

Display

Permite definir el comportamiento de una caja con respecto a sus adyacentes, recordando que existen dos tipos de elementos en HTML, inline y block, los valores de la propiedad son:

- **inline:** hace que una caja que se comporte como si fuera un elemento en línea.
- **Block:** con este valor, la caja se comportaría como un elemento en bloque.
- **inline-block:** permite añadir a un elemento inline las propiedades width y height
- **none:** oculta el elemento pero todo lo que utilice el elemento se sigue renderizando
- **table:** imita el comportamiento de una tabla
- **list-item:** imita el comportamiento de una lista
- **flex:** Maquetar componentes como un menú
- **grid:** sistema de grilla

PseudoElementos

Se utilizan para dar estilo a partes específicas de un elemento, no son componentes reales, son una especie de componentes virtuales que solo existen en los estilos del elemento. La sintaxis de un pseudo elemento es

`Selector::pseudo-elemento{ estilos }`

Existen 5 pseudoelementos básicamente:

- **i::first-line** -> Solo se aplica para elementos en bloque y lo que hace es añadir un comportamiento o estilo a la primera línea del elemento que se indique.
- **::first-letter** -> Funciona igual que first-line solo que aquí se aplica para únicamente la primera letra.
- **::before** -> añade estilos o contenido justo antes del componente indicado, es obligatorio el uso de la propiedad content, puede ser inicializada en vacío.
- **::after** -> añade estilos o contenido justo después del componente indicado, es obligatorio el uso de la propiedad content, puede ser inicializada en vacío.
- **::selection** -> añade estilo al momento de seleccionar una parte o todo el elemento.

PseudoClases

Son selectores que reaccionan en tiempo real, principalmente cuando un usuario interactúa con alguno de los elementos, una característica es que funcionan con cualquier tipo de selector e incluso sin ninguno.

```
selector:pseudoclase{
  estilos
}

::pseudoclase{
  estilos
}
```

Índice de las pseudo-classes estándar

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

Fuente: MDN web docs, Pseudo-classes,
<https://developer.mozilla.org/es/docs/Web/CSS/Pseudo-classes>

Según tipos se pueden dividir en:

Dinámicas:

- **link:** un enlace no visitado
- **visited:** un enlace visitado
- **active:** al momento de dar clic sobre el elemento
- **hover:** cuando se posiciona el mouse sobre el elemento
- **focus:** cuando un elemento tiene el foco o está seleccionado, por ejemplo un input

Objetivo:

- **target:** se aplica cuando un elemento es un marcador o ancla.

Lenguaje:

- **lang():** Elemento que contiene el identificador de idioma que se le indica, es una función.

UI de estado:

- **Enable:** campo de formulario que estén activados, enable es el valor por defecto de todo input
- **Disable:** se aplica a los elementos que estén desactivados.
- **Checked:** se aplica cuando un elemento es marcado o está checked.
- **in-range:** Se aplica cuando el elemento se encuentra dentro del rango
- **out-of-range:** se aplica cuando el valor en el elemento se encuentra fuera del rango
- **optional:** Por defecto todos los campos son opciones, se pueden aplicar efectos a dichos campos.
- **required:** Aplicar un estilo a los elementos que se marcan como obligatorios.
- **valid:** aplicar a un elemento cuando el valor del campo es válido
- **Invalid:** Aplicar estilo cuando el valor del elemento no es válido.

Estructurales:

- **root:** representa la raíz del documento, se le da estilos al HTML
- **empty:** estilos a los elemento que no tengan contenido
child: estilos sobre los hijos.

- **first-child:** primer hijo
- **last-child:** último hijo
- **nth-child():** selecciona al n-ésimo hijo
- **nth-last-child():** Selecciona el n-ésimo hijo pero iniciando desde el último

only-child: funciona solo cuando hay un único hijo.

odd: impares

even: pares

- números enteros
- ecuaciones

Type:

- Al igual que con los child para un tipo de etiqueta o una clase definida.

Negación:

- **not():** se aplica a todos los elementos excepto lo que se indique en el not.

Variables CSS

Una variable es un espacio en memoria en donde se almacena un color para poder utilizarlo o modificarlo.

Las variables en CSS necesitan estar dentro de un selector, tener en cuenta el :root.

Tienen herencia y cascada

Existen variables globales y locales

```

/* Para declararla */
:root{
  --nombre-variable: valor;
}

/* Para utilizarla */
.selector{
  propiedad: var(--nombre-variable)
}

```

Background

Explicación background-color, image, attachment (efecto parallax), clip, origin, repeat, position, origin, size multiple.

Textos y tipografías

- **Son los tipos de letra que existen, se dividen en dos grupos.**
- Familias tipográficas, tienen nombres propios como Arial, Times New Roman, Verdana, Calibri.
- Familias genéricas o seguras, son grupos asociadas por sus características, Serif, sans serif, cursive...

Se recomienda usar fuentes de respaldo

- **Los modificadores de font son:**
 - family
 - size
 - weight
 - style
- **Los modificadores de bloques de text son (aplicar al contenedor del texto):**
 - transform
 - align
 - decoration
 - indent
 - line-height (truco de centrado)
 - letter-spacing