

UD3.1 INTRODUCCIÓN AL CSS

ÍNDICE

1. INTRODUCCIÓN A CSS (CASCADING STYLE SHEETS).....	2
1.1. ¿Qué es CSS?	2
1.2. ¿Por qué es importante CSS en el diseño web?	2
1.3. Historia y evolución de CSS	3
2. INCORPORANDO CSS EN HTML	4
2.1. Selectores de Tipo, Clase e ID	4
2.2. Uso de CSS en línea, Incrustado y Externo	6
2.3. Pseudo-Clases.....	7
3. PROPIEDADES Y VALORES	9
4. EL MODELO CAJA.....	11
4.1. Cajas en Bloque y en Línea	11
4.2. Partes de una caja.....	12
4.3. El modelo de cajas CSS estándar.....	13
4.4. El modelo de cajas CSS alternativo	14

1. INTRODUCCIÓN A CSS (CASCADING STYLE SHEETS)

1.1. ¿Qué es CSS?

CSS, que significa "Cascading Style Sheets" en inglés, es un lenguaje de hojas de estilo utilizado en la creación y diseño de páginas web. Su función principal es controlar la presentación y el aspecto visual de un sitio web, permitiendo a los diseñadores y desarrolladores web definir cómo se verán los elementos HTML en el navegador. A través de CSS, se pueden establecer propiedades como el color, la tipografía, el espaciado y la posición de los elementos en una página web.

Entre sus características básicas están:

- **Separación de contenido y presentación:** Una de las características clave de CSS es la separación de contenido y presentación. Esto significa que podemos mantener el contenido de una página web (HTML) separado de su estilo (CSS), lo que facilita la gestión y el mantenimiento del sitio.
- **Reutilización:** El código CSS se puede utilizar para muchas páginas distintas y aprovecharlo.
- **Cascading (Cascada):** El término "cascading" en CSS se refiere a la forma en que las reglas se aplican y se combinan. Las reglas pueden heredarse y anularse en función de su especificidad y ubicación en el código, lo que brinda un alto grado de flexibilidad en el diseño.

1.2. ¿Por qué es importante CSS en el diseño web?

CSS desempeña un papel fundamental en el diseño web por varias razones:

- **Mejora la usabilidad:** El uso adecuado de CSS permite mejorar la usabilidad de un sitio web al proporcionar una presentación limpia y coherente. Los usuarios pueden navegar y comprender el contenido con mayor facilidad.
- **Flexibilidad de diseño:** CSS ofrece una gran flexibilidad en la creación de diseños web. Puedes posicionar elementos, cambiar tamaños, colores y fuentes de manera precisa, lo que te permite conseguir diseños únicos y atractivos.
- **Adaptabilidad:** Con CSS, puedes crear sitios web que se adaptan a diferentes dispositivos y tamaños de pantalla, lo cual es esencial en la era de los dispositivos móviles.
- **Eficiencia en el mantenimiento:** Al mantener la presentación por separado del contenido, cualquier cambio en el estilo se refleja de manera consistente en todas las páginas del sitio, lo que simplifica las actualizaciones y el mantenimiento.

1.3. Historia y evolución de CSS

El desarrollo de CSS se remonta a la década de 1990, cuando se reconoció la necesidad de separar la presentación del contenido en la web. A lo largo de los años, CSS ha experimentado varias versiones y mejoras, lo que ha llevado a un estándar más maduro y poderoso. Algunos hitos en la evolución de CSS incluyen:

- **CSS1** (1996): La primera versión de CSS, que introdujo las propiedades básicas para el estilo de texto y el diseño de páginas.
- **CSS2** (1998): Amplió las capacidades de CSS, permitiendo un mayor control sobre el diseño y la presentación de páginas web.
- **CSS3** (en curso): CSS3 es una especificación en constante evolución que ha introducido numerosas características nuevas, como animaciones, sombras, transformaciones y soporte para fuentes web personalizadas.
- **CSS4** (en desarrollo): A medida que la web continúa evolucionando, se están desarrollando nuevas características y módulos para CSS, lo que amplía aún más su potencial.

2. INCORPORANDO CSS EN HTML

Una parte esencial de trabajar con CSS es aprender cómo integrarlo en un documento HTML. Existen varias maneras de hacerlo, y cada una tiene sus propias ventajas y desventajas. A continuación, exploraremos las diferentes maneras de incorporar CSS en un documento HTML:

2.1. Selectores de Tipo, Clase e ID

Antes de sumergirnos en las técnicas de integración, es importante comprender los tipos de selectores que puedes utilizar en CSS. Esto influirá en la forma en que aplicas tus estilos.

- **Selector de Tipo:** Selecciona todos los elementos de un tipo específico. Por ejemplo, para aplicar estilos a todos los párrafos en tu documento HTML, usarías el selector de tipo "p".

```
p {  
  color: blue;  
}
```

- **Selector de Clase:** Selecciona elementos que tienen una clase específica. Puedes aplicar la misma clase a varios elementos en tu HTML y luego estilizarlos de manera uniforme. El selector de clase se indica con un punto, como ".clase".

```
<p class="destacado">Este párrafo está en la clase destacado.</p>
```

```
.destacado {  
  color: red;  
  font-weight: bold;  
}
```

- **Selector de ID:** Selecciona un elemento con un ANEU específico. Los IDs deben ser únicos en una página, por lo que este selector es útil para aplicar estilos a un solo elemento. Se indica con un símbolo de cojinete, como "#aneu".

```
<div id="encabezado">Este es el encabezado</div>
```

```
#encabezado {  
  background-color: gray;  
  color: white;  
}
```

- **Selector universal:** El selector universal selecciona todos los elementos en la página. Puedes aplicar estilos generales a todos los elementos de esta manera. Aquí tienes un ejemplo:

```
* {  
  margin: 0;  
  padding: 0;  
}
```

- **Selector de descendencia:** El selector de descendencia te permite seleccionar elementos hijos dentro de otros elementos. Por ejemplo:

```
div p {  
  color: green;  
}
```

2.2. Uso de CSS en línea, Incrustado y Externo

CSS se puede incorporar en un documento HTML de diversas formas. Cada enfoque tiene sus propias aplicaciones y ventajas.

- **CSS en línea:** Puedes agregar estilos directamente a un elemento HTML utilizando el atributo "style". El CSS en línea es útil para estilos específicos en elementos individuales, pero puede ser difícil de mantener en páginas grandes. Por ejemplo:

```
<p style="color: blue;" >Este es un párrafo azul</p>.
```

- **CSS incrustado (en línea interna):** Puedes insertar CSS en la sección <head> de tu documento HTML utilizando la etiqueta <style>. Esto permite que el CSS se aplique a toda la página. El CSS incrustado es útil para aplicar estilos en toda la página, pero todavía no proporciona la separación ideal de contenido y presentación. Por ejemplo:

```
<head>
<style>
p {
  color: blue;
}
</style>
</head>
```

- **CSS externo:** La forma más común y eficiente de aplicar CSS es mediante un archivo externo. Crea un archivo CSS separado (por ejemplo, "estils.css") y vincúlalo a tu documento HTML con la etiqueta <link> en la sección <head>. El CSS externo promueve la separación de contenido y presentación, facilitando la gestión de estilos en sitios web grandes y complejos. Además, los cambios en el archivo CSS se reflejarán automáticamente en todas las páginas que lo utilizan. Por ejemplo:

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

2.3.Pseudo-Clases

Las pseudo-clases en CSS son selectores especiales que te permiten aplicar estilos a elementos en función de su estado o posición en relación con otros elementos. Aquí hay algunas pseudo-clases comunes junto con ejemplos para ilustrar su uso:

- **:hover.** Esta pseudo-clase se aplica cuando el cursor del mouse se sitúa sobre un elemento. Es muy útil para crear efectos de resaltado o interactividad.

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

En este ejemplo, cuando el cursor se sitúa sobre un enlace ('<a>'), el texto del enlace se volverá rojo.

- **:active.** Se aplica cuando un elemento está siendo activado, como cuando se hace clic en un enlace. Es común usarlo para cambiar el estilo de un elemento durante una interacción.

```
button:active {  
  background-color: #00f;  
  color: #fff;  
}
```

Ahí, cuando se hace clic en un botón, su fondo se vuelve azul ('#00f') y el texto se vuelve blanco ('#fff').

- **:focus.** Esta pseudo-clase se aplica cuando un elemento recibe el enfoque, generalmente mediante tabulación en un formulario o haciendo clic en un elemento interactivo. Es útil para proporcionar retroalimentación visual a los usuarios.

```
input:focus {  
  border: 2px solid green;  
}
```

Cuando un campo de entrada ('<input>') obtiene el foco, su borde se vuelve verde.

- **:first-child.** Selecciona el primer elemento hijo de un elemento padre.

```
ul li:first-child {  
  font-weight: bold;  
}
```

En este caso, se aplicará un estilo de texto en negrita al primer elemento de la lista dentro de un elemento ''.

- **:last-child**. Selecciona el último elemento hijo de un elemento padre.

```
ul li:last-child {  
  text-decoration: underline;  
}
```

El último elemento de la lista dentro de un '' tendrá una decoración de subrayado.

- **:nth-child(n)**. Permite seleccionar elementos específicos en función de su posición en relación con sus hermanos. Puedes usar fórmulas matemáticas para definir 'n' y seleccionar elementos concretos.

```
li:nth-child(odd) {  
  background-color: #f2f2f2;  
}
```

Los elementos impares en una lista '' tendrán un fondo gris claro.

- **:not(selector)**. Excluye elementos que coinciden con el selector especificado.

```
p:not(.destacat) {  
  color: #666;  
}
```

Todos los párrafos ('<p>') que no tienen la clase '.destacado' tendrán un color de texto gris ('#666').

3. PROPIEDADES Y VALORES

CSS. Guía de referencia rápida

Sintaxis	Unidades
<pre>/* Comentario */ selector { propiedad: valor; }</pre>	<p>px em ex % in cm mm pt pc #ff6633 #f63 rgb(0,255,0) rgb(0%,100%,50%) aqua black blue fuchsia gray green lime maroon navy olive purple red silver teal white yellow</p>
Selectores	Pseudo-classes y pseudoelementos
<p>* Todos los elementos</p> <p>div <div></p> <p>div * Elementos dentro de <div></p> <p>div span dentro de <div></p> <p>div, span <div> y </p> <p>div > span con <div> como padre</p> <p>div + span precedido de <div> </div></p> <p>.foto Elementos de la clase "foto"</p> <p>div.class <div> de la clase "foto"</p> <p>#casa Elementos con id "casa"</p> <p>div#itemid <div> con id "casa"</p> <p>a[class] <a> con el atributo class</p> <p>a[class='x'] <a> cuando su clase es "x"</p> <p>a[class~='x'] <a> cuando su clase es una lista de valores y uno de ellos es 'x'</p> <p>a[lang]='en'] <a> cuando lang comienza por "en"</p>	<p>:active Elemento seleccionado</p> <p>:focus Elemento enfocado</p> <p>:hover Elemento con el cursor del ratón encima</p> <p>:link Vínculo no visitado</p> <p>:visited Vínculo visitado</p> <p>:first-child Elemento que es el primer hijo de otro elemento</p> <p>:lang Language utilizado en un elemento</p> <p>:first-letter La primera letra de un texto</p> <p>:first-line La primera línea de un texto</p> <p>:before Contenido antes de un elemento</p> <p>:after Contenido después de un elemento</p>

Propiedad / Valor

Cajas	
border (border-left, border-bottom, border-right, border-top) <i>border-width border-style border-color</i>	
border-width (border-left-width; border-top-width; border-right-width; border-bottom-width) thin medium thick <i>longitud</i>	
border-color (border-left-color; border-top-color; border-right-color; border-bottom-color) <i>color</i>	
border-style (border-left-style; border-top-style; border-right-style; border-bottom-style) none hidden dotted dashed solid double groove ridge inset outset	
margin (margin-top; margin-right; margin-bottom; margin-left) auto <i>longitud</i>	
padding (padding-top; padding-right; padding-bottom; padding-left) <i>longitud</i>	

Posicionamiento	Dimensiones
left, top, right, bottom auto <i>longitud</i>	height auto <i>longitud</i>
clip shape auto	line-height * normal number <i>longitud</i>
overflow visible hidden scroll auto	max-height none <i>longitud</i>
position static relative absolute fixed	max-width none <i>longitud</i>
vertical-align baseline sub super top text-top middle bottom text-bottom <i>longitud</i>	min-height <i>longitud</i>
z-index auto <i>número</i>	min-width <i>longitud</i>
	width auto <i>longitud</i>

* Propiedades que se heredan. En rojo, propiedades de acceso rápido y, entre paréntesis, las propiedades a las que llaman. En cursiva, valores que deben ser incluidos por el usuario.

Contorno	Tablas
outline <i>outline-color outline-style outline-width</i> outline-color <i>color invert</i> outline-style none dotted dashed solid double groove ridge inset outset outline-width thin medium thick <i>longitud</i>	border-collapse * collapse separate border-spacing * <i>longitud longitud</i> caption-side * top bottom left right empty-cells * show hide table-layout auto fixed
Fondo	Listas
background <i>background-color background-image background-repeat background-attachment background-position</i> background-attachment scroll fixed background-color <i>Color transparent</i> background-image <i>url none</i> background-position top left top center top right center left center center center right bottom left bottom center bottom right x-% y-% x-pos y-pos background-repeat repeat repeat-x repeat-y no-repeat	list-style * <i>list-style-type list-style-position list-style-image</i> list-style-image * none <i>url</i> list-style-position * inside outside list-style-type * none disc circle square decimal decimal-leading-zero lower-roman upper-roman lower-alpha upper-alpha lower-greek lower-latin upper-latin hebrew armenian georgian cjk-ideographic hiragana katakana hiragana-iroha katakana-iroha
Fuente	Texto
font * <i>font-style font-variant font-weight font-size/line-height font-family</i> caption icon menu message-box small-caption status-bar font-family * family-name generic-family font-size * xx-small x-small small medium large x-large xx-large smaller larger <i>longitud</i> font-size-adjust * none <i>número</i> font-stretch * normal wider narrower ultra-condensed extra-condensed condensed semi-condensed semi-expanded expanded extra-expanded ultra-expanded font-style * normal italic oblique font-variant * normal small-caps font-weight * normal bold bolder lighter 100 200 300 400 500 600 700 800 900	color * <i>color</i> direction * ltr rtl letter-spacing * normal <i>longitud</i> text-align * left right center justify text-decoration none underline overline line-through blink text-indent * <i>longitud</i> text-shadow none <i>color longitud</i> text-transform * none capitalize uppercase lowercase unicode-bidi normal embed bidi-override white-space * normal pre nowrap word-spacing * normal <i>longitud</i>
Miscelánea I	Miscelánea II
clear left right both none cursor * <i>url</i> auto crosshair default pointer move e-resize ne-resize nw-resize n-resize se-resize sw-resize s-resize w-resize text wait help content <i>string url counter(name) counter(name, list-style-type) counters(name, string) counters(name, string, list-style-type) attr(X) open-quote close-quote no-open-quote no-close-quote</i> counter-increment none <i>número identificativo</i> counter-reset none <i>número identificativo</i>	display none inline block list-item run-in compact marker table inline-table table-row-group table-header-group table-footer-group table-row table-column-group table-column table-cell table-caption float left right none marker-offset auto <i>longitud</i> position static relative absolute fixed quotes * none <i>string string</i> visibility visible hidden collapse

4. EL MODELO CAJA

Todo en CSS tiene una caja alrededor, y comprender estas cajas es clave para poder crear diseños con CSS o para alinear elementos con otros elementos. En este punto, daremos un vistazo más de cerca al modelo de cajas en CSS con el que podrás crear diseños de compaginación más complejos con una comprensión de cómo funciona y la terminología relacionada.

4.1. Cajas en Bloque y en Línea

En CSS, en general, hay dos tipos de cajas: cajas en bloque y cajas en línea. Estas características se refieren al modo cómo se comporta la caja en términos de flujo de página y en relación con otras cajas de la página:

Si una caja se define como un bloque, se comportará de las maneras siguientes:

- La caja fuerza un salto de línea al llegar al final de la línea.
- La caja se extenderá en la dirección de la línea para llenar todo el espacio disponible que haya en su contenedor. En la mayoría de los casos, esto significa que la caja será tan ancha como su contenedor, y llenará el 100% del espacio disponible.
- Se respetan las propiedades `width` y `height`.
- El relleno, el margen y el borde mantienen a los otros elementos alejados de la caja.
- Salvo que decidimos cambiar el tipo de visualización a en línea, elementos como los encabezados (por ejemplo, `<h1>`) y todos los elementos `<p>` usan por defecto `block` como tipo de visualización externa.

Si una caja tiene una visualización externa de tipo `inline`, entonces:

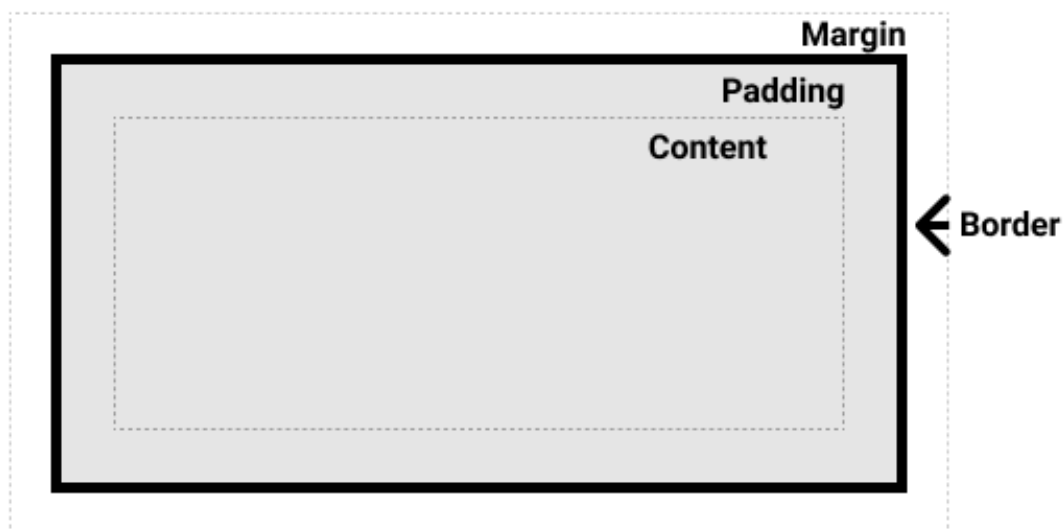
- La caja no fuerza ningún salto de línea al llegar al final de la línea.
- Las propiedades `width` y `height` no se aplican.
- Se aplican relleno, margen y bordes verticales, pero no mantienen alejadas otras cajas en línea.
- Se aplican relleno, margen y bordes horizontales, y mantienen alejadas otras cajas en línea.
- El elemento `<a>`, que se utiliza para los enlaces, y los elementos ``, `` y `` son ejemplos de elementos que se muestran en línea por defecto.

4.2. Partes de una caja

Al hacer una caja de tipo bloque en CSS tenemos los siguientes elementos:

- **El contenido de la caja (o content box):** El área donde se muestra el contenido, cuyo tamaño puede cambiarse utilizando propiedades como width y height.
- **El relleno de la caja (o padding box):** El relleno es espacio en blanco alrededor del contenido; es posible controlar su tamaño usando la propiedad padding y otras propiedades relacionadas.
- **El borde de la caja (o border box):** El borde de la caja envuelve el contenido y el de relleno. Es posible controlar su tamaño y estilo utilizando la propiedad border y otras propiedades relacionadas.
- **El margen de la caja (o margin box):** El margen es la capa más externa. Envuelve el contenido, el relleno y el borde como espacio en blanco entre la caja y otros elementos. Es posible controlar su tamaño usando la propiedad margin y otras propiedades relacionadas.

El diagrama siguiente muestra estas capas:



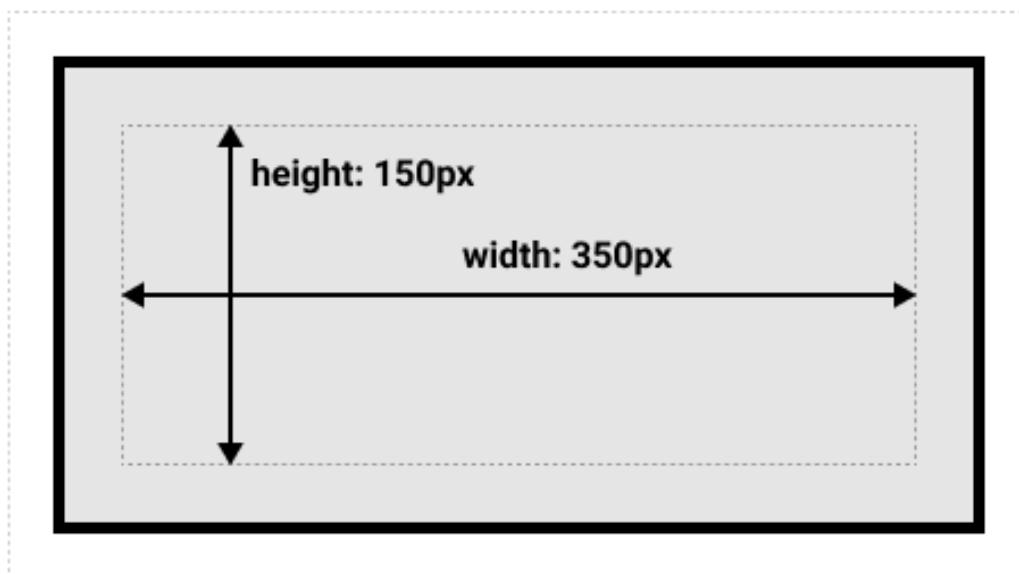
4.3.El modelo de cajas CSS estándar

En el modelo de cajas estándar, cuando estableces los atributos `width` y `height` para una caja, defines el ancho y el alto del contenido de la caja. Cualquier área de relleno y borde se afianza a ese ancho y alto para obtener el tamaño total que ocupa la caja. Esto se muestra en la imagen que encontrarás a continuación.

Si suponemos que la caja tiene el CSS siguiente, que establece los valores para las propiedades `width`, `height`, `margin`, `border`, y `padding`:

```
.box {  
  width: 350px;  
  altura: 150px;  
  margen: 10px;  
  relleno: 25px;  
  borde: 5px negro sólido;  
}
```

El espacio que ocupa nuestra caja usando el modelo de cajas estándar será en realidad de 410 px ($350 + 25 + 25 + 5 + 5$); y su altura, de 210 px ($150 + 25 + 25 + 5 + 5$), porque el área de relleno y el borde se afianza al ancho que se utiliza para el contenido de la caja.



4.4. El modelo de cajas CSS alternativo

Podrías pensar que es más bien incómodo tener que sumar el borde y el área de relleno para obtener el tamaño real de la caja, y tienes razón! Por este motivo, CSS introdujo un modelo de caja alternativo algún tiempo después del modelo de cajas estándar. Con este modelo, cualquier ancho es el ancho de la caja visible en la página, por lo tanto, el ancho del área de contenido es ese ancho menos el ancho para el relleno y el borde. El mismo CSS que hemos usado antes daría entonces el siguiente resultado (ancho = 350 px, altura = 150 px).



Por defecto, los navegadores usan el modelo de cajas estándar. Si deseas activar el modelo de cajas alternativo para un elemento, hazlo configurando `box-sizing: border-box`. Con ello, le dices al navegador que tome como el borde de la caja el área definida por cualquier tamaño que establezcas.

```
.box {  
  box-sizing: border-box;  
}
```

Si quieres que todos tus elementos usen el modelo de cajas alternativo (opción común entre los desarrolladores) debes establecer la propiedad `box-sizing` en el elemento `<HTML>`. Después tienes que configurar todos los demás elementos para que heredan ese valor, como se ve en el fragmento de código siguiente. Si deseas comprender qué hay detrás, consulta el artículo de CSS-Tricks sobre el tamaño de las cajas.

```
HTML {  
  box-sizing: border-box;  
}  
*,  
*::before,  
*::after {  
  box-sizing: inherit;  
}
```