

Curso CSS desde Cero

CSS

CSS





CASCADE

STYLE

SHEET

Hoja de Estilo en Cascada: Se irá leyendo de arriba a abajo. La última propiedad siempre gana.

Agregando CSS

En línea:

```
<elemento style= “estilos”> Texto </elemento>
```

```
<style>
```

Estilos

```
</style>
```

```
<link rel="stylesheet" href="ruta del archivo">
```

Sintaxis

```
/*Bloque*/
```

```
selector {
```

```
    propiedad-1: valor;
```

```
    propiedad-2: valor;
```

```
}
```

Selectores

Selector de :

- Etiqueta (tipo) (HTML)
- Clases
- ID
- Universal (*)
- Descendiente (padre [espacio] hijo { ... })
- De atributo

<http://apps.workflower.fi/vocabs/css/en>

Variables CSS

Es un espacio de memoria donde almacena un valor.

```
--nombre-variable : valor ;
```

```
--color-primary : red;
```

```
--ancho-maximo: 720px;
```

Utilizando la variable:

```
background-color: var( --color-primary );
```

Estilo para textos

- Color: *Da color a la fuente*
- Text-align: *Alineación horizontal*
- Text-decoration: Especifica la decoración agregada al texto
- Text-transform: Controla mayúsculas y minúsculas del texto
- Letter-spacing: *Incrementa o decrementa el espacio entre letras*
- Line-height: Espacio del interlineado
- Word-spacing: Separación entre palabras
- Text-shadow: Sombreado al texto

Texto

- Color: `red` | `#ff22AC` | `rgb(255,100,0)` | `hsl(240, 100%, 25%)`;
 - https://www.w3schools.com/colors/colors_names.asp
- Text-align : `center` | `left` | `right` | `justify`;
- Text-decoration : `none` | `overline` | `line-through` | `underline` ;
- Text-transform : `uppercase` | `lowercase` | `capitalize`;
- Letter-spacing : `(+/-) px` | `em` | `rem`;
- Word-spacing : `(+/-) px` | `em` | `rem`;
- Line-height : `px` | `em` | `rem` | `unidad`;
- Overflow-wrap : `normal` | `break-word` ; `->`
- white -space : `nowrap` | `wrap` | `pre` | `normal` ;
 - https://www.w3schools.com/cssref/pr_text_white-space.asp
- Text-overflow : `clip` | `ellipsis` | `initial` | `inherit` ; `->`

Unidades de medidas

Absolutas

Pulgadas (in)

Centímetros (cm)

Milímetros (mm)

Pixels (px)

Puntos (pt)

Unidades de medidas

Relativas

Contenedor (%)

Tipografía

- em (Contenedor)
- rem (Root -> html)

Viewport

- Ancho ventana(vw)
- Alto ventana (vh)

Fonts

- Font-family : *Define la fuente tipográfica que será aplicada al texto.*
- Font-size : *Define el tamaño de la fuente*
- Font-style : *Formato de la fuente*
- Font-weight : *Le da el peso a la fuente (que tan bold será)*

Fonts

- Font-family : **web fonts**;
- Font-size : **px | em | rem**;
- Font-style : **normal | italic** ;
- Font-weight : **normal | bold | lighter | bolder | 100-900** ;
- **@font-face**

Recursos:

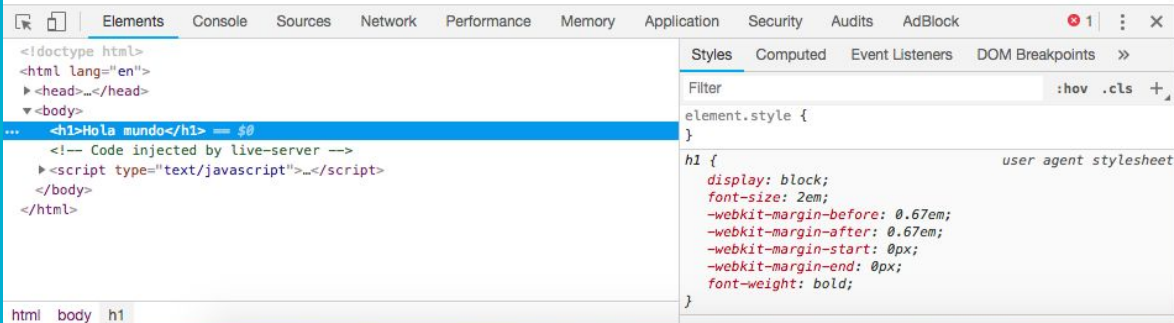
- **<https://fonts.google.com>**

Estilos por defecto

El navegador contiene sus propios estilos por defecto en todos los elementos del HTML.

[Normalize](#)

Hola mundo



Selectores

Selector de Tipo

Selector de tipo (etiqueta HTML), NO los uses a menos que sea el estilo base o de tema.

```
body{
```

```
}
```

```
h1{
```

```
}
```

```
p{
```

```
}
```


Selector de ID

NO los uses.

`<h1 id= "title"> Titulo del sitio </h1>`

`#title { /* referencia al pdf */`

`color: red;`

`}`

Selector de Clase

Siempre debes usar clases para tus estilos.

```
<h1 class = "title"> Titulo del sitio </h1>
```

```
.title{
```

```
    text-align: center;
```

```
}
```

Especificidad (Specificity)

Inline = 1000

ID = 100

Class = 10

Typo = 1

!important /*Nunca lo uses a menos que sea de vida o muerte*/ Su valor es infinito

https://www.w3schools.com/css/css_specificity.asp

<https://developer.mozilla.org/en-US/docs/Web/CSS/Specificity>

Selectores compuestos

Están formados por más de una palabra.

- Descendientes (*Espacio entre selectores*)

```
ancestro descendiente { /* Estilos */  
  
}
```

- Hijo directo (>)

```
padre > hijo { /* Estilos */  
  
}
```

Selectores compuestos

- Hermano siguiente (adyacente) (+)

```
elemento + hermano-siguiente { /* Estilos */  
  
}
```

- Hermanos siguientes (~) (Todos los siguientes)

```
elemento ~ hermanos-siguientes { /* Estilos */  
  
}
```

Selectores compuestos

- Selector compuesto (sin espacios) [tenga ambos selectores]

```
selectorselector { /* Estilos */  
  
}
```

- Selectores agrupados (,) [separados por comas]

```
selector, selector, ... { /* Estilos */  
  
}
```

Selectores de atributo

Sintaxis:

```
[attribute]{ /* Estilos */
```

```
}
```

```
[href]{
```

```
    text-decoration: none;
```

```
}
```

[https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction to CSS/Attribute selectors](https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction%20to%20CSS/Attribute%20selectors)

[https://developer.mozilla.org/en-US/docs/Web/CSS/Attribute selectors](https://developer.mozilla.org/en-US/docs/Web/CSS/Attribute_selectors)

Selectores de atributo

- Comienza con (^)
 - `[title^="palabra"]{ ... }`
- Termina con (\$)
 - `[src$=".jpg"]{ ... }`
- Contiene (*)
 - `[href*=".pdf"]{ ... }`
- Comienza específicamente con (|)
 - `[class="menu"]{ ... }`
- Contiene específicamente la palabra (~)
 - `[id~="header"]{ ... }`
- Ignora mayúsculas y minúsculas (i o I) `[src="imagen" i]{ ... }`

Pseudo clases

Pseudo-classes (:)

Son selectores dinámicos, responden a ciertas eventos que suceden con el elemento.

Sintaxis:

```
selector:pseudo-class { /* Estilos */  
  
}
```

```
:pseudo-class{ ... } /*se aplica a cualquier elemento*/
```

[https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction to CSS/Pseudo-classes and pseudo-elements](https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction%20to%20CSS/Pseudo-classes%20and%20pseudo-elements)

https://www.w3schools.com/css/css_pseudo_classes.asp

Pseudo-clases (:)

:hover -> Se activa al pasar el puntero sobre el elemento

:active -> Cuando se da click sobre el elemento

:link -> Link sin visitar aún

:visited -> Cuando ya ha sido visitado el enlace

:target -> Cuando mandan a llamar al elemento

:not(selector) -> Selecciona los elementos que NO coinciden con el selector.

- https://www.w3schools.com/cssref/sel_not.asp
- <https://developer.mozilla.org/en-US/docs/Web/CSS/:not>

Pseudo-clases (:)

:empty -> Te indica un elemento que está vacío, es decir, que no tiene hijos o texto

:checked -> Si se activo la propiedad checked (radio, checkbox, select)

:focus -> Se activa cuando tiene el foco el input

:enabled -> Se ejecuta cuando el input está disponible

:disabled -> Se ejecuta cuando el input está deshabilitado

:required -> Se activa cuando el input es requerido, pero no opcional

:optional -> Cualquier input, select o textarea que no tenga required

Pseudo-clases (child)

:first-child -> *Encuentra el primer hijo*

:last-child -> *Encuentra al último hijo*

:nth-child(n) -> *Encuentra al hijo “n” [odd, even, X,...]*

:nth-last-child(n) -> *Es igual a nth-child, solo que la cuenta comienza del final al principio*

:only-child -> *Selecciona cuando es un solo hijo, si son más no lo aplica*

Pseudo-clases (child)

:first-of-type -> *Encuentra el primer tipo de elemento*

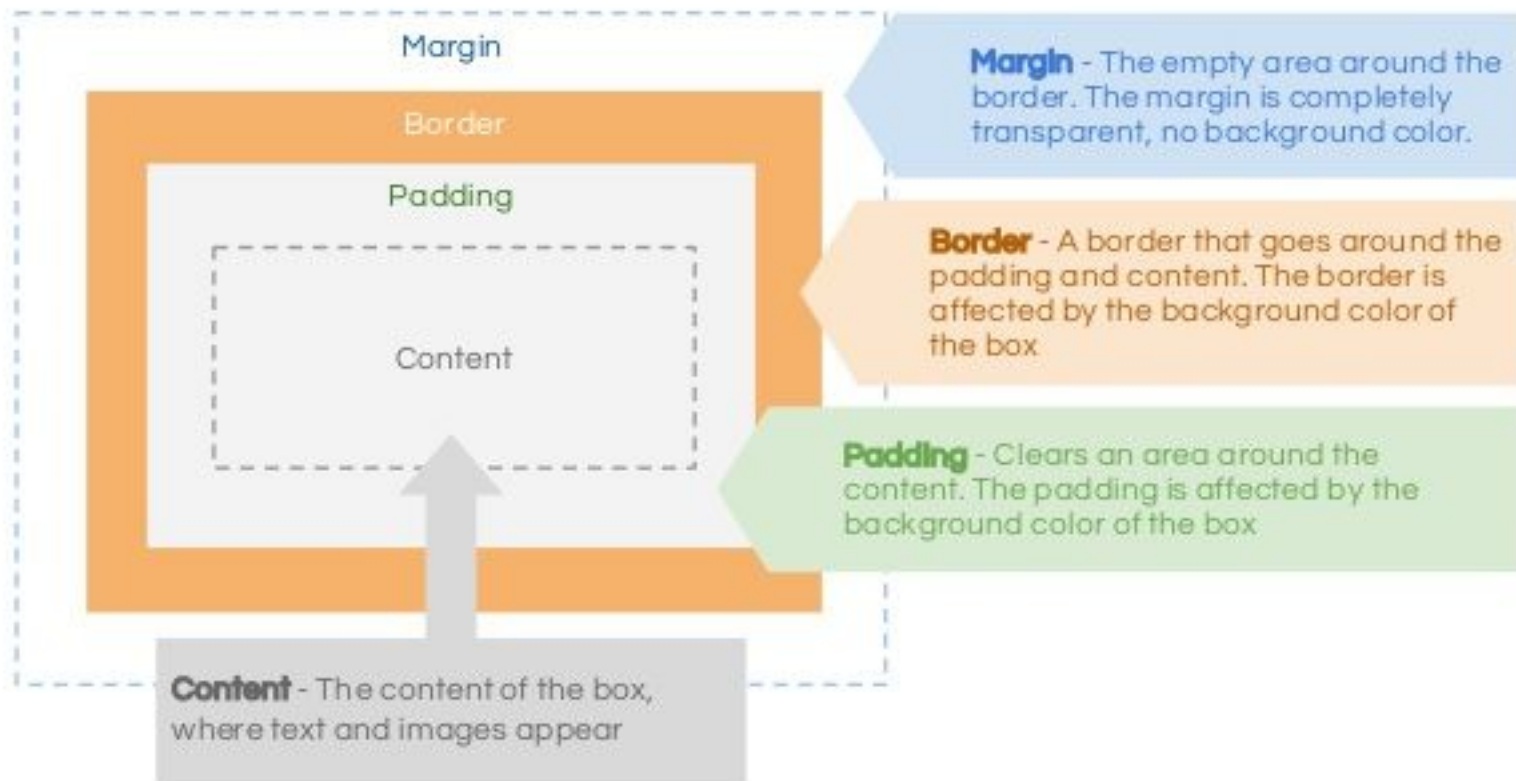
:last-of-type -> *Encuentra al último tipo de elemento*

:nth-of-type(n) -> *Encuentra al tipo de elemento “n” [odd, even, X,...]*

:nth-last-of-type(n) -> *Es igual a nth-of-type, solo que la cuenta comienza del final al principio*

Modelo de Caja

CSS Box Model



Elements

Block vs Inline

Margin

Margin-top: **px** | **em** | **rem** | **%** | **auto**;

Margin-bottom: **px** | **em** | **rem** | **%** | **auto**;

Margin-left: **px** | **em** | **rem** | **%** | **auto**;

Margin-right: **px** | **em** | **rem** | **%** | **auto**;

Margin: margin-top margin-right margin-bottom margin-left;

Margin: (margin-top & margin-bottom) (margin-right & margin-left);

Margin: (margin-top & margin-right & margin-bottom & margin-left); /*todos*/

Colapsado de margenes

Padding

Padding-top: px | em | rem | %;

Padding-right: px | em | rem | %;

Padding-bottom: px | em | rem | %;

Padding-left: px | em | rem | %;

Padding: padding-top padding-right padding-bottom padding-left;

Padding: (*padding-top* & *padding-bottom*) (*padding-right* & *padding-left*);

Padding: (*padding-top* & *padding-right* & *padding-bottom* & *padding-left*);

Border-radius

Ancho y Alto

Width: px | em | rem | % | view | auto;

Height: px | em | rem | % | view | auto;

Max-width: px | em | rem | % | view | auto;

Min-width: px | em | rem | % | view | auto;

Max-height: px | em | rem | % | view | auto;

Min-height: px | em | rem | % | view | auto;

Borde

Border-width: px | em | rem | thin | medium | thick;

Border-style: none | hidden | dotted | dashed | solid | double | groove | ridge | inset | outset;

Border-color: <rgb()> | <rgba()> | <hsl()> | <hsla()> | <hex-color> | <named-color>;

Border: <br-width> || <br-style> || <color>; /**shorthand/

Box-sizing

Background

Color

Imagen

Tamaño

Repetir

Posición

Origen

Shorthand

Pseudo-elementos

::before / ::after

Interacción

Colores

Colores

- RGB, RGBA -> Valores 0 - 255
- HSL, HSLA
- Keyword
- Hex (RGB) -> #000000 - #ffffff

Opacidad

Filter

Overflow

Float

Rompe el flujo. Deja de ser contenido por su padre.

Float: `left` | `right` | `none` ;

<https://developer.mozilla.org/en-US/docs/Web/CSS/float>

https://www.w3schools.com/css/css_float.asp

Con overflow envuelve a sus hijos flotados.

POSITION

Relative | Absolute | Fixed | Sticky

Top - Left - Right - Bottom

Context

Z-index

Text Shadow

<https://developer.mozilla.org/en-US/docs/Web/CSS/text-shadow>
https://www.w3schools.com/cssref/css3_pr_text-shadow.asp

FLEXBOX

Flexbox

Necesita de una caja contenedora con sus respectivos hijos.

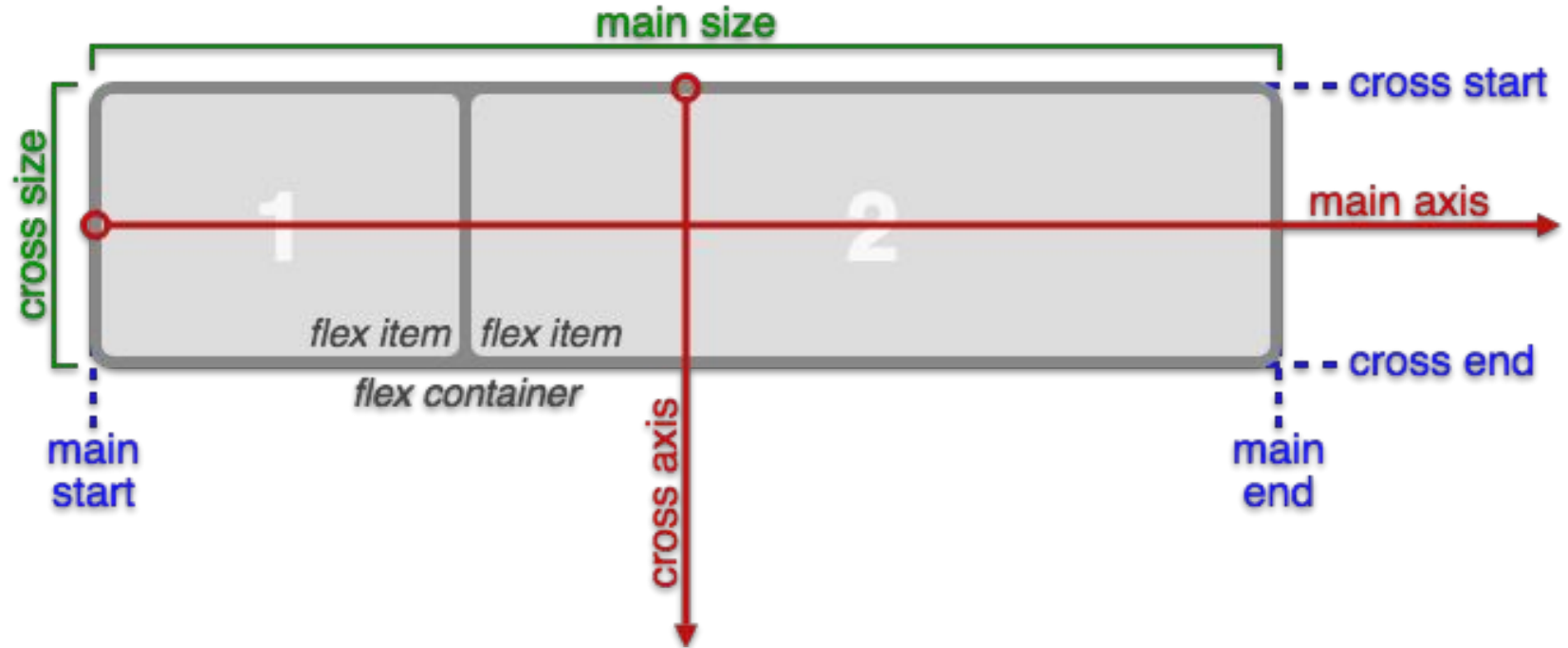
Si no tiene hijos; es decir, otra caja pero tiene texto, el texto es su hijo.

https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Flexbox

<https://www.w3.org/TR/css-flexbox-1/>

display: flex | inline-flex

Flexbox



Flexbox – Container

Flex-direction: **row** | **row-reverse** | **column** | **column-reverse**

flex-wrap: **nowrap** | **wrap** | **wrap-reverse**

flex-flow: **flex-direction** || **flex-wrap** -> shorthand

Flexbox – Container (*Alineamiento*)

Main Axis:

Justify-content: flex-start | flex-end | center | space-between | space-around | space-evenly;

Cross Axis:

Align-content: flex-start | flex-end | center | space-between | space-around | space-evenly ; // (wrap)

Align-items: flex-start | flex-end | center | baseline | stretch; // (no-wrap)

Flexbox – Item (flex)

Flex: **none** | **flex-grow** || **flex-shrink** || **flex-basis** ; -> shorthand

Flex-grow

Flex-shrink

Flex-basis

- Los hijos no crecen por default, pero si se encogen para entrar. Siempre tratan de ponerse en una sola línea.
- Flex-basis gana a width.



Flexbox – Item (*Alineamiento*)

Align-self: auto | flex-start | flex-end | center | baseline | stretch;

Order: <número entero>

Responsive Design

Patrones de Responsive

Media Querys

https://www.w3schools.com/css/css_rwd_mediaqueries.asp

https://www.w3schools.com/Css/css3_mediaqueries_ex.asp

https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries/Using_media_queries

Frameworks

Frameworks

- [Bootstrap](#)
- [Foundation](#)
- [Material Design Lite](#)
- [Pure.CSS](#)
- [Materialize](#)
- [Bulma](#)

Preprocesadores

[SASS](#)

[LESS](#)

[Stylus](#)

[PostCSS](#)