



# Introducción a CSS

Cordero Hernández,  
Marco R.



Siguiente >>



## En sesiones pasadas...

- HTML
  - Formularios
  - iFrames
  - HTML semántico





## En esta sesión

**01**

Introducción a  
**CSS**

**02**

Formas de  
**CSS**

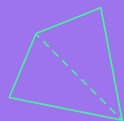
**03**

Sintaxis

**04**

Selectores y  
**combinadores**

Siguiente >>



# 01

## INTRODUCCIÓN A CSS





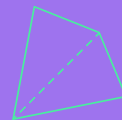
# ¿Qué es CSS?

CSS = **C**ascading **S**tyle **S**heets

Lenguaje que permite describir la presentación de documentos *HTML* = Estilizado.

- No es un lenguaje de programación, es un lenguaje de hojas de estilo
- Le añade “estilo” a la página web, ej: tipo de letra, fondo de pantalla. Organización de la pantalla, efectos visuales, animaciones, comportamientos, etcétera
- Permite **adaptar** el contenido al dispositivo





# 02

FORMAS DE CSS





# Inline

De forma directa dentro de etiquetas/elementos de HTML mediante el atributo *style*.

```
<!-- Inline -->  
<p style="color: ■ white; background-color: ■ darkcyan; text-align:center;">  
    Esto es un párrafo  
</p>
```

Esto es un párrafo



# Inline

*Tiene la más alta precedencia sobre los otros métodos.*

## Desventajas

- No permite reusar los estilos (se deben copiar y pegar)
- Al hacer modificaciones es necesario hacerlo en cada etiqueta
- Puede ser lento el proceso de búsqueda de cada atributo para editarlos
- No es posible añadir estilos para algunos “estados” como cuando el mouse se posiciona sobre un elemento (pseudo-clases no son posibles).





# Interno

Dentro del mismo documento a través de la etiqueta `<style>` dentro de la sección de `<head>`.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, ...>
  <title>Sesión 6 - Formas de añadir estilos</title>

  <!-- Interno -->
  <style>
    div p {
      color: purple;
      background-color: lime;
      text-align: center;
    }
  </style>
</head>
```

## Ventajas

- Permite añadir estilos uniformes a todos los elementos del documento
- Es posible añadir estilos a estados de los elementos

## Desventajas

- Solo modifica un documento
- Se deben copiar hacia todos los documentos sobre los cuales se quieran usar



# Externo

Dentro del mismo documento a través de la etiqueta `<style>` dentro de la sección de `<head>`.

```
formas.html U x
Sesiones > 02 - CSS > ses6 (12-02) > formas.html > html > body > section > p
17      <!-- Externo -->
18      <link rel="stylesheet" href="estilos.css">
19  </head>
20  <body>
21  <!-- Inline -->

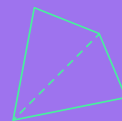
estilos.css U x
Sesiones > 02 - CSS > ses6 (12-02) > estilos.css > ...
1  section p {
2      color: lightblue;
3      background-color: black;
4      text-align: right;
5  }
```

## Ventajas

- Permite añadir estilos uniformes a todos los elementos del documento
- También es posible añadir estilos a estados
- Aplicable a muchos documentos
- Fácil de mantener (documentos)

## Desventajas

- No permite un control más específico para ciertos elementos



# 03

SINTAXIS





# Sintaxis básica - Reglas

```
/* REGLA */  
/* Inicio de declaración */  
p { /* Selector */  
    /* Propiedad y valor */  
    color: ■ white;  
    text-align: center;  
}  
/* Fin de declaración */
```

**Selector:** describe los elementos a seleccionar para añadirles estilo.

**Propiedad:** formas de dar estilo a los elementos HTML

**Valor:** asigna el valor a la propiedad

**Declaración:** Especifica a cuál de las propiedades se les añadirá estilo y su valor



# Sintaxis básica - Reglas

- Cada regla se encapsula entre { corchetes }
- Los dos puntos : permiten separar la propiedad y su valor dentro de una declaración
- El punto y coma ; separa una declaración de otra
- Los comentarios se definen de una sola forma usando

**`/*`**

Diagonales y asteriscos (estilo C y Java)

**`*/`**



# 04

## SELECTORES





# Selectores básicos

- **Selector universal (\*):** Selecciona todos los elementos
- **De tipo o etiqueta:** Selecciona todos los elementos de HTML del tipo indicado

```
h1 {  
  ...  
}
```

```
p {  
  ...  
}
```

```
a {  
  ...  
}
```



# Selectores básicos

- **Por identificador:** Selecciona el elemento que tiene el ID especificado *(solo puede ser un elemento)*

```
<p id="identificador"> Párrafo </p>
```

```
#identificador {  
    background-color: blue;  
}
```





# Selectores básicos

- **Por clase:** Selecciona a *todos los elementos* que tengan la clase indicada

```
<h1 class="principal"> Nota importante </h1>
```

```
<p class="principal"> Se anda quemando la primavera 🌸 </p>
```

```
.principal {  
    background-color: red;;  
}
```



## Selectores básicos - ejemplos de propiedades

**color:** Especifica el color del texto

- Colores en inglés: red, blue, navy, orange, white, ...
- `rgb(0, 255, 255) = rgb(0%, 100%, 100%) = #0ff = #00ffff`
- `rgba(r, g, b, a) = rgba(0, 255, 255, 0.5)`
- Hexadecimal
- HSL
- *transparent*

**Referencia:** [https://www.w3schools.com/colors/colors\\_picker.asp](https://www.w3schools.com/colors/colors_picker.asp)



## Selectores básicos - ejemplos de propiedades

### background

- **background-color**
- **background-image:** *url("archivo.png")*
- **background-repeat:** *no-repeat, repeat, repeat-x, repeat-y, no-repeat, initial, inherit*
- **background-position:** *right top, left top, left center, left bottom, right top, right center, right bottom, center center, x% y%, xpos ypos*
- **background-size:** *Xpx Ypx, auto, length, cover, contain, initial, inherit*



## Selectores básicos - ejemplos de propiedades

### font

- **font-family:** family-name, generic-family, family-name, generic-family, initial, inherit

```
h1 {
```

```
    font-family: "Times New Roman", Times, serif;
```

```
}
```

- **font-size:** xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger, Npx, N%, initial, inherit
- **font-weight:** normal (400), bold (700), bolder, lighter, 100-900 (múltiplos de 100), initial, inherit



## Selectores básicos - ejemplos de propiedades

text		
Propiedad	Descripción	Valores
color	Color del texto	<i>color</i>
direction	Dirección del texto	<i>ltr, rtl</i>
line-height	Distancia entre líneas	normal, number, length, %
letter-spacing	Distancia entre caracteres	normal, length
text-align	Alineación del texto	<i>left, right, center, justify</i>
text-decoration	Decoración del texto	<i>none, underline, over, ...</i>



## Selectores básicos - ejemplos de propiedades

text		
Propiedad	Descripción	Valores
text-indent	Agrega indentación a la primera línea del elemento	length, %
text-shadow	Agrega sombra al texto	<i>none</i> , X Y color
text-transform	Modifica las letras del texto	<i>none</i> , capitalize, uppercase, lowercase
white-space	Controla el espacio en blanco dentro del elemento	<i>none</i> , <i>pre</i> , <i>nowrap</i>



## Selectores básicos - ejemplos de propiedades

text		
Propiedad	Descripción	Valores
word-spacing	Aumento o disminuye el espacio entre palabras	<i>none</i> , anchura



# Selectores por atributo

Los elementos en una página que tienen el atributo especificado o con un valor asignado a un atributo.

`<input type="checkbox" name="chk" value="val1 val2" checked>`

- Atributo [attr] → **[checked] {...}**
- Atributo con valor [attr=val] → **[value="val1"] {...}**
- Atributo contiene una lista separada por espacios y contiene una palabra buscada [attr~=val] → **[value~="val2"] {...}**
- Atributo inicia con un valor [attr^=val] → **[value^="v"] {...}**
- Atributo termina con un valor [attr\$=val] → **[value\$="2"] {...}**
- Atributo contiene una subcadena [attr\*=val] → **[value\*="a"] {...}**
- *HTML permite poner atributos definidos por el usuario con **data-**, ejemplo: **data-nota="h4\$\$h-d3-n0t4"***





# Pseudo-clases

Una **pseudo-clase** es una palabra clave que se añade a los selectores que describe *un **estado del elemento seleccionado***.

Ejemplo: **:hover** indica cuando el cursor está *encima* de un elemento.

<a href="#"><u>:active</u></a>	<a href="#"><u>:fullscreen</u></a>	<a href="#"><u>:link</u></a>	<a href="#"><u>:read-only</u></a>
<a href="#"><u>:checked</u></a>	<a href="#"><u>:focus</u></a>	<a href="#"><u>:not()</u></a>	<a href="#"><u>:read-write</u></a>
<a href="#"><u>:default</u></a>	<a href="#"><u>:hover</u></a>	<a href="#"><u>:nth-child()</u></a>	<a href="#"><u>:required</u></a>
<a href="#"><u>:dir()</u></a>	<a href="#"><u>:indeterminate</u></a>	<a href="#"><u>:nth-last-child()</u></a>	<a href="#"><u>:right</u></a>
<a href="#"><u>:disabled</u></a>	<a href="#"><u>:in-range</u></a>	<a href="#"><u>:nth-last-of-type()</u></a>	<a href="#"><u>:root</u></a>
<a href="#"><u>:empty</u></a>	<a href="#"><u>:invalid</u></a>	<a href="#"><u>:nth-of-type()</u></a>	<a href="#"><u>:scope</u></a>
<a href="#"><u>:enabled</u></a>	<a href="#"><u>:lang()</u></a>	<a href="#"><u>:only-child</u></a>	<a href="#"><u>:target</u></a>
<a href="#"><u>:first</u></a>	<a href="#"><u>:last-child</u></a>	<a href="#"><u>:only-of-type</u></a>	<a href="#"><u>:valid</u></a>
<a href="#"><u>:first-child</u></a>	<a href="#"><u>:last-of-type</u></a>	<a href="#"><u>:optional</u></a>	<a href="#"><u>:visited</u></a>
<a href="#"><u>:first-of-type</u></a>	<a href="#"><u>:left</u></a>	<a href="#"><u>:out-of-range</u></a>	



# Pseudo-clases de interfaz

Patrón	Significado
E:link E:visited E:active	Selecciona a un elemento E siendo la fuente de un enlace en el que su objetivo aún no está visitado (:link) o ya ha sido visitado (:visited) o se está presionando (:active)
E:hover E:focus	Selecciona a un elemento E durante cierta acción: hover (mouse encima), focus (ej. un input tiene el control de edición).
E:target	Selecciona al elemento siendo el objetivo de referencia de la URI
E:lang(fr)	Selecciona a un elemento de tipo E con el atributo lang con el valor "fr"
E:enabled E:disabled	Selecciona al elemento E que este habilitado o deshabilitado
E:checked	Selecciona a un elemento de interfaz E que tenga la propiedad checked como radio-button o checkbox.



# Pseudo-clases de estructuras

Patrón	Significado
E:root	Elemento E tal que sea el raíz del documento
E:nth-child(n)	Elemento E tal que sea el n-ésimo hijo de su padre
E:nth-last-child(n)	Elemento E tal que sea el n-ésimo hijo de su padre contando desde el último
E:nth-of-type(n)	Elemento E tal que sea el n-ésimo hermano de su tipo
E:nth-last-of-type(n)	Elemento E tal que sea el n-ésimo hermano del tipo desde el último
E:first-child	Elemento E tal que sea el primer hijo de su padre
E:last-child	Elemento E tal que sea el ultimo hijo de su padre
E:first-of-type	Elemento E tal que sea el primer hermano de su tipo
E:last-of-type	Elemento E tal que sea el último hermano de su tipo
E:only-child	Elemento E tal que sea el único hijo de su padre
E:only-of-type	Elemento E tal que sea el único hermano de su tipo
E:empty	Elemento E tal que no tenga hijos (incluyendo nodos de texto)
E:not(s)	Elemento E tal que no cumpla con el selector simple s



# Pseudo-elementos

Permiten añadir estilos a una parte *específica* del documento.

Ejemplo:

- Para dar estilo a la primera letra, línea, o elemento
- Para insertar contenido antes o después del contenido de un elemento

Se pueden usar con cuatro puntos `::` o solo dos `:`

<u><code>::after</code></u>	Inserta algo después del contenido del elemento seleccionado.
<u><code>::before</code></u>	Inserta algo antes del contenido del elemento seleccionado.
<u><code>::first-letter</code></u>	Selecciona la primer letra de cada elemento seleccionado
<u><code>::first-line</code></u>	Selecciona la primer línea de cada elemento seleccionado
<u><code>::selection</code></u>	Da estilo a lo que el usuario seleccione respecto del selector indicado.



# Ejemplo de todo lo anterior

## Introducción a CSS

### Entender la sintaxis básica de CSS

Texto relacionado con entender la sintaxis

Ejemplo de selectores:

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

### Entendiendo Selectores básicos

Texto relacionado a selectores Lorem ipsum dolor sit amet consectetur adipisicing elit. Saepe quia id laboriosam in quas doloribus et laudantium ducimus nostrum ab? Est cumque unde quod exercitationem minus, in quas consequuntur nobis.

Ejemplo de selectores:

```
p
{
  color: blue;
  text-align: left;
  background-color: red;
}
```

## Resumen

Lorem ipsum dolor sit amet consectetur adipisicing elit. Accusamus, quasi! Cum quos iusto ipsam inventore. Incidunt necessitatibus quod corrupti sunt earum ab itaque, dignissimos laudantium maiores vitae? Numquam, quaerat totam.

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Querat rem harum, mollitia similique pariatur amet, consequuntur sequi at tempore, ea voluptatem cupiditate possimus tempora ipsum ipsa! Cum eveniet earum totam!



# Ejercicio

```
<li class="tag" data-desc="no-no uso-normal re-no-e">center</li>
<li class="tag" data-desc="no-no uso-mucho re-neutro">p</li>
<li class="tag" data-desc="no-no uso-mucho re-si-e+">div</li>
<li class="tag" data-desc="no-no uso-mucho re-si">html</li>
<li class="tag" data-desc="no-si uso-poco re-neutro">article</li>
<li class="tag" data-desc="no-no uso-normal re-si-e+">textarea</li>
<li class="tag" data-desc="no-si uso-poco re-si">video</li>
<li class="tag" data-desc="no-no uso-poco re-no">shadow</li>
```

center

*p*

*div*

*html*

**article**

**textarea**

**video**

**shadow**

Siguiente >>



# Ejercicio

## Tablas

Descripción	Gasto
Libros	400.0
Medicinas	520.0
Alimentos	320.0
Total	1240.0

## Links

Enlace a email  
Ya di click hacia edad

## Inputs

Escribe tu correo:

Escribe tu edad:

Siguiente >>



# Combinadores

- **Grupo de selectores (S1, S2):** En una regla de CSS se pueden usar un grupo de selectores separados por una coma. La regla se aplica a todos los selectores.
- **Descendientes (S1 S2):** Separados por un espacio. Selecciona todos los elementos que son descendientes (anidados) de los elementos seleccionados por el primer selector.
- **Selector hijo (S1 > S2):** Selecciona todos los elementos que son hijos inmediatos de los elementos seleccionados por el primer selector.
- **Selector de hermano adyacente (S1 + S2):** Selecciona todos los elementos que están a un mismo nivel (hermanos) y que son adyacentes (inmediatamente al lado) de los elementos seleccionados por el primer selector.





# Combinadores

- **Selector de hermanos general (S1 ~ S2):** Como el anterior, pero selecciona a todos los hermanos (no tiene que estar al lado/ser adyacente).



# Ejercicio

## Cuantos valores puedes atrapar

Reglas:

- Debes hacer que se vea similar a la imagen de solución
- Cada color representa una regla de CSS
- **No añadir atributos o cambiar contenido al HTML.**

### Primera tabla

-1	2	-1	-1	3	-1	1	4	-1	-1
1	1	1	1	1	1	1	1	1	1
-1	-1	-1	-1	-1	-1	1	-1	-1	-1
-1	-1	-1	-1	-1	-1	1	1	1	1
1	1	1	-1	5	-1	1	-1	-1	1

### Segunda tabla

-1	-1	-1			3	-1	5
1		3	-1	-1	1		
-1	1		3	-1	-1	-1	