A thick black L-shaped frame surrounds the text. It starts at the top left, goes right, then down, then right again at the bottom.

TEMA 3 PARTE 1

DEFINICIÓN DE

ESTILOS EN

ENTORNOS WEB. CSS

Lenguaje de Marcas y sistemas de Gestión de la
Información

CONTENIDOS

3.1 FORMAS DE AÑADIR CSS

3.1.1 ORDEN DE APLICACIÓN

3.2 REGLAS CSS

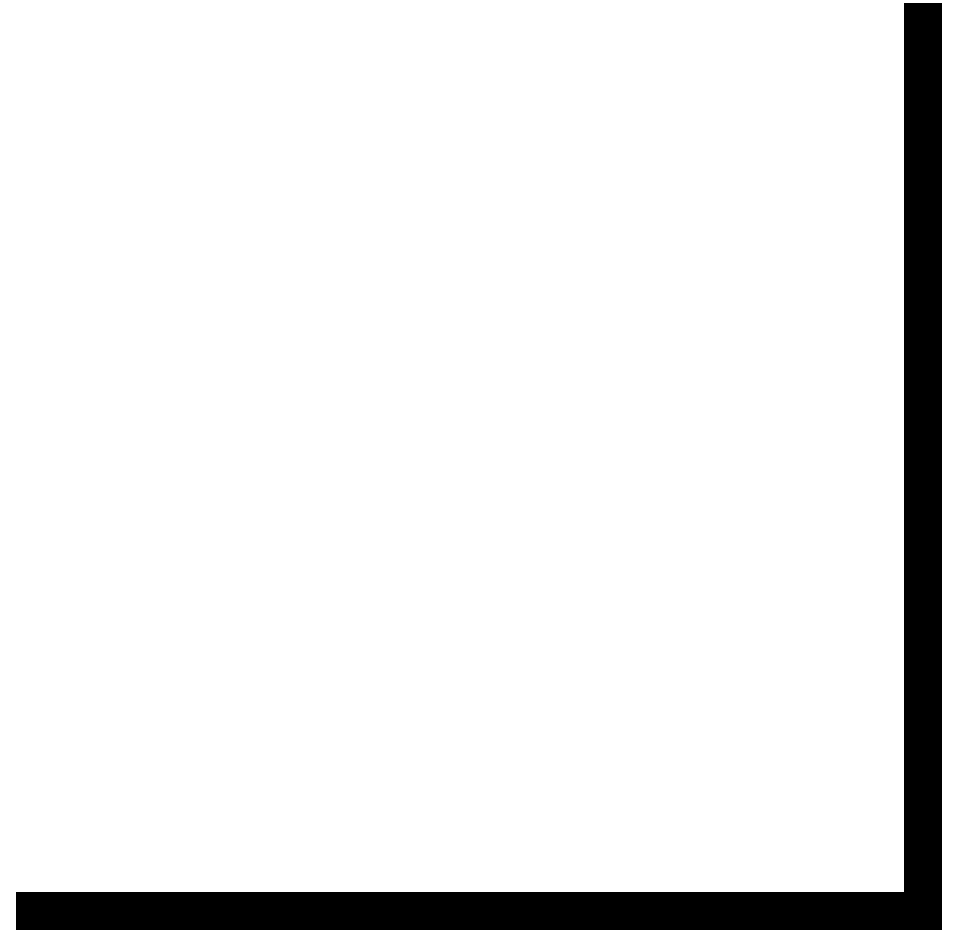
3.3 SELECTORES

3.3.1 SELECTORES SIMPLES

3.3.2 SELECTORES COMBINADORES

3.3.3 SELECTORES ATRIBUTOS

3.3.4 COMBINACIÓN SELECTORES



3.1 FORMAS DE AÑADIR CSS



CSS 3

Cascading Style Sheet;

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML.

CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas.

Obliga a crear documentos HTML bien definidos y con significado completo.

Mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.

Al crear una página web, se utiliza en primer lugar el lenguaje HTML para marcar los contenidos, es decir, para designar la función de cada elemento: párrafo, titular, texto destacado, tabla, lista de elementos, etc.

Una vez creados los contenidos, se utiliza el 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.

FORMAS DE AÑADIR CSS

ESTILOS EN LÍNEA (Definir el estilo directamente en las propias etiquetas de HTML usando el atributo style).

Esta forma es la peor y la menos utilizada, debes evitarla siempre que te sea posible ya que es una mala práctica.

A cualquier etiqueta **HTML** puedes añadirle la propiedad "style" y dentro de ella añadir tantos estilos como desees separados por punto y coma.

Esta práctica está totalmente desrecomendada ya que a la larga complica el mantenimiento de los estilos de tu página obligándote a escribir más código del necesario.

Observe cómo las reglas de estilo adoptan la forma propiedad:valor y van separadas por un punto y coma “;”. Todo ello se encierra entre comillas como con cualquier otro atributo.

FORMAS DE AÑADIR CSS

ESTILOS EN LÍNEA

```
<!DOCTYPE html>
```

```
<html lang="es">
```

```
<head>
```

```
    <title>Ejemplo de estilos CSS en el propio documento</title>
```

```
</head>
```

```
<body>
```

```
    <p>Un párrafo de texto.</p>
```

```
    <h1 style="color:red; font-size:2em">Título de la página</h1>
```

```
    <p style="color:red;"> Este es otro párrafo de la página 1 </p>
```

```
</body>
```

```
</html>
```

FORMAS DE AÑADIR CSS

ESTILOS INTERNOS (Definir la hoja de estilos dentro del propio documento HTML usando la etiqueta `<style>` dentro de la sección `<head>` del documento)

Los estilos se definen en una zona específica del propio documento HTML. Se emplea la etiqueta `<style>` de HTML y solamente se pueden incluir en la cabecera del documento (sólo dentro de la sección `<head>`).

Este método se emplea cuando se define un número pequeño de estilos o cuando se quieren incluir estilos específicos en una determinada página HTML que completen los estilos que se incluyen por defecto en todas las páginas del sitio web.

El principal inconveniente es 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.

Se puede colocar cualquier número de elementos `STYLE` en la cabecera del documento. Si un navegador no soporta hojas de estilo o no soporta el lenguaje de hojas de estilo utilizado en `STYLE`, no deberá mostrar el contenido del elemento. Algunos lenguajes de hojas de estilo soportan recursos que permiten ocultar al navegador la información de estilo si este no la soporta.

FORMAS DE AÑADIR CSS

EJEMPLO ESTILOS INTERNOS

```
<!DOCTYPE html>
```

```
<html lang="es">
```

```
<head>
```

```
    <title>Ejemplo de estilos CSS en el propio documento</title>
```

```
    <style type="text/css">
```

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

```
    </style>
```

```
</head>
```

```
<body>
```

```
    <p>Un párrafo de texto.</p>
```

```
</body>
```

```
</html>
```


FORMAS DE AÑADIR CSS

ESTILOS EXTERNOS (Enlazar un archivo externo, para ello necesitamos la etiqueta `<link>` que ubicaremos dentro de la etiqueta `<head>`)

Todos los estilos CSS se incluyen en un archivo de tipo CSS que las páginas HTML enlazan mediante la etiqueta `<link>`. Un archivo de tipo CSS no es más que un archivo simple de texto cuya extensión es `.css`. Se pueden crear todos los archivos CSS que sean necesarios y cada página HTML puede enlazar tantos archivos CSS como necesite.

Si se quiere incluir los estilos del ejemplo anterior en un archivo CSS externo, se deben seguir los siguientes pasos:

1. Se crea un archivo de texto con la extensión `.css` desde el programa de desarrollo un archivo tipo **CSS**, y se le añade solamente el siguiente contenido:
`p { color: black; font-family: Verdana; }`
2. Se debe poner especial atención a que el archivo tenga **extensión `.css`**.
3. En la página **HTML** se enlaza el archivo **CSS** externo mediante la etiqueta `<link>`.

FORMAS DE AÑADIR CSS

EJEMPLO ESTILOS EXTERNOS

```
<!DOCTYPE html>
```

```
<html lang="es">
```

```
<head>
```

```
    <title>Ejemplo de estilos CSS en un archivo externo</title>
```

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

```
</head>
```

```
<body>
```

```
    <p>Un párrafo de texto.</p>
```

```
</body>
```

```
</html>
```

Archivo estilos.css

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

FORMAS DE AÑADIR CSS

ESTILOS EXTERNOS

De todas las formas de incluir **CSS** en las páginas **HTML**, esta es la más utilizada con mucha diferencia. La principal ventaja es que se puede incluir un mismo archivo **CSS** en multitud de páginas **HTML**, por lo que se garantiza la aplicación homogénea de los mismos estilos a todas las páginas que forman un sitio web.

Con este método, el mantenimiento del sitio web se simplifica al máximo, ya que un solo cambio en un solo archivo **CSS** permite variar de forma instantánea los estilos de todas las páginas **HTML** que enlazan ese archivo.

El grado más alto de flexibilidad y eficacia se logra mediante las hojas de estilo externas, es decir, definiendo las reglas de estilo en un archivo externo al documento. De este modo puede cambiarse la apariencia de todo un conjunto de páginas con solo cambiar la hoja de estilo externa.

FORMAS DE AÑADIR CSS

Las tres formas pueden utilizarse para un mismo documento.

En general, utilizar una **hoja de estilo externa** al documento, es el **más racional** pues permite dar una **apariencia homogénea** a **todas las páginas** que se integran en un sitio web y permite realizar todos los **cambios necesarios sin reescribir el código fuente de cada página**.

Si se opta por incrustar la información de **estilo en la cabecera** del documento, su **alcance estará limitado al cuerpo** de ese mismo documento.

Si se introduce la información de **estilo en línea**, es decir, aplicándola a elementos puntuales del cuerpo de un documento, el **alcance estará limitado a ese elemento** de ese documento.

3.1.1 ORDEN DE APLICACIÓN



ORDEN DE APLICACIÓN

Si repito la misma regla en el archivo externo, en la cabecera y en el párrafo, siempre se ejecutará la orden que este más cerca del elemento. Por ejemplo:

*Si en el archivo **estilos.css** escribo*

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

*en el **<style>** de la cabecera escribo*

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

*y en el **párrafo** utilizo el atributo*

```
style="{color:orange;}
```

*El párrafo se pintará de **color orange**, si quisiera que fuera **blue**, tendría que añadir la palabra **!important** y quedaría así:*

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

ORDEN DE APLICACIÓN

Sea cual sea el método o métodos utilizados, el autor deberá especificar el lenguaje de estilo por defecto para la información de estilo asociada al documento.

Esto puede hacerse mediante la utilización del elemento **meta**.

Por ejemplo, para especificar que el lenguaje de estilo por defecto es **CSS** el autor deberá incluir en la cabecera del documento la siguiente declaración:

```
<meta http-equiv="Content-Style-Type" content="text/css">
```

ORDEN DE APLICACIÓN

En las hojas de estilos complejas, es habitual que varias reglas **CSS** se apliquen a un mismo elemento **HTML** el método genérico seguido por **CSS** para resolver las colisiones:

1. Determinar todas las declaraciones que se aplican al elemento para el medio **CSS** seleccionado.
2. Ordenar las declaraciones según su origen (**CSS** de navegador, de usuario o de diseñador) y su importancia (palabra clave !important).
3. Ordenar las declaraciones según lo específico que sea el selector. Cuanto más genérico es un selector, menos importancia tienen sus declaraciones.
4. Si después de aplicar las normas anteriores existen dos o más reglas con la misma prioridad, se aplica la que se indicó en último lugar.
5. Si dos o más hojas CSS modifican la misma propiedad, se impone la hoja que se ha añadido en último lugar.

EJEMPLOS

Ejemplos estilos.css:

```
h2 {color: orange;  
}
```

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

```
.destacado {color: darkblue;  
}
```

```
#miparrafo {color: pink;  
}
```

Este es el título de la página 1

Este es un párrafo de la página 1

Este es otro párrafo de la página 1

Ahora añadido otro párrafo

Este es el subtítulo de la página

EJEMPLOS

Ejemplos pagina1.html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Página 1 del sitio</title>
  <meta http-equiv="Content-Style-Type" content="text/css">
  <link rel="stylesheet" type="text/css" href="estilos.css" />
  <style type="text/css">
    h1 {color: red;}
    p {color: green;}
  </style>
</head>
<body>
  <h1> Este es el <span class="destacado"> titulo </span> de la página 1 </h1>
  <p id="miparrafo"> Este es un párrafo de la página 1 </p>
  <p style="color:red;"> Este es otro párrafo de la página 1 </p>
  <p class="destacado"> Ahora añadido otro párrafo </p>
  <h2> Este es el subtítulo de la página </h2>
</body>
</html>
```

Título 1

Título 2

Párrafo 1

Párrafo 2

EJEMPLOS

Ejemplos pagina2.html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Página 2 del sitio</title>
  <meta http-equiv="Content-Style-Type" content="text/css">
  <link rel="stylesheet" type="text/css" href="estilos.css" />
</head>
<body>
  <h1> Título 1 </h1>
  <h2> Título 2 </h2>
  <p> Párrafo 1 </p>
  <p> Párrafo 2 </p>
</body>
</html>
```

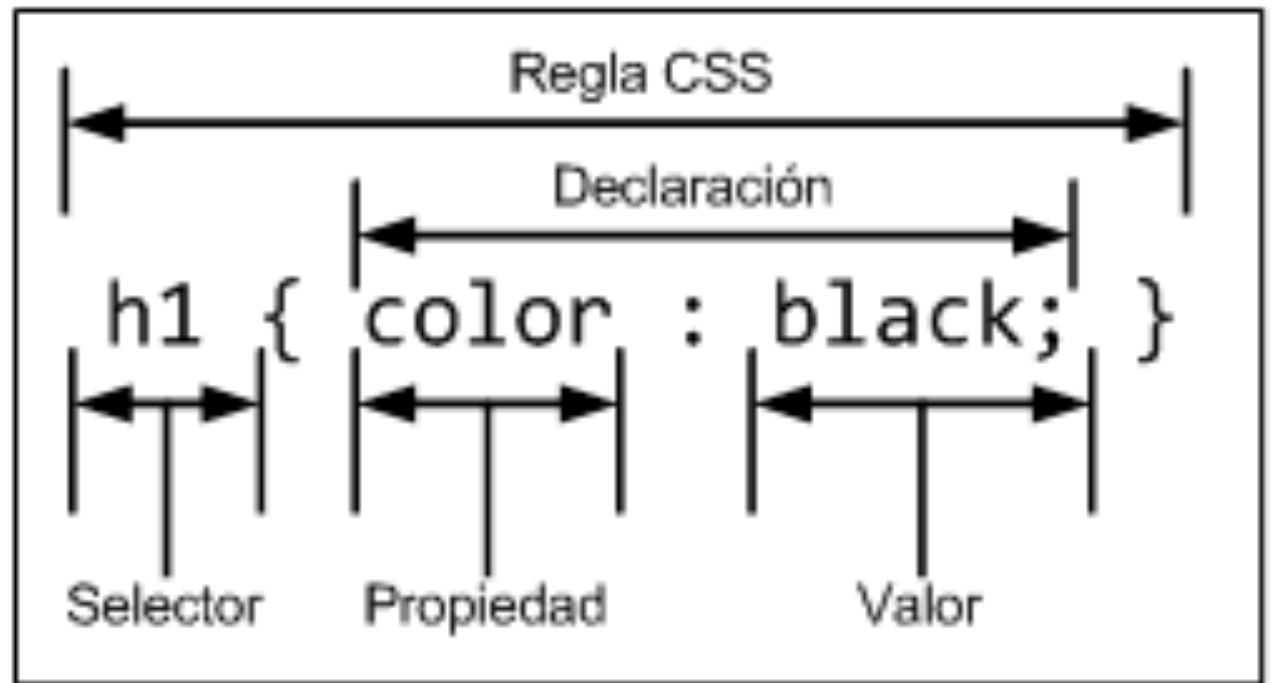
3.2 REGLAS CSS



REGLA CSS

Regla: cada uno de los estilos que componen una hoja de estilos CSS. Cada regla está compuesta de una parte de "selectores", un símbolo de "llave de apertura" ({), otra parte denominada "declaración" y, por último, un símbolo de "llave de cierre" (}).

Selector: indica el elemento o elementos HTML a los que se aplica la regla CSS.



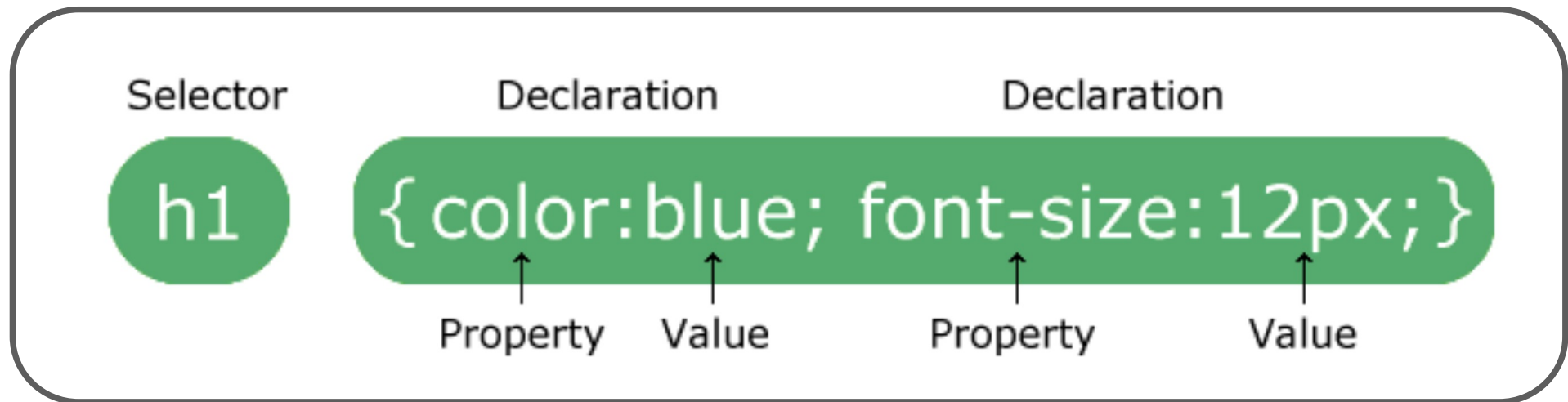
Declaración: especifica los estilos que se aplican a los elementos. Está compuesta por una o más propiedades CSS.

Propiedad: permite modificar el aspecto de una característica del elemento.

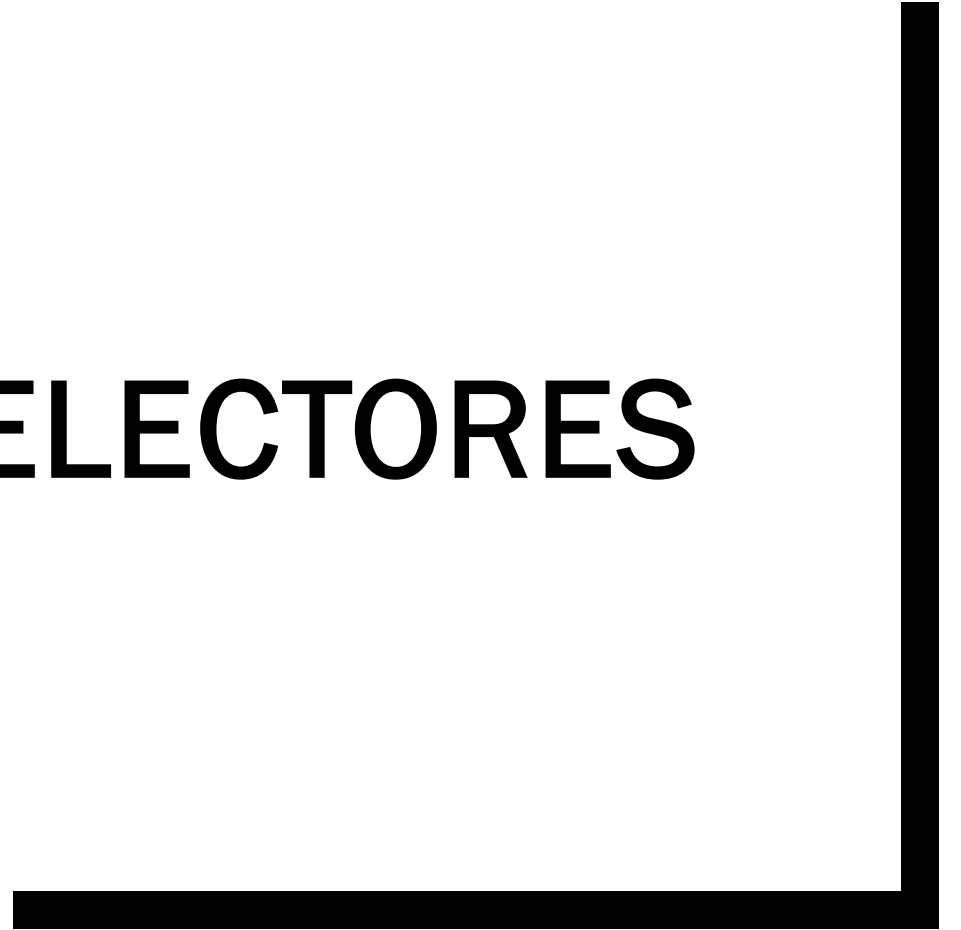
Valor: indica el nuevo valor de la característica modificada en el elemento.

selector { propiedad: valor; propiedad2: valor; }

CSS Rule



3.3 SELECTORES



3.3.1 SELECTORES SIMPLES



SELECTORES SIMPLES

SELECTOR DE ID (IDENTIFICADOR)

Este tipo de selectores tiene dos partes, tenemos que preparar primero el código **HTML** y luego explicar la apariencia en **CSS**. Este tipo de selectores solamente seleccionan un elemento de la página porque el valor del atributo **id** no se puede repetir en dos elementos diferentes de una misma página.

En **HTML** incluiremos un atributo **id** en la etiqueta a la que queramos cambiar el aspecto, por ejemplo:

```
<p id="enelmedio">Mi párrafo </p>
```

En **CSS** definiremos como será el aspecto del identificador "enelmedio" y lo hacemos de la siguiente manera:

```
#enelmedio {color:red;}
```

*La diferencia entre **clase** e **identificador** es que se supone que utilizaremos “**clase**” cuando sea una apariencia que queramos a aplicar a diferentes elementos y utilizaremos “**identificador**” cuando queramos aplicarlo a un único elemento. Es como si en el primer caso se lo aplicáramos a todos los alumnos y en el segundo al alumno que se llama Pepe.*

SELECTORES SIMPLES

SELECTOR UNIVERSAL

Se utiliza para seleccionar todos los elementos de la página.

Ejemplo **CSS**:

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

El selector universal se indica mediante un asterisco (*).

El selector universal (*), es útil cuando deseas aplicar un estilo a todos los elementos en una página web sin excepción. Es una forma rápida y sencilla de establecer reglas generales que afectarán a todos los elementos en el documento. Por ejemplo, si deseas eliminar todos los márgenes y rellenos predeterminados de los elementos en tu página para empezar desde cero, puedes usar el selector universal para restablecerlos.

SELECTORES SIMPLES

SELECTOR ETIQUETAS

Podemos utilizar cualquiera de las etiquetas de **HTML**, y lo que hará la regla es que cambiará de golpe el aspecto de todas las etiquetas **h1** de la página.

Por ejemplo, si pongo **h1 {color: blue;}** hará que todos los encabezados 1 de la página sean de color azul.

Si se quiere aplicar los mismos estilos a dos etiquetas diferentes, se pueden encadenar los selectores.

```
h1, h2, h3{color: blue;  
}
```

SELECTORES SIMPLES

SELECTOR DE CLASE

Este tipo de selectores tiene dos partes, tenemos que preparar primero el código **HTML** y luego explicar la apariencia en **CSS**.

En **HTML** incluiremos un atributo en la etiqueta a la que queramos cambiar el aspecto, por ejemplo:

```
<p class="nombre">Mi párrafo </p>
```

En **CSS** definiremos como será el aspecto de la clase "**nombre**", para que el navegador no confunda este selector con los otros tipos de selectores, se prefija el valor del atributo class con un punto (.) y lo hacemos de la siguiente manera:

```
.nombre {color: green;}
```

NOTA: si solamente quiero cambiar una palabra o unas pocas, no todo el párrafo, utilizaré la etiqueta **** y quedara así:

```
<span class="nombre">sencilla</span>
```

Si lo que quiero es remarcar una etiqueta que tiene el atributo clase lo hare así:

```
th.destacado {color: yellow;}
```

y esto pondrá de amarillo solo las etiquetas **th** que tengan el atributo class="**destacado**"

SELECTORES SIMPLES

Selector	Example	Example description
<u>#id</u>	#firstname	Selects the element with id="firstname"
<u>.class</u>	.intro	Selects all elements with class="intro"
<u>element.class</u>	p.intro	Selects only <p> elements with class="intro"
<u>*</u>	*	Selects all elements
<u>element</u>	p	Selects all <p> elements
<u>element,element,..</u>	div, p	Selects all <div> elements and all <p> elements

3.3.2 SELECTORES COMBINADORES



SELECTORES COMBINADORES

SELECTOR HERMANOS PRÓXIMOS

El selector de hermanos general selecciona todos los elementos que son hermanos y están después del elemento principal y siendo hermanos.

CSS

```
div ~ p {  
    background-color: yellow;  
}
```

HTML

```
<p>Paragraph 1.</p>  
<div>  
    <p>Paragraph 2.</p>  
</div>  
<p>Paragraph 3.</p>  
<code>Some code.</code>  
<p>Paragraph 4.</p>
```

Paragraph 1.

Paragraph 2.

Paragraph 3.

Some code.

Paragraph 4.

SELECTORES COMBINADORES

SELECTOR DESCENDENTES

Para indicar que quiero cambiar el aspecto de una etiqueta que está dentro de otra (o de otras) voy poniendo en orden de aparición las etiquetas que lo contienen:

CSS

```
div p code {
```

```
    background-color: yellow;
```

```
}
```

HTML

```
<div>
```

```
    <p>Paragraph 1 in the div.</p>
```

```
    <p>Paragraph 2 in the div. <code>function</code></p>
```

```
    <section><p>Paragraph 3 in the div.</p></section>
```

```
</div>
```

```
<p>Paragraph 4. Not in a div.</p>
```

Paragraph 1 in the div.

Paragraph 2 in the div. **function**

Paragraph 3 in the div.

Paragraph 4. Not in a div.

SELECTORES COMBINADORES

SELECTOR DE HIJOS

Selector similar al selector descendente, pero diferente en su funcionamiento. Se utiliza para **seleccionar un elemento que es hijo directo de otro elemento** y se indica mediante el "signo de mayor que" (>). En el **selector de hijos** el elemento debe **ser hijo directo** de otro elemento. Mientras que en el selector descendente solamente importa que un elemento esté dentro de otro, independientemente de lo profundo que se encuentre.

CSS

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

HTML

```
<p><span>Texto1</span></p>
```

```
<p> <a href="#"> <span>Texto2</span> </a></p>
```

En el ejemplo anterior, el selector `p > span` se interpreta como *“cualquier elemento `` que sea hijo directo de un elemento `<p>`”*, por lo que el primer elemento `` cumple la condición del selector. Sin embargo, el segundo elemento `` no la cumple porque es descendiente pero no es hijo directo de un elemento `<p>`.

SELECTORES COMBINADORES

SELECTOR ADYACENTE

El selector adyacente se emplea para seleccionar elementos que son *hermanos* (su elemento padre es el mismo) y están seguidos en el código HTML.

CSS

```
h1 + h2 { color: red }
```

HTML

```
<body>
  <h1>Titulo1</h1>
  <h2>Subtítulo</h2>
  <h2>Subtítulo 2</h2>
  <h2>Subtítulo 3</h2>
  ...
  <h2>Otro subtítulo</h2>
</body>
```

Titulo1

Subtítulo

Subtítulo 2

Subtítulo 3

...

Otro subtítulo

Los estilos del selector `h1 + h2` se aplican al primer elemento `<h2>` de la página, pero no al segundo `<h2>`.

SELECTORES COMBINADORES

Selector	Example	Example description
<u><i>element element</i></u>	div p	Selects all <p> elements inside <div> elements
<u><i>element>element</i></u>	div > p	Selects all <p> elements where the parent is a <div> element
<u><i>element+element</i></u>	div + p	Selects the first <p> element that are placed immediately after <div> elements
<u><i>element1~element2</i></u>	p ~ ul	Selects every element that are preceded by a <p> element

3.3.3 SELECTORES ATRIBUTOS



SELECTORES ATRIBUTOS

Permiten seleccionar elementos **HTML** en función de sus atributos y/o valores de esos atributos.

[*attr*] Selecciona los elementos que **tienen** el atributo ***attr***.

*/** Elementos **<a>** con un atributo **title** **/*

a[title] { color: purple; }

*/** Elementos **<a>** con un atributo **class** **/*

a[class] {color: purple; }

[*attr=value*] Selecciona los elementos cuyo atributo ***attr*** tenga **exactamente** el valor ***value***.

*/** Elementos **<a>** con un **href** que coincida con **"https://example.org"** **/*

a[href="https://example.org"] { color: green; }

*/** Elementos **<a>** con un **class** que coincida con **"externo"** **/*

a[class="externo"] {color: purple; }

SELECTORES ATRIBUTOS

[*attr*~=*value*] Selecciona los elementos cuyo atributo *attr* tenga por valor una lista de palabras **separadas por espacios**, una de las cuales sea *value*. **class="logo empresa"**

/ Elementos <a> cuyo atributo class contenga la palabra "logo" */*

```
a[class~="logo"] { padding: 2px; }
```

[*attr*|=*value*] Selecciona los elementos cuyo atributo *attr* tenga exactamente el valor *value* o empiece por *value* seguido de un **guión -**. Se puede usar para coincidencias de subcódigos en otros idiomas.

```
*[lang|= "es"] { color : red }
```

[*attr*^=*value*] Selecciona los elementos cuyo atributo *attr* tenga un valor **prefijado** por *value*.

/ Elementos <a> con un href que comience con "#" */*

```
a[href^="#"] { color: #001978; }
```

SELECTORES ATRIBUTOS

[attr\$=value] Selecciona los elementos cuyo atributo *attr* cuyo valor tiene el sufijo (seguido) de *value*.

```
/* Elementos <a> con un href que termine en ".org" */  
a[href$=".org"] { font-style: italic; }
```

[attr*=value] Selecciona los elementos cuyo atributo *attr* tenga un valor que contenga *value*.

```
/* Elementos <a> con un href que contenga "example" */  
a[href*="example"] { font-size: 2em; }
```

SELECTORES ATRIBUTOS

Selector	Example	Example description
<code>[<u>attribute</u>]</code>	<code>[target]</code>	Selects all elements with a target attribute
<code>[<u>attribute=value</u>]</code>	<code>[target="_blank"]</code>	Selects all elements with target="_blank"
<code>[<u>attribute~=value</u>]</code>	<code>[title~="flower"]</code>	Selects all elements with a title attribute that contains a space-separated list of words, one of which is "flower"
<code>[<u>attribute =value</u>]</code>	<code>[lang]="en"]</code>	Selects all elements with a lang attribute value starting with "en"
<code>[<u>attribute^=value</u>]</code>	<code>a[href^="https"]</code>	Selects all <a> elements with a href attribute value starting with "https"
<code>[<u>attribute\$=value</u>]</code>	<code>a[href\$=".pdf"]</code>	Selects all <a> elements with a href attribute value ending with ".pdf"
<code>[<u>attribute*=value</u>]</code>	<code>a[href*="w3schools"]</code>	Selects all <a> elements with a href attribute value containing the substring "w3schools"

SELECTORES ATRIBUTOS

Atributo	Descripción
<code>[href]</code>	El atributo <code>href</code> existe en la etiqueta.
<code>[href="#"]</code>	El atributo <code>href</code> existe y su valor es igual al texto <code>#</code> .
<code>[href*="eniun"]</code>	El atributo <code>href</code> existe y su valor contiene el texto <code>eniun</code> .
<code>[href^="https://"]</code>	El atributo <code>href</code> existe y su valor comienza por <code>https://</code> .
<code>[href\$=".pdf"]</code>	El atributo <code>href</code> existe y su valor termina por <code>.pdf</code> (es un enlace a un PDF).
<code>[class~="eniun"]</code>	El atributo <code>class</code> contiene una lista de valores, que contiene <code>eniun</code> .
<code>[lang]="es"]</code>	El atributo <code>lang</code> contiene una lista de valores, donde alguno empieza por <code>es-</code> .

3.3.4 COMBINACIÓN SELECTORES



COMBINACIÓN SELECTORES

CSS permite la combinación de uno o más tipos de selectores para restringir el alcance de las reglas **CSS**.

Ejemplos:

.aviso .especial { ... }

div.aviso span.especial { ... }

ul#menuPrincipal li.destacado a#inicio { ... }

T3-P1-selectores