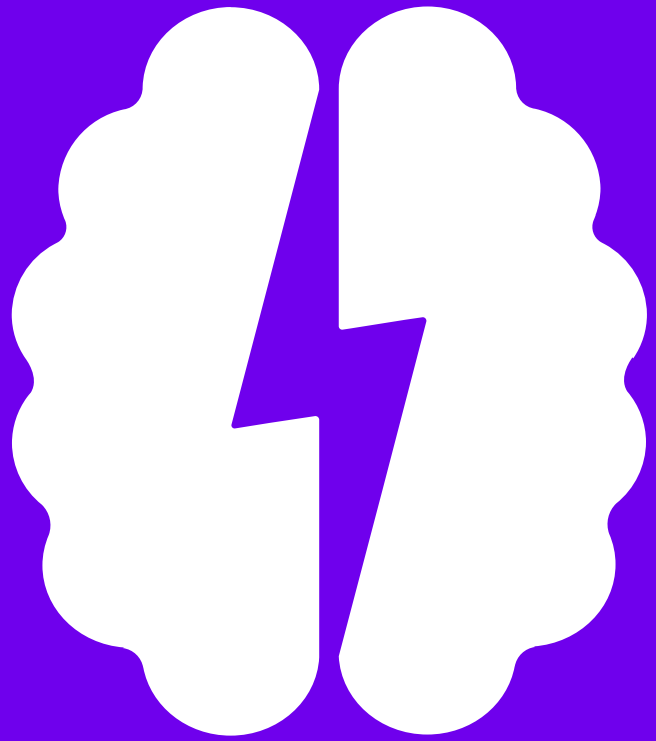
CURSO: DESARROLLO DE APLICACIONES WEB

Sesión 2: Fundamentos de CSS



OBJETIVOS

Al finalizar la sesión, el alumno logrará:

- Comprender y aplicar los conceptos básicos de CSS, incluyendo la creación de archivos externos CSS y la estructura básica de las reglas CSS.
- Utilizar de manera adecuada y eficiente los diferentes tipos de selectores en CSS, como los selectores de elemento, id, clase, pseudoclases, pseudoelementos y atributo.
- Entender los conceptos de especificidad y combinadores en CSS para poder determinar qué regla CSS se aplicará.
- Construir páginas web con interfaces más amigables haciendo uso de propiedades básicas de CSS.

AGENDA

- Introducción a CSS
- Estructura Básica
- Selectores (Elemento, id, clase, Pseudoclases, Pseudoelementos y atributo)
- Especificidad de selectores
- Combinadores de selectores (Simple y Lógicos)
- Crear archivos externos CSS
- Reglas especiales en CSS
- Propiedades CSS más utilizadas (Texto, colores, longitudes, porcentajes y multimedia)
- CSS Box Model
- Recomendaciones de calidad de código

i d a t

inicio desarrollo aplicación término

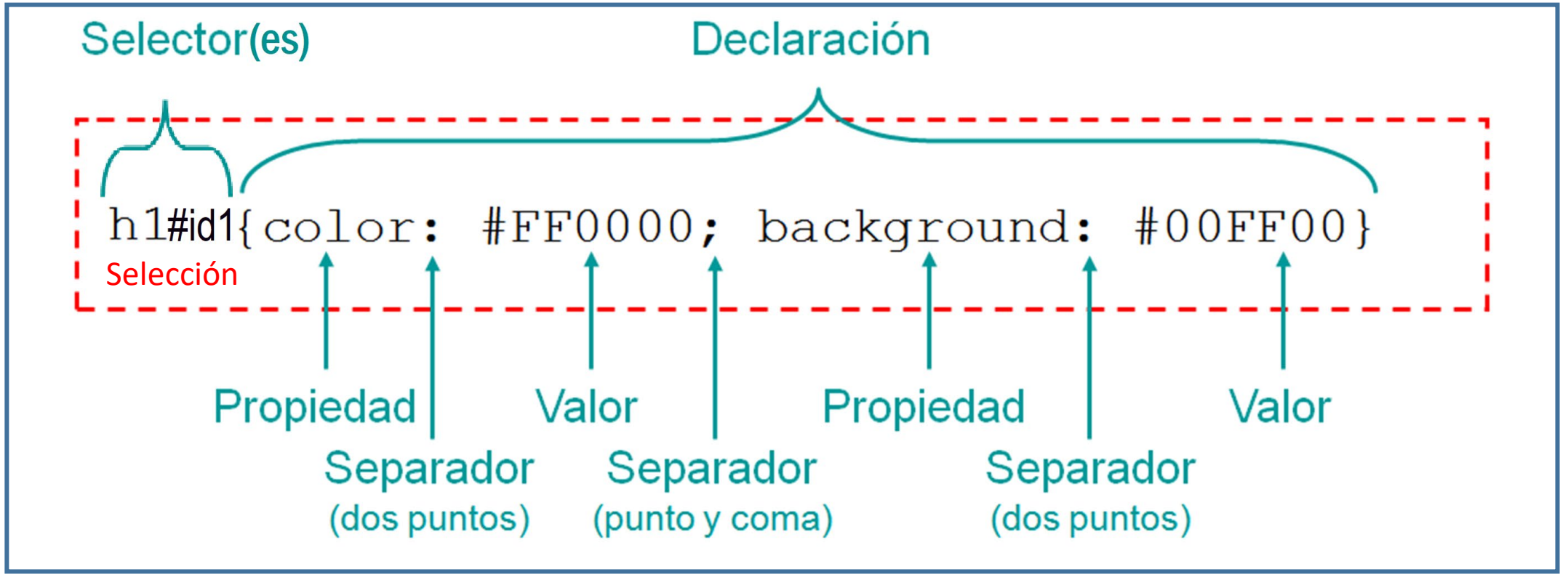
INTRODUCCIÓN A CSS

- CSS (en inglés Cascading Style Sheets) u Hojas de estilo en cascada se usa para estilizar las etiquetas HTML que tengas en tu página web.
- Actualmente está en su versión 3 (CSS3).
- Existe preprocesadores como SASS y LESS que te permiten escribir CSS en un idioma dinámico.
- No es un lenguaje de programación, tampoco es un lenguaje de marcado. Es un lenguaje de hojas de estilo (**Reglas**).
- Permite aplicar estilos de manera selectiva a elementos en documentos HTML .
- Permite separar la parte gráfica (visualización) del contenido.
- Pueden estar en un archivo CSS aparte o dentro del html (En elemento style del head o dentro de la propiedad style de algún elemento).
- Tienen una sintaxis determinada.
- Reglas compuestas por selectores, propiedades y valores.

Ejemplo:

SELECTOR {font-size: 2em}

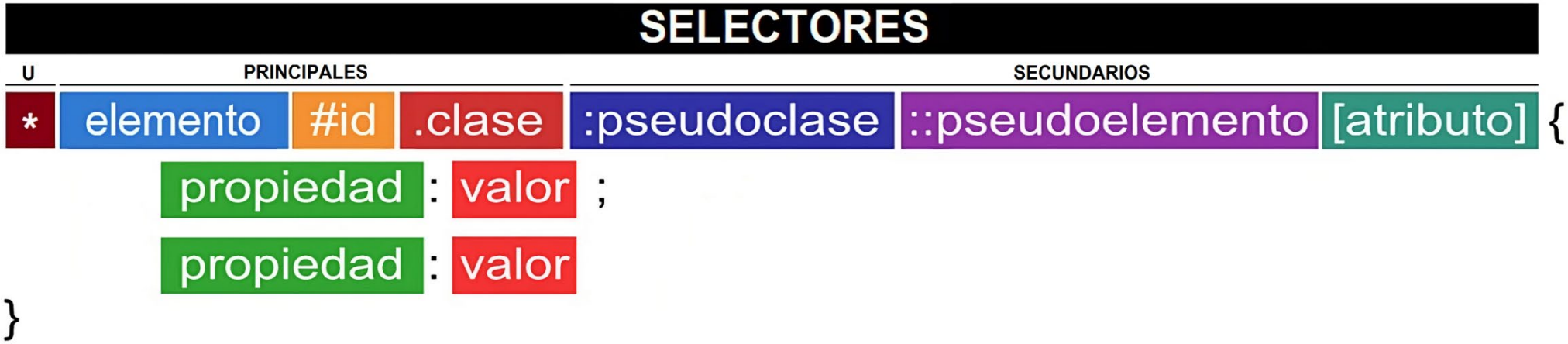
ESTRUCTURA BÁSICA



REGLA CSS

SELECTORES

Un selector CSS es un patrón o criterio utilizado en lenguaje de estilos CSS para seleccionar y aplicar estilos a elementos HTML específicos en una página web. Los selectores CSS permiten identificar y apuntar a elementos HTML. A continuación, mostramos cada uno de los selectores que podemos utilizar:



Por ejemplo:

```
p {
  color: yellow;
  text-align: center;
}
```

```
.parrafo {
  background-color: yellow;
}
```

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

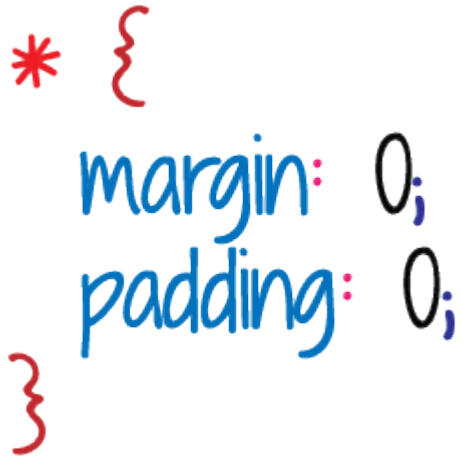
```
div.parrafo#primer{
  color:blue;
}
```

```
<p class="parrafo">Este es un parrafo</p>
```

```
<p>hola</p>
<p class="parrafo"> hola</p>
<div class ="parrafo"> hola </div>
<div id="prime" class ="parrafo" > hola </ div >
```

SELECTOR UNIVERSAL

Este selector se usa empleando el asterisco (*) y hace referencia a todos los elementos. Si deseamos hacer una especie de reset en nuestros estilos podemos emplear este selector.



SELECTORDE ETIQUETA O ELEMENTO(TAG SELECTOR)

Este selector corresponde al elemento HTML y aplica la regla correspondiente a todos los elementos que usen esta etiqueta.

```
p {
    color: yellow;
    text-align: center;
}
```

SELECTORDE ID (ID SELECTOR)

Este selector es usado para colocar una regla a un único elemento, porque un elemento puede tener muchas clases en el atributo class, pero solo puede tener un id, en el atributo id y debe ser único por documento. Se inicia con el numeral (#), seguido del nombre del id, y se especifican los estilos que se desea que este selector tenga.

```
#primer-parrafo {
  background-color: blue;
}
```

SELECTORDE CLASE(CLASS SELECTOR)

Este selector se aplica para los elementos HTML que contengan el atributo clase o class (que es como se escribe en el elemento). Se inicializa con el '.' seguido de su nombre, que contendrá la regla de estilos que deseemos definir. En un elemento HTML podemos tener más de una clase.

```
.yellow {
    color: yellow;
}

.parrafo {
    background-color: yellow;
}
```

SELECTOR DE PSEUDO-CLASES (PSEUDO-CLASSES SELECTOR)

La pseudo-clase permite colocarle estilos a un estado especial de los elementos, como: hover, focus, visited, etc. Inicia con los dos puntos (:), seguido de la pseudo-clase. Estos selectores son considerados como secundarios y se recomienda utilizarlos siempre con un selector principal (En especial un selector de tipo elemento).

```
a:hover {
    color: yellow;
    text-align: center;
}
```

TIPOS DE PSEUDO-CLASES	
TIPO	**FINALIDAD**
Interacción	Permiten aplicar los estilos a elementos según las acciones del usuario.
Ubicación	Permiten aplicar los estilos a enlaces e hipervínculos según el comportamiento del mismo
Idioma	Permiten aplicar los estilos a elementos según el atributo de idiomas.
Estructura	Permiten aplicar los estilos a elementos según estructura del documento HTML.
Formulario	Permiten aplicar los estilos a elementos según las acciones en un formulario HTML.
Estado	Permiten aplicar los estilos a elementos según el estado de un modal o similar.
Paginado	Permiten aplicar los estilos a elementos según la página del documento HTML.

PSEUDO-CLASES DE INTERACCIÓN	
PSEUDOCLASE	**DESCRIPCIÓN**
:hover	Selecciona el elemento si el usuario pasa el ratón sobre dicho elemento.
:active	Selecciona el elemento si el usuario se encuentra pulsando dicho elemento.
:focus	Selecciona el elemento cuando tiene el foco (está en primer plano).
:focus-within	Selecciona el elemento si uno de sus miembros hijos ha ganado el foco.
:focus-visible	Selecciona el elemento cuando tiene el foco sólo de forma visible (TAB, por ejemplo).

MAS PSEUDO-CLASES EN LA SIGUIENTE URL: https://lenguajecss.com/css/selectores/pseudoclases/

SELECTOR DE PSEUDO-ELEMENTOS (PSEUDO-ELEMENTS SELECTOR)

El pseudo-elemento permiten colocarle estilos a un elemento o una parte muy específica de él. Inicia con los dos dobles puntos (::), seguido de la pseudo-elemento. Estos selectores son considerados como secundarios y se recomienda utilizarlos siempre con un selector principal (En especial un selector de tipo elemento).

```
p::first-letter {
    color: yellow;
    font-size: 24px;
}
```

PSEUDOELEMENTO	DESCRIPCIÓN
::before	Aplica los estilos antes del elemento indicado.
::after	Aplica los estilos después del elemento indicado.
::first-letter	Aplica los estilos en la primera letra del texto.
::first-line	Aplica los estilos en la primera línea del texto.
::marker	Aplica estilos a los elementos de cada ítem de una lista.
::backdrop	Aplica estilos al fondo exterior de un elemento en primer plano (sin que afecte a este).

PSEUDOELEMENTO	DESCRIPCIÓN
::selection	Aplica estilos al fragmento de texto seleccionado por el usuario.
::target-text	Aplica estilos al fragmento de texto enlazado tras el ancla de la URL.
::spelling-error	Aplica estilos al fragmento de texto resaltado por error ortográfico.
::grammar-error	Aplica estilos al fragmento de texto resaltado por error gramatical.
::placeholder	Aplica estilos a los textos de sugerencia de los campos <input>.
::file-selector-button	Aplica estilos a los botones de campo <input> de subir archivos.

SELECTOR DE ATRIBUTO(ATRIBUTE SELECTORS)

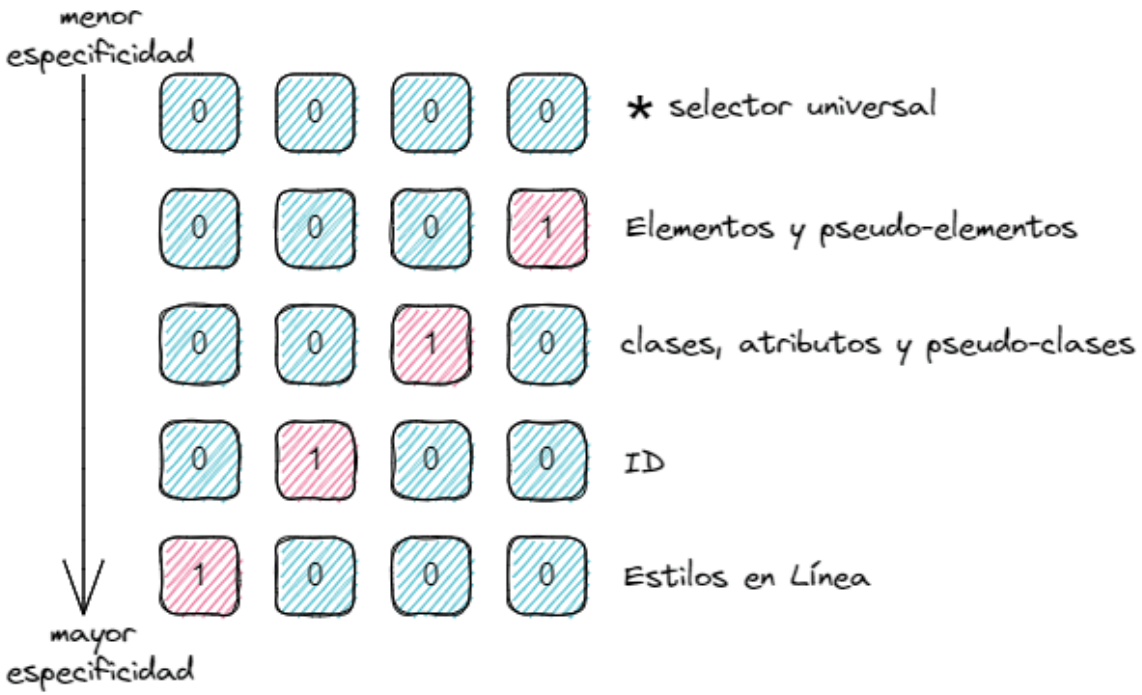
Estos selectores secundarios tienen la particularidad que permiten aplicarle estilos a los elementos que tienen atributos. Por ejemplo, si deseamos aplicarle estilos a una etiqueta <input /> que tienen un atributo type, podemos hacer la regla especifica a ese type.

```
input[type="text"] {
  color: coral;
  backgroun-color: yellow;
}
```

```
input[type="submit"] {
  border: 1px solid coral;
  backgroun-color: light-coral;
}
```

ESPECIFICIDAD DE SELECTORES

La especificidad es un valor numérico que indica cuán preciso es un estilo. Mientras mayor sea ese valor para un estilo, más prioridad tiene de cara al resto de estilos para aplicarse finalmente



!important gana a todos

Ejemplo:

Especificidad : (0,1,2,2)

```
body   #content  .data   img:hover { ... }
                                    border-color: red; !important
```

Elemento (0,0,0,1) + ID (0,1,0,0) + clase (0,0,1,0) + Elemento (0,0,0,1) + pseudo-clase (0,0,1,0) = (0,1,2,2)

```
<img  style="border-color:blue" src ="imagen.jpg">
```

```
.content div.parrafo#primer{
  color:blue;
}
```

(0,0,1,0) + (0,0,0,1)+ (0,0,1,0) + (0,1,0,0) = (0,1,2,1) = 121

COMBINADORES DE SELECTORES (SIMPLES)

Un combinador CSS es un símbolo que permite combinar dos o más selectores CSS, formando uno más complejo y potente.

NOMBRE	SÍMBOLO	EJEMPLO	SIGNIFICADO
Combinador descendiente	(espacio)	#page div { }	Aplica los estilos a los selectores que este dentro de otro (cualquier nivel).
Combinador hijo	>	#page > div { }	Aplica los estilos a los selectores hijos directos (primer nivel).
Combinador hermano adyacente	+	div + div { }	Aplica los estilos a los selectores contiguos a otros (mismo nivel).
Combinador hermano general	~	div ~ div { }	Aplica los estilos a los selectores que siguen a otros (mismo nivel).
Combinador universal	*	#page * { }	Aplica los estilos a todos los selectores dentro del principal (cualquier nivel).

```
<div id="page">
  <div> hola </div>
  <div>
    <h1>hola </h1>
    <div> hola </div>
   <div> hola </div>
  </div>
</div>
<div> hola </div>
```

COMBINADORES DE SELECTORES (LÓGICOS)

Estos combinadores lógicos nos permiten seleccionar elementos con ciertas restricciones y funcionan como una pseudoclase, sólo que se le pueden pasar parámetros, ya que son de tipo pseudoclase funcional.

NOMBRE	SÍMBOLO	EJEMPLO	SIGNIFICADO
Combinador por agrupación	,	p, div { }	Aplica los estilos a varios selectores separándolos por comas.
Combinador :is()	:is()	p:is(div, h1){ }	Idéntico al anterior, pero permite combinar con otros selectores.
Combinador :where()	:where()	p:where(div){ }	Idéntico al anterior, pero con menor especificidad CSS.
Combinador :has()	:has()	p:has(div) { }	Aplica los estilos a los selectores padres que tengan ciertas características en sus hijos.
Combinador :not()	:not()	p:not(div) { }	Aplica los estilos a los selectores que no cumplan ciertas características.

```
<div id="page">
  <div> hola </div>
  <p>
    <h1>hola</h1>
    <div> hola</div>
    <p> hola</p>
  </p>
</div>
<div> hola </div>
```

CREAR ARCHIVOS EXTERNOS CSS

ARCHIVO EXTERNO (*.CSS)

Crear un archivo *.css y referenciarlo desde el HTML

enuciado1.css

```
p { color : green; border: solid red; }
```

enunciado1.html

```
<HEAD>
   <LINK href=" enuciado1.css" rel="stylesheet" type="text/css">
</HEAD>
<BODY>
    <P>Este párrafo debería tener texto especial verde.</P>
 </BODY>
```

Este párrafo debería tener texto especial verde.

Laboratorio Nº 1: Crear una página básica con CSS

- Crear una Página Web con la estructura HTML básica y aplica los distintos selectores CSS presentados en la sesión .
- En este laboratorio, el alumno diseñará su primera página web básica con CSS.

REGLAS ESPECIALES EN CSS

En CSS tenemos las denominadas «at-rules» (reglas precedidas del carácter @). Son un tipo de declaración especial que permite indicar comportamientos especiales en muchos contextos. Su sintaxis suele determinarse incluyendo una palabra clave que comienza por @ (como por ejemplo @media o @import) y dependiendo de la regla en cuestión

Ejemplo:

```css
/* Regla @import para importar estilos CSS de otro fichero */
@import url("index.css");

/* Media query para definir estilos de móvil */
@media screen and (max-width: 640px) {
  .page {
    width: 100%;
  }
}
```

REGLA	DESCRIPCIÓN
@import	Incluye una hoja de estilos externa, indicando su URL.
@charset	Indicaba la codificación de carácteres usada en el fichero .css actual.
@supports	Establece un bloque de estilos que se aplicará si se cumple la condición.
@media	Aplica los estilos si cumple las condiciones indicadas. Usado para responsive.
@keyframes	Crea los fotogramas clave de una animación.
@scope	Aplica estilos con un límite de ámbito en elementos descendientes.
@font-face	Indica una tipografía externa que el navegador debe descargar.
@font-feature-values	Activa/desactiva características especiales en una tipografía
@when / @else	Establece unos estilos si se cumple la condición y otros si no se cumple.
@page	Modifica propiedades relacionadas con una página impresa.
@counter-style	Indica el estilo que se utilizará en los contadores CSS.
@layer	Establece que estilos se incluirán en una capa (que se fusionará más tarde).

PROPIEDADES CSS MÁS UTILIZADAS

Fuentes	Texto
font-family: tipos *(arial,verdana,...).* *Se puede especificar una lista de opciones* **font-weight**: grosor *(negrita,...)* **font-size**: tamaño **font-style**: estilo *(itálica,..)*	**color**: color **text-decoration**: efectos *(subrayado,...)* **text-align**: alineación **line-height**: interlineado **text-indent**: tabulación **letter-spacing**:espaciado entre letras
Listas	**Fondo**
list-style-type: tipo de viñeta *(círculo, cuadrado, número, letra...)* **list-style-image**: utilizar imágenes en las viñetas **list-style-position**: alineación del texto con la viñeta	**background-color:** color **background-image**: poner una imagen **background-repeat**: cómo se repite la imagen de fondo *(vertical, horizontal,..)* **background-position**: posición de la imagen de fondo

Anulación	
Comentario regla	*/* selector { propiedad: valor; } */*
Comentar propiedad-valor	selector { */* propiedad: valor; */ }

Posición y flujo	
display	*block, inline, inline-block, list-item, compact, run-in, marker, none*
float	*left, right, none*
clear	*left, right, both, none*
position	*static, relative, absolute, fixed*
visibility	*visible, hidden, collapse*
overflow	*visible, hidden, scroll, auto*
overflow	*visible, hidden, scroll, auto*
z-index	*auto, inherit or (1,2,-1,-2...)*

PROPIEDADES CSS MÁS UTILIZADAS

Contenido	**Borde**
width: anchura **height**: altura **line-height**: distancia entre líneas **overflow**: qué presentar cuando el contenido excede el espacio asignado (*scroll*, ocultar, mostrar...)	**border-width**: anchura **border-style**: tipo *(linea, puntos, guiones...)* **border-color:** color **border-radius:** redondeo de las esquinas
Padding y Márgenes	
padding: anchura de la zona de relleno **margin**: anchura de los márgenes *Pueden aplicarse valores distintos a cada lado:* *arriba (top), derecha (right), abajo (bottom), izquierda (left)* *Ejemplos: margin-top* *padding-right* *border-width-bottom*	

PROPIEDADES CSS MÁS UTILIZADAS

valores absolutos

Pixels (**px**)
(realmente es un valor relativo ya que el tamaño del pixel depende de la resolución de la pantalla)

Milímetros (**mm**) y **centímetros** (**cm**)

Puntos (**pt**) y **picas** (**pc**)
(1 pt = 0,35 mm ; 1 pc = 12 pt)

Pulgadas (**in**)
(1 pulgada = 72 pt)

valores relativos

Porcentajes (**%**)

Factor de escala (**em**) y (**ex**)
1 em: el tamaño de la fuente usada
1 ex: 0,5em (no es exacto, depende de la fuente)

Los porcentajes y escalas se calculan respecto al tamaño de fuente del elemento padre (contenedor).

Ej: si en la sección <body> hay 16px, un elemento incluido en ella (p.ej. H1) tendrá este valor:

h1: 150% → 24px [16px + 50% (8px)]
h1: 1.5em → 24px
h1: 1.5ex → 12px (puede variar según fuente)

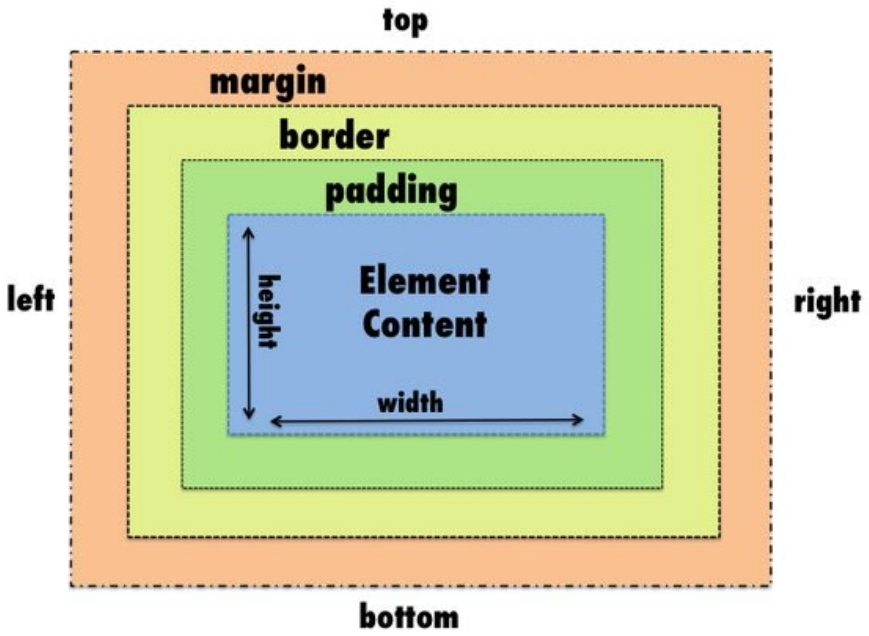
Palabras clave

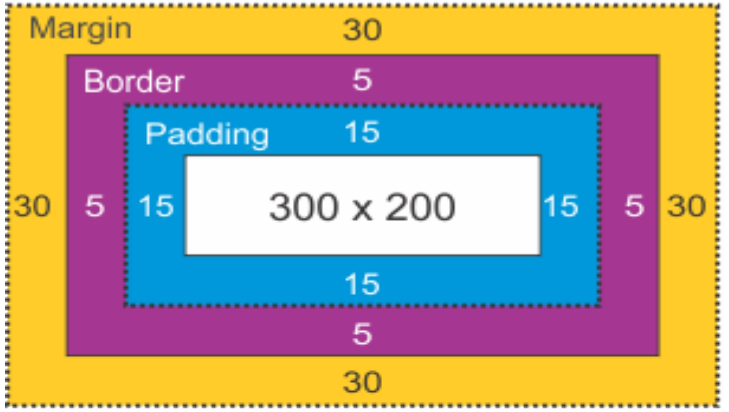
De menor a mayor tamaño *(valores absolutos, similares a tamaños por defecto de títulos hx)*
xx-small, x-small, small, medium, large, x-large, xx-large

Tamaño relativos a la fuente del contenedor:
Más pequeño: smaller
Más grande: larger

CSS BOX MODEL

Cada caja se compone de cuatro partes, definidas por sus respectivos limites: el límite del contenido(Content), el límite del relleno(Padding), el límite del borde(Border) y el límite del margen(Margin)





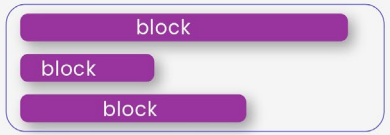

```
div{
        width: 300px;
        height: 200px;
        padding: 15px;
        border: 5px solid grey;
        margin: 30px;
        -moz-box-sizing: content-box;
        -webkit-box-sizing: content-box;
        box-sizing: content-box;
    }
```

CSS BOX MODEL

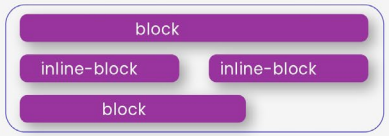
Se puede utilizar las propiedades DISPLAY y POSITION para controlar la ubicación de un elemento

Display


```
.container {
  display: inline;
}
```



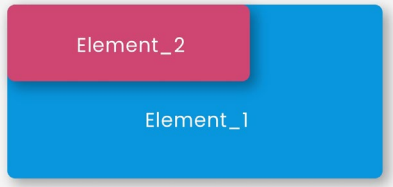

```
.container {
  display: block;
}
```




```
.container {
  display: inline-block;
}
```

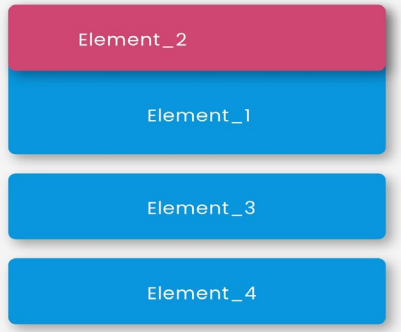
Position


```
.element_1 {
  position: relative;
}
.element_2 {
  position: relative;
}
```




```
.element_1 {
  position: relative;
}
.element_2 {
  position: absolute;
}
```


```
.element_1 {
  position: relative;
}
.element_2 {
  position: fixed;
}
```




```
.element_1 {
  position: relative;
}
.element_2 {
  position: stciky;
}
```

RECOMENDACIONES DE CALIDAD DE CÓDIGO

- Comenta Tu Código
- Utiliza Hojas de Estilo Separadas para los Proyectos más Grandes
- Minimiza tu Hoja de Estilos
- Utiliza un Preprocesador
- Considera un Framework CSS
- Evita la Redundancia
- Haz que el CSS sea Accesible
- Evita la Etiqueta !Important

Laboratorio Nº 2: Aplicar CSS a un formulario HTML

- Aplicar CSS al formulario de la página web desarrollada en la sesión anterior.
- En este laboratorio, el alumno aplicara todos los conocimientos de CSS y HTML desarrollados en clase.

LINKS IMPORTANTES

- https://fjph32html.wordpress.com/2015/04/14/teoria-parte-2-selectores-en-css3-basicos/
- https://platzi.com/clases/2467-frontend-developer/40836-tipos-de-selectores-pseudoclases-y-pseudoelementos/
- https://lenguajecss.com/css/
- https://www.mclibre.org/consultar/htmlcss/css/css-pseudoclases.html
- https://developer.mozilla.org/es/docs/Learn/CSS/Building_blocks/Selectors/Pseudo-classes_and_pseudo-elements
- https://www.um.es/docencia/barzana/DAWEB/2017-18/daweb-tema-11-selectores-en-css.html
- https://lenguajecss.com/css/selectores/selectores-basicos/
- https://vanessamarely.medium.com/selectores-en-css-2996ada199d9
- https://lenguajecss.com/css/
- https://tech.tribalyte.eu/blog-entendiendo-css-parte-1
- https://dev.to/lupitacode/especificidad-en-css-que-es-y-como-funciona-52k6
- https://mycoderbrain.wordpress.com/2018/04/24/selectores-css/
- https://medium.com/laboratoria-how-to/introducci%C3%B3n-a-css-computing-specificity-parte-ii-46982c55b3b3
- https://keepcoding.io/blog/pseudo-clases-pseudo-elementos-en-css/
- https://htmlcheatsheet.com/css/
- http://cheatsheets.shecodes.io/css
- https://acchou.github.io/html-css-cheat-sheet/html-css-cheat-sheet.html#css
- https://devhints.io/csshttps://www.digitallearning.es/curso-wordpress/css-lenguaje-de-estilos.html
- https://specificity.keegan.st/

¿Qué te llevas de la sesión?

idat

PREGUNTAS

!GRACIASi

REFERENCIAS:

- Introducción CSS (Raúl Rodríguez)
- Introducción a CSS (Maria Zubieta)
- El lenguaje de estilos CSS (Digital Learning)
- Entendiendo CSS (Maribel Diaz)
- Selectores en CSS (Vanessa Aristizabal)
- CSS: selectores básicos (Fernando López)
- Lenguaje CSS (Manz.dev)
- Especificidad en CSS (Hameyalli Elizalde)
- Ayuda rápida css (EBWeb)
- Cascading Style Sheets (Veign)
- CSS3 Cheatsheet (Emezeta)
- Box Model (Jhonnatan Flores)
- Modelo de cajas y posicionamiento CSS (Diego Martin)
- CSS Position (DevTown)