- b. hover.
- c. focus.



Lee el enunciado e indica la opción correcta:

¿Qué significan las siglas de TTS?

- a. Text Type Speaking.
- b. Type Text Speech.
- c. Text to Speech.



Lee el enunciado e indica la opción correcta:

Un pseudoelemento se define con...

- a. ::
- b. :
- c. ;

RESUMEN

En la primera unidad vimos lo que eran las **interfaces gráficas** y cómo podíamos implementarlas en páginas web, mediante los **elementos conceptuales y visuales**.

Durante esta unidad, y teniendo claros esos conceptos, hemos entrado a conocer el lenguaje que nos permite cambiar la **apariencia** de las páginas web, mediante los llamados **estilos**. Este lenguaje de marcas es el que se denomina **hojas de estilo en cascada**, o comúnmente, **CSS**.

Hemos visto diferentes maneras de **crear y aplicar estos estilos a las páginas web**, bien sean en **línea, incrustados**, o de manera **externa**. Cada uno aporta sus ventajas, aunque el último método es el más utilizado. Además, hemos visto el **modelo de cajas** y cómo nos permite **estructurar la información** de manera sencilla dentro de nuestra página web.

Finalmente hemos introducido el concepto de **hojas de estilo auditivas**, que permiten a personas con deficiencias visuales obtener una mejor experiencia a la hora de visitar páginas web.

RECURSOS PARA AMPLIAR



PÁGINAS WEB

- Color-hex Color Codes: <https://www.color-hex.com/> [Consulta septiembre 2022].
- Pseudo-classes - CSS | MDN: <https://developer.mozilla.org/es/docs/Web/CSS/Pseudo-classes> [Consulta septiembre 2022].
- Pseudoelementos - CSS | MDN: <https://developer.mozilla.org/es/docs/Web/CSS/Pseudo-elements> [Consulta septiembre 2022].
- Referencia de voz CSS: https://es.w3hmong.com/cssref/css_ref_aural.htm [Consulta septiembre 2022].



BIBLIOGRAFÍA



PÁGINAS WEB

- CSS | MDN: <https://developer.mozilla.org/es/docs/Web/CSS> [Consulta septiembre 2022].
- HTML: Lenguaje de etiquetas de hipertexto | MDN: <https://developer.mozilla.org/es/docs/Web/HTML> [Consulta septiembre 2022].
- W3C: <https://www.w3.org/> [Consulta septiembre 2022].



GLOSARIO

- **Código hexadecimal:** código numérico que permite representar números mediante dígitos con 16 opciones (0-9, A-F). Muy usado en informática, por ejemplo, para la representación de colores.
- **Píxel:** cada uno de los puntos que representan una imagen digital. Sirven para representar también tamaños, tanto en alto como en ancho.
- **Responsive:** método para hacer las páginas web adaptables a dispositivos de distintos tamaños, como navegadores web, tabletas o teléfonos móviles.
- **TTS (Text to Speech):** tecnología que permite convertir un texto a voz, mediante sintetizadores. Usado, por ejemplo, por GPS o para personas con deficiencias visuales.