



Módulo: HTML y CSS

INTRODUCCIÓN A CSS

A thick, vibrant red diagonal stripe runs from the top right corner towards the bottom left, dividing the white background.

1.

¿QUÉ SIGNIFICA
CSS?

CSS

Cascading
Style
Sheets

Hojas de **estilo** en cascada.

Compuestas de REGLAS,
SELECTORES y
DECLARACIONES.

¿Y PARA QUÉ NOS
SIRVEN LAS
HOJAS DE ESTILO?

```
<div class="portfolio app">  
  <div class="hover-bg">  
    <a href="#">  
      <div class="hover-text">  
        <h4>Logo Design</h4>  
        <small>Branding</small>  
      </div>  
    </div>  
  </div>  
  <div class="clearfix"></div>  
  <i class="fa fa-plus"></i>  
</div>  
  
```

```
192 .mfp-arrow:after, .mfp-arrow .mfp-a {  
193   border-top-width: 13px;  
194   border-bottom-width: 13px;  
195   top: 8px; }  
196 .mfp-arrow:before, .mfp-arrow .mfp-b {  
197   border-top-width: 21px;  
198   border-bottom-width: 21px;  
199   opacity: 0.7; }  
200 .mfp-arrow-left {  
201   left: 0; }  
202 .mfp-arrow-left:after, .mfp-arrow-left .mfp-a {  
203   border-right: 17px solid #FFF;  
204   margin-left: 31px; }  
205 .mfp-arrow-left:before, .mfp-arrow-left .mfp-b {  
206   margin-left: 25px;  
207   border-right: 27px solid #3F3F3F; }  
208 .mfp-arrow-right {  
209   right: 0; }  
210 .mfp-arrow-right:after, .mfp-arrow-right .mfp-a {  
211   border-left: 17px solid #FFF;  
212   margin-left: 39px; }  
213 .mfp-arrow-right:before, .mfp-arrow-right .mfp-b {  
214   border-left: 27px solid #3F3F3F; }  
215  
216 .mfp-iframe-holder {  
217   padding-top: 40px;  
218   padding-bottom: 40px; }
```

¡DEFINAMOS!

Las hojas de estilo sirven para **estilizar** nuestro contenido **HTML**. Con **CSS** podemos cambiar colores, fondos, tipografías, anchos, altos, etc. Así como también generar animaciones y transiciones.

Contamos con **3 métodos** para vincular nuestros archivos CSS con el documento HTML:

VINCULACIÓN INTERNA

A través de la etiqueta `<style>` dentro del `<head>`.

VINCULACIÓN EN LÍNEA

Usando el atributo **style** en cada elemento de nuestro HTML.

```
<p style="color:red"></p>
```

Y LA MÁS USADA...

VINCULACIÓN EXTERNA

Escribiendo todos nuestros estilos en un archivo CSS y vinculándolos, usando la etiqueta `<link>` dentro del `<head>` de nuestro documento.

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



valor

Ruta de la ubicación de mi hoja de estilo



rel

Indica qué relación hay entre los documentos a enlazar



valor

Este valor siempre es el mismo.

2.

SELECTORES DE CSS

Los **selectores** nos van a permitir “atrapar” a los elementos HTML que queramos modificar y así **aplicarles nuestros estilos.**

SELECTORES DE CSS

Los **selectores** nos van a permitir “atrapar” a los elementos HTML que queramos modificar y así aplicarles nuestros estilos.

ID `<p id="saludo"></p>`

A través del atributo `id` le asignamos un nombre al elemento.

Sintaxis: `#saludo` (se usa el `#` seguido del nombre)

CLASE `<h1 class="títulos"></h1>`

A través del atributo `class` le asignamos un nombre al elemento.

Sintaxis: `.títulos` (se usa el `.` seguido del nombre)

ETIQUETA `<u1></u1>`

Afecta a la etiqueta que nombremos en nuestro CSS.

Sintaxis: `ul` (se usa el nombre de la etiqueta)

SELECTORES DE CSS

COMBINADOS

Afectan a los que cumplan todas las condiciones.

Ejemplo: `p.saludo` (estamos seleccionando los `<p>` que tengan asignada la clase `saludo`)

DESCENDENTES

Sirven para agregar especificidad. Estos se utilizan con un espacio.

Ejemplo: `ul li.especial` (estamos seleccionando los `` dentro de un `` que tengan asignada la clase `especial`)

SELECTORES DE ID

Este selector va a atrapar al elemento HTML que tenga asignado el **atributo ID**. Se recomienda usar nombres únicos para cada elemento y no repetirlos a lo largo del documento.

```
<h3 id="saludo">¡Hola!</h3>
```

SINTAXIS

```
#saludo
```

Para llamarlo desde el css usamos el # seguido del nombre del ID.

EJEMPLO

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

SELECTORES DE CLASE

Este selector va a atrapar al elemento HTML que tenga asignado el **atributo CLASS**. Podemos asignarle la cantidad de clases que queramos a un mismo elemento. Para hacerlo, sólo hace falta separarlas con un espacio.

```
<h1 class="titulo noticias">Noticias</h1>
```

SINTAXIS

.titulo

Para llamarlo desde el css usamos el **.** seguido del **nombre de la CLASE**.

EJEMPLO

```
.titulo {  
    font-size: 22px;  
}
```

SELECTORES DE ETIQUETA

Este selector va a atrapar al elemento HTML con el mismo nombre de etiqueta que llamemos desde nuestro CSS.

`<p>Párrafo uno</p>`

SINTAXIS

`p`

Para llamarlo desde el css usamos el **nombre de la ETIQUETA**.

EJEMPLO

```
p {  
    color: gray;  
}
```

SELECTORES COMBINADOS

Estos selectores afectan a el/los elemento/s que cumplan con todas las condiciones que establezca. En el siguiente ejemplo, vamos a atrapar al elemento **h2** que tenga asignada la clase **subtitulo**.

```
<h2 class="subtitulo">Subtítulo</h2>
```

SINTAXIS

h2.subtitulo

Para llamarlos desde el css sólo hace falta **agregar un selector al lado del otro**, cada uno con la sintaxis que le corresponda.

EJEMPLO

```
h2.subtitulo {  
    color: gray;  
}
```

SELECTORES DESCENDENTES

Estos selectores sirven para agregar especificidad. En el ejemplo vamos a atrapar al elemento **li** que esté dentro del **ul** con el id **lista**.

```
<ul id="lista">
  <li>Primer ítem</li>
</ul>
```

SINTAXIS

```
ul#saludo li
```

Para llamarlos desde el css escribimos los **selectores separados por un espacio** (esto indica paternidad)

EJEMPLO

```
ul#saludo li {
    text-align: center;
}
```

DATO IMPORTANTE



El CSS siempre va a priorizar a los selectores **más específicos** para aplicar los estilos.