

Diplomatura

FullStack

Developer

Clase 4: **Introducción Css. Sintaxis y Selectores (básicos y combinados)**

CSS (Cascading Style Sheets) es un lenguaje de hojas de estilo utilizado para **describir la presentación de un documento escrito en HTML o XML.**

CSS **controla cómo se muestran los elementos en la página web.**



Con Css podemos definir reglas

Las reglas se aplican selectivamente a los elementos a través de selectores, lo que te permite tener un control preciso sobre el estilo de cada elemento en una página.



Adjuntar estilos

Usamos un **archivo externo**, generando una carpeta de archivo con extensión **.css**.

Entonces en el html hacemos uso de la etiqueta **<link>**, dentro del **<head>** donde se encuentras los datos para el navegador, y no tiene etiqueta de cierre, solo un **>** final

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



<link> Atributos

- **rel** describe la relación entre dos documentos, hablamos del Html actual y otro **.css** que queramos enlazar.
El nombre por defecto es “stylesheet”

- **href**
especifica la ubicación del archivo

Selectores

Básicos

Selector de tipo:

selecciona todos los elementos que coinciden con el nombre del elemento especificado.

Por ejemplo si queremos dar estilo a los elementos del tipo input

En css → `input{`
`}`

Selector de clase:

En Html hay un atributo que poseen todos los elementos y es el “class”, que se usa para ponerle nombre al o los elementos. Es decir que puedo modificar más de uno a la vez.

<p class="subTitle">Este es el elemento en cuestión</p>

En css→

.subTitle{
}

Selectores

Básicos

Selector de id:

es un selectos de elementos pero **solo se aplica a uno de ellos a la vez.**

```
<p id="subTitle1">Elemento</p>  
<p id="subTitle2">Elemento</p>
```

En css → `#subTitle{`
 }
}

Selector universal:

es aquel que selecciona todos los elementos.

Opcionalmente puede estar restringido a un espacio de nombre específico o a todos los espacios de nombres.

En css →

```
*{  
}
```

Selectores

Combinadores

Combinador general de hermanos:

Se usa el símbolo `~` y selecciona hermanos → el segundo elemento sigue al primero, y ambos heredan del mismo elemento padre.

```
p ~ span{  
    font-size: 20 px;  
}
```

¿Cómo se interpreta?

El elemento afectado será **span**, que está después de un elemento **<p>** y que ambos son hijos de un mismo elemento **<div>**

Selectores Combinadores

Combinador de hijo:

se usa el combinador > que selecciona todos los elementos que son hijos directos del primer elemento.

En el ejemplo se seleccionarán todos los elementos de la clase contenedor

en CSS →

```
.contenedor>p{  
}
```

Selectores

Pseudo Classes

pseudoclasse es una
palabra clave

se añade a los selectores y con
esto especifica un estado
especial

Primero va el **selector**, **:**, la
palabra clave y **:**

en css→

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

Pseudo Classes: descripciones

<code>:active</code>	Se aplica cuando el elemento está siendo activado por el usuario.
<code>:checked</code>	Se aplica a elementos de entrada (input) cuando están seleccionados o marcados.
<code>:disabled</code>	Se aplica a elementos desactivados.
<code>:empty</code>	Se aplica a elementos que no contienen ningún hijo o texto.
<code>:enabled</code>	Se aplica a elementos habilitados.
<code>:first-child</code>	Se aplica al primer hijo de un elemento padre.
<code>:first-of-type</code>	Se aplica al primer elemento de un tipo específico dentro de un elemento padre.
<code>:focus</code>	Se aplica cuando el elemento tiene el enfoque.
<code>:hover</code>	Se aplica cuando el cursor del ratón se encuentra sobre el elemento.
<code>:last-child</code>	Se aplica al último hijo de un elemento padre.
<code>:last-of-type</code>	Se aplica al último elemento de un tipo específico dentro de un elemento padre.

Pseudo Classes: descripciones

<code>:not(selector)</code>	Se aplica a elementos que no coinciden con el selector especificado.
<code>:nth-child(n)</code>	Se aplica a elementos que son el enésimo hijo de su padre.
<code>:nth-last-child(n)</code>	Se aplica a elementos que son el enésimo hijo contando desde el final de su padre.
<code>:nth-last-of-type(n)</code>	Se aplica a elementos que son el enésimo elemento de un tipo específico contando desde el final de su padre.
<code>:nth-of-type(n)</code>	Se aplica a elementos que son el enésimo elemento de un tipo específico dentro de su padre.
<code>:only-child</code>	Se aplica a elementos que son el único hijo de su padre.
<code>:only-of-type</code>	Se aplica a elementos que son el único elemento de un tipo específico dentro de su padre.

Pseudo Classes: descripciones

<code>:placeholder-shown</code>	Se aplica a elementos de entrada (input) cuando su texto de marcador (placeholder) está visible.
<code>:read-only</code>	Se aplica a elementos de entrada (input) que tienen el atributo "readonly" especificado.
<code>:read-write</code>	Se aplica a elementos de entrada (input) que no tienen el atributo "readonly" especificado.
<code>:required</code>	Se aplica a elementos de entrada (input) que tienen el atributo "required" especificado.
<code>:target</code>	Se aplica cuando el elemento es el destino de una URL en la barra de direcciones.
<code>:valid</code>	Se aplica a elementos de entrada (input) que contienen datos válidos.
<code>:visited</code>	Se aplica a enlaces visitados.

Pseudo Classes: sintaxis

1. Selectores:

se utilizan para seleccionar los elementos del html a los que se les dará estilo. Pueden ser

- atributos específicos → p, h1,
- clases → .clase o
- identificadores → #id

2. Propiedades

son los atributos que definen cómo se verá un elemento seleccionado. Tiene un valor asociado que determinará cómo se aplicará la propiedad.

Por ejemplo:

- color
- font-size
- background-color

Pseudo Classes: sintaxis

3. Valores:

son los posibles ajustes que se pueden aplicar a una propiedad.

Por ejemplo, en el color se puede definir de 3 forma

- **nombre del color** → green, yellow
- **códigos hexadecimales** → #FF0000
- **valores RGB** → rgb[255,0,0]

4.Reglas

están compuestas por un selector y un conjunto de declaraciones rodeadas por **{ }**

- Las declaraciones son propiedades y valores
- La regla determinará como se verán esos elementos.

5. Comentarios: se utiliza **/*** y ***/** sirven para explicar el código

Pseudo Classes: propiedades

color	Establece el color del texto.
font-size	Define el tamaño de la fuente del texto.
font-family	Especifica la fuente utilizada para el texto.
background-color	Establece el color de fondo de un elemento.
width	Define el ancho de un elemento.
height	Establece la altura de un elemento.
margin	Controla los márgenes externos de un elemento.
padding	Define el relleno interno de un elemento.
border	Establece el estilo, el ancho y el color del borde de un elemento.
display	Define cómo se muestra un elemento en el flujo del documento.
position	Controla la posición de un elemento en relación con su contenedor.
float	Permite que un elemento flote hacia la izquierda o la derecha.
text-align	Alinea el contenido de texto dentro de un elemento.
text-decoration	Agrega decoración al texto, como subrayado o tachado.
box-shadow	Agrega una sombra alrededor de un elemento.

Próxima clase: Box Model

