

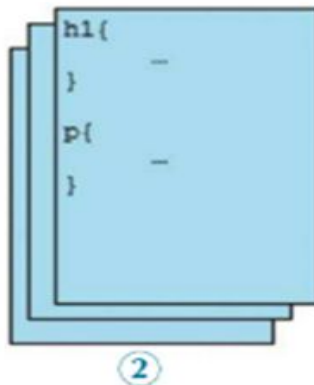
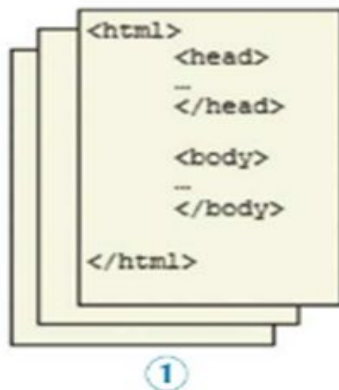
UT4- Hojas de estilos CSS

Cascading Style Sheets

Motivación

- Cuando se trabaja con documentos escritos en HTML y en XHTML, es necesario que presenten un aspecto determinado y un estilo concreto.
- CSS (Cascading Style Sheets) es un lenguaje basado en hojas de estilo que será el encargado de controlar el aspecto y apariencia del sitio web.
- La forma más eficiente para llevar esto a cabo consiste en distinguir entre la parte en la que se sitúan los contenidos y la parte del proyecto donde se recogen los aspectos relativos a la apariencia.
- El uso de hojas de estilo aporta numerosas ventajas: mejora la accesibilidad del documento, facilita la visualización en dispositivos diferentes, reduce la complejidad de su mantenimiento, tanto si va a ser realizado por uno mismo como por terceros.

Motivación



1. Creación de documentos estructurados en HTML/XHTML, donde se sitúan los contenidos, es decir, donde se crean los elementos de cada página (párrafo, titular, texto destacado, tabla, lista de elementos, etc.).
2. Creación de hojas de estilo en lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc.
3. La combinación de estos documentos da como resultado la vista final del proyecto web.





- No todos los navegadores ofrecen el mismo soporte a las guías de estilo. Es conveniente conocerse el motor de cada navegador, dado que es la parte que se encarga de interpretar el código HTML y CSS. En CSS disponemos hasta la versión 3 y en cuanto a motores del navegador: Gecko, Webkit, Presto.

1. ¿ Cómo utilizar hojas de estilo?

Para añadir estilos a una página web podemos hacerlo de las siguientes maneras:

1. Haciendo uso del atributo style que poseen las etiquetas HTML

```
<body style="background-color:  aqua;">  
  <header style="background-color:  beige;">
```

2. En la cabecera <head> del documento html haciendo uso de la etiqueta <style>

```
<head>  
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859  
  -1"/>  
  <title>  
    Ejemplo de estilos CSS en el propio documento  
  </title>  
  <style type="text/css">  
    p {  
      color: black; font-family: Verdana;  
    }  
  </style>  
</head>
```

1. ¿ Cómo utilizar hojas de estilo?

3. Colocar las hojas de estilo en uno o más documentos separados del contenido HTML. Desde el fichero principal de la página desarrollada en HTML se accede al fichero en código CSS, a través de la etiqueta <link> en la cabecera <head>. El fichero donde se definen los estilos tendrá la extensión .css

```
~/Desktop/Libro Diseño Interfaces WEB/cap4/cap4.html
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2 [http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <html>
4 <head>
5   <meta http-equiv="Content-type" content="text/html; charset=utf-8"/>
6   <title>
7     Ejemplo de estilos CSS en un archivo externo
8   </title>
9   <link rel="stylesheet" type="text/css" href="guia.css" media="screen" />
10 </head>
11 <body>
12   <p>Un párrafo de texto.</p>
13 </body>
14
15 </html>
16
```

```
~/Desktop/Libro Diseño Interfaces WEB/cap4/guia.css
1 p{
2   color: red;
3   font-family:
4   Verdana;
5 }
```

1. ¿ Cómo utilizar hojas de estilo?

La opción más recomendable es la tercera, separando los estilos del HTML ya que usar el CSS desde el propio código HTML tiene el principal inconveniente que, si se quiere hacer una modificación en los estilos definidos, es necesario modificar todas las páginas que incluyen el estilo que se va a modificar.

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
  <title>
    Ejemplo de estilos CSS en el propio documento
  </title>
  <link rel="stylesheet" type="text/css" media="screen, handheld"
    href="hojaEst.css">
</head>
<body>
  <p>Un párrafo de texto.</p>
</body>
</html>
```

Al configurar los estilos en un fichero a parte cualquier cambio realizado en el diseño afectaría al mismo tiempo a todas las páginas HTML que utilicen dicho fichero CSS consiguiendo uno de los principios fundamentales en el diseño que es, mantener la uniformidad de estilo a lo largo de todas las páginas de un sitio web.

1. ¿ Cómo utilizar hojas de estilo?

```
<link rel="stylesheet" type="text/css" media="screen, handheld"  
href="hojaEst.css">
```

La función de cada uno de los atributos usados para enlazar el documento HTML con la hoja de estilos en CSS es:

- **rel:** Indica el tipo de relación existente entre el recurso enlazado, el archivo CSS y la página HTML. Para los archivos CSS el valor es "stylesheet".
- **type:** indica el tipo de recurso enlazado, en el caso de los archivos CSS su valor es "text/css".
- **href:** Indicamos la URL del archivo CSS, esta puede ser relativa o absoluta y puede apuntar a un recurso interno o externo al sitio web.
- **media:** indica el tipo de medio en el que se van a aplicar los estilos del archivo CSS.

Dentro del fichero [hojaEst.css](#) tendríamos por ejemplo el siguiente contenido:

```
body{  
    background-color:olive;  
    color:white;  
}
```

1. ¿ Cómo utilizar hojas de estilo?

```
body{  
  background-color:olive;  
  color:white;  
}
```

La estructura siempre es la misma:

- ▶ Primero se indica el **selector** (para saber a quién afecta el código), que en este caso es **body**. La línea se finaliza la línea con una llave **{**.
- ▶ Posteriormente se indica la **propiedad** a modificar (en este caso **background-color** para modificar el color de fondo y **color** para especificar el color del texto).
- ▶ Seguido a la propiedad se indica el nuevo **valor** (separados ambos por dos puntos :).
- ▶ Para finalizar cada una de las sentencias CSS se utiliza obligatoriamente el símbolo **;** del **punto y coma**.
- ▶ Para acabar el bloque CSS utilizamos la **llave de cierre }**.

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


2. Adecuación al medio o dispositivo de visualización

- Las hojas de estilo permiten especificar cómo se representa un documento, en función del tipo de medio de visualización. Como medios entendemos los canales a través de los cuales se muestra la información del sitio web: auditivos, visuales, impresos, etc. Lo habitual es que se utilice una propiedad definida para cada tipo de canal. El más común es el utilizado para definir el aspecto del sitio web en pantalla (screen) y el de la página para ser impresa (print).
- En el cuadro se recogen los tipos más frecuentes:

Medio	Descripción
<i>all</i>	Todos los medios definidos.
<i>print</i>	Impresoras.
<i>braille</i>	Dispositivos táctiles que emplean el sistema braille.
<i>embossed</i>	Impresoras braille.
<i>handheld</i>	Dispositivos de mano: móviles, PDA, etc.
<i>projection</i>	Proyectores y dispositivos para presentaciones.
<i>screen</i>	Pantallas de ordenador.
<i>speech</i>	Sintetizadores para navegadores de voz utilizados por personas discapacitadas.
<i>tv</i>	Televisores y dispositivos con resolución baja.

2. Adecuación al medio o dispositivo de visualización

Para especificar el medio de destino se utilizan varios mecanismos, desde la hoja de estilos en CSS o desde el propio fichero HTML del sitio web.

@media

La regla @media especifica los tipos de medios de destino. A través de esta opción es posible que, en una misma hoja de estilo, aparezcan reglas para distintos medios. Si un mismo estilo se aplica a varios medios, basta con indicar sus nombres separados por comas.

Sintaxis:

```
}  
@medianombreTipoMedio{  
  selector{  
    propiedad: valor;  
  }  
}
```

En el siguiente ejemplo, se especifican unas reglas para el dispositivo print, otras para screen y otras que se aplicaran tanto para screen como print.

@media print { h1 { font-size: 15pt; } }	@media screen { h1 { font-size: 15px; } }	@media screen, print { h1 { line-height: 1.2 ; } }
--	---	--

2. Adecuación al medio o dispositivo de visualización

@import

La regla `@import` permite importar nuevas reglas desde otros ficheros CSS, e indicar el medio o medios sobre los que debe aplicarse el nuevo estilo. En este caso, a diferencia de `@media`, en primer lugar, aparece la URL que apunta al fichero CSS que contiene la guía de estilo, y, a continuación, se indica el medio sobre el que se va a aplicar. Hay que tener en cuenta que estas reglas deben preceder a todas las demás reglas.

Si los datos de estilo del fichero se van a aplicar a varios medios al mismo tiempo, se indicará el nombre de estos separados por comas.

Sintaxis:

```
@import url(nombreFichero.css) nombreTipoMedio;
```

Es posible encontrar las reglas combinadas entre sí en un mismo fichero escrito en HTML. Por ejemplo, anidando las siguientes líneas

- Utilizando *link*:

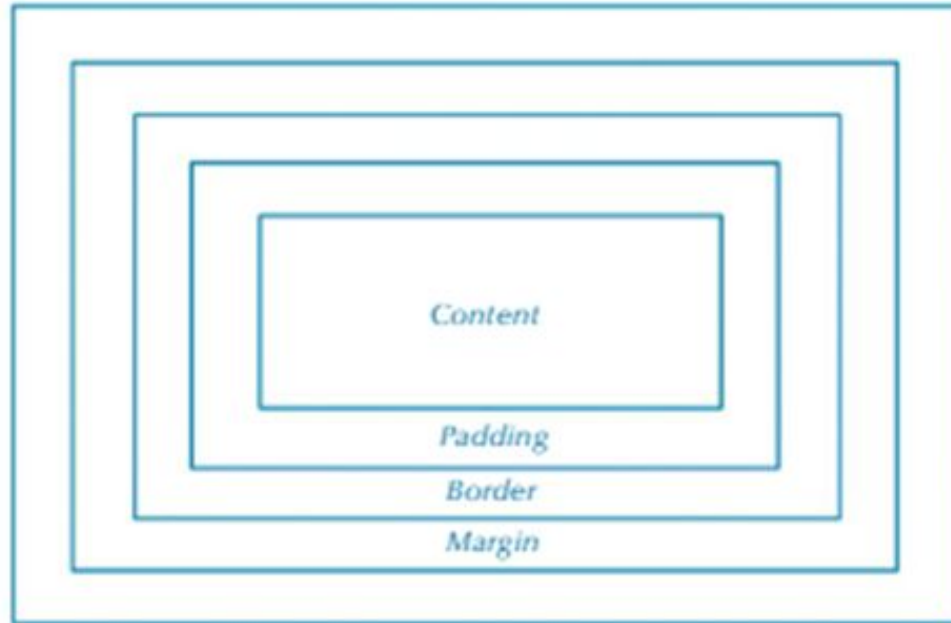
```
<link rel="stylesheet" type="text/css" media="screen" href="basico.css"/>
```
- Utilizando `@import`:

```
@import url("estilos_seccion.css") screen;
```
- Utilizando `@media`:

```
@media print {  
    /* Estilos específicos para impresora */  
}
```

3. Modelo de cajas CSS

Cuando se crea una página en HTML, el uso de cada etiqueta simula lo que podría considerarse una caja, dotando a cada una de ellas de un identificador, indicado por el nombre de la etiqueta. Este sistema de cajas permite, desde el fichero CSS, seleccionar cada caja y establecer el estilo deseado.



- Diagrama de modelo de cajas en CSS

3. Modelo de cajas CSS

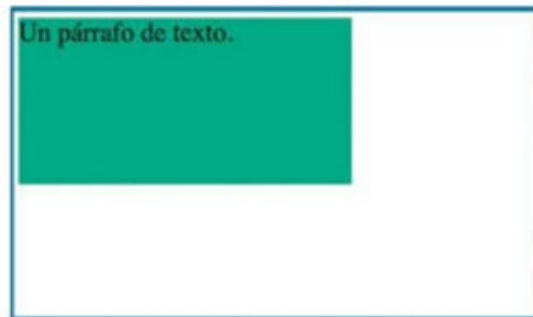
Propiedades principales con las que se trabaja en cada una de las partes del modelo de cajas:

Content

- Propiedades ***width*** y ***height*** establecen el ancho y el alto del área donde se muestra el contenido de la caja.
- Otra propiedad importante del área de contenido es el fondo de la caja: ***background-image*** (imagen de fondo), ***background-color*** (color de fondo). Mostrarán una imagen o un color detrás del contenido y el espacio de relleno.

Ejemplo:

```
article {  
    background: #5DEAAB;  
    height: 100px;  
    width: 200px;  
}
```



3. Modelo de cajas CSS

Padding

Espacio que existe entre la caja de contenido y el borde. Se puede modificar la altura y anchura. Se distinguen, por tanto, cuatro zonas posibles de acción: zona inferior (***padding-bottom***), zona superior (***padding-top***), zona izquierda (***padding-left***) y zona derecha (***padding-right***).

Ejemplo:

```
article {  
  padding-top: 10px;  
  padding-right: 30px;  
  padding-bottom: 10px;  
  padding-left: 20px;  
}
```

- Si solo escribimos padding, se aplicará el mismo formato a todos los extremos

```
article {  
  padding: 10px;  
}
```

Podemos colocar la palabra padding seguida de varias cifras separadas sin comas. Si aparecen dos valores, el primero es asignado al límite superior y al inferior, mientras que la segunda cifra será la que se aplique al relleno vertical. En el caso de tener tres valores, el primero y último serán los aplicados a los límites superior e inferior, y la segunda cifra indicará el valor de los lados izquierdo y derecho.

3. Modelo de cajas CSS

Border

Esta zona es la que encierra el contenido y el relleno, *content* y *padding*. Es posible modificar su **grosor**, **estilo** y **color**; además, al igual que ocurre con padding, es posible dar formato a los cuatro extremos de la caja a la vez utilizando solo la palabra ***border*** o hacerlo de forma individual.

Ejemplo:

```
article {  
    border: 1px solid #000000;  
}
```

- Si se quiere dar un formato diferente a cada uno de los extremos, se utilizarán algunas de las propiedades individuales como:

border-top, ***border-right***, ***border-bottom***, ***border-left***:

modifican tanto el grosor, el estilo y el color de cada lado del borde.

border-width, ***border-style***, ***border-color***: modifican de forma individual grosor, el estilo y el color de los cuatro extremos a la vez.

border-top-width: modifica el grosor del borde superior.

border-top-style: modifica el estilo del borde superior.

border-top-color: modifica el color del borde superior.

....

3. Modelo de cajas CSS

Margin

Envuelve al resto de elementos CSS y sostiene a otras cajas del diseño.

Ejemplo:

```
article {  
    margin-top: 10px;  
    margin-right: 10px;  
    margin-bottom: 10px;  
    margin-left: 10px;  
}
```

- Las propiedades individuales son ***margin-top***, ***margin-right***, ***margin-bottom*** y ***margin-left***. El funcionamiento es el mismo que el de la propiedad padding.

4. Propiedades heredables

La herencia en CSS es el mecanismo mediante el cual determinadas propiedades de un elemento “padre” se transmiten a sus “hijos”. No todas las propiedades CSS son heredables. Por ejemplo, los márgenes no se heredan porque es poco probable que un elemento “hijo” necesite los mismos márgenes que su “padre”. Cuando se establece el valor de una propiedad CSS en un elemento, sus elementos descendientes heredan de forma automática el valor de esa propiedad.

Ejemplo:

```
<style type="text/css">
    body {color: blue; }
</style>
```

En el ejemplo, el selector body solo establece el color de la letra para este elemento pero esta propiedad se hereda de forma automática lo que significa que los elementos descendientes mostrarán el mismo color.

La herencia automática se puede anular estableciendo de forma explícita otro valor para la propiedad que se hereda.

Existen otras propiedades cuyo valor no se hereda, como es el caso de los borders (border); es decir, si se define un estilo relativo al borde para el cuerpo, este no será aplicado al resto de elementos HTML.

4. Propiedades heredables

Algunas propiedades heredables por defecto en HTML:

color	text-align
font	visibility
letter-spacing	white-space
direction	border-collapse
word-spacing	quotes
line-height	list-style-image

Para las propiedades no heredables existe un mecanismo mediante el cual es posible indicar que un elemento herede el valor de una propiedad. Para ello se utilizan los valores: ***inherit***, ***initial*** y ***unset*** que modifican el valor de la propiedad respecto a la herencia.

- ***inherit***: aplica el valor de la propiedad del elemento “padre”
- ***initial***: aplica el valor que presente inicialmente la propiedad.
- ***unset***: hereda el valor de la propiedad del elemento “padre”; si este no existe, aplica el valor inicial.

4. Propiedades heredables

En este ejemplo, el elemento *h1* hereda todas las propiedades establecidas en el *body*. El color se hereda de forma automática y al no establecerse otro valor para el color en *h1* quedará de color rojo. Ahora bien, en este ejemplo las propiedades de borde también serán heredadas puesto que así es indicado por el valor *inherit*.

```
body {  
    border-width: 10px;  
    border-color: blue;  
    color:red;  
}  
  
h1 {  
    border:inherit;  
}
```