

5

CS5

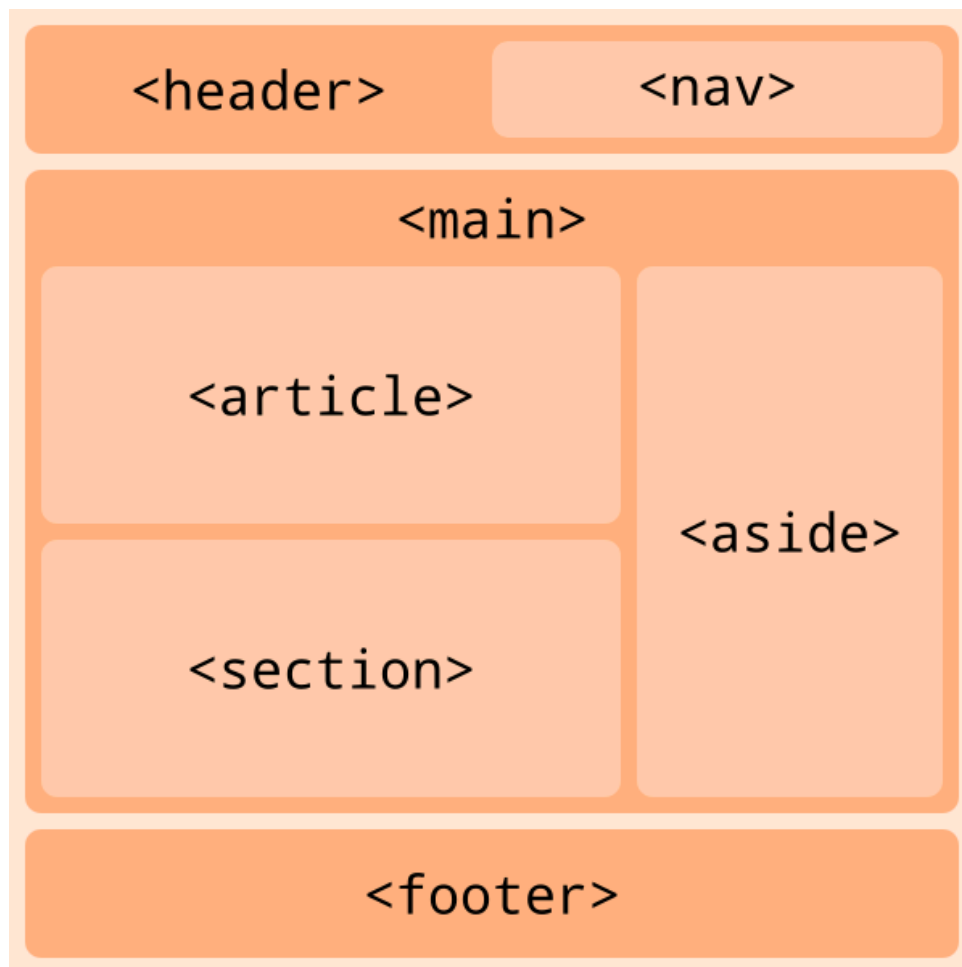


Estructura Documento HTML.....	5
Link CSS y ICON.....	5
Fuentes Link.....	5
@Font-face.....	6
Overflow.....	6
• Visible.....	6
• Hidden.....	6
• Scroll.....	7
• Auto.....	7
Position.....	8
• Absolute.....	8
• Static.....	8
• Fixed.....	9
Z-Index.....	9
Selectores Hijos.....	9
Selectores de Hijos Específicos.....	10
:first-child.....	10
:last-child.....	10
:nth-child(n).....	10
:nth-of-type(n).....	11
:first-of-type.....	11
:last-of-type.....	11
:only-child.....	11
:only-of-type.....	11
Pseudo-clases de estado.....	11
:hover.....	11
:active.....	11
:focus.....	11
:focus-visible.....	11
:focus-within.....	11
Pseudo-clases de Enlaces.....	11
:link.....	11
:visited.....	11
@Keyframes.....	12
Elementos Individuales:.....	12
animation-name.....	12
Animation-duration.....	12
animation-timing-function.....	13
animation-delay.....	13
animation-iteration-count.....	13
animation-direction.....	13

animation-fill-mode.....	13
animation-play-state.....	14
Sintaxis.....	14
Propiedades Animar:.....	14
transform.....	14
opacity.....	14
width, height.....	14
left, top, right, bottom.....	14
margin, padding.....	14
background-color, color, border-color,box-shadow.....	14
SASS.....	15
instalar SASS.....	15
Variables.....	17
Anidación.....	18
Mixins.....	18
Operaciones.....	19
Condiciones (@if, @else if, @else).....	19
FLEXBOX.....	20
Display.....	20
Flex.....	20
Inline-Flex.....	20
Flex-direction.....	21
Column.....	21
Row-Reverse.....	21
Row.....	21
Flex-Wrap.....	21
nowrap.....	21
wrap.....	21
Order.....	21
1,2,3,4,5.....	21
Flex-grow.....	21
1.....	21
2.....	21
3.....	21
Flex-shrink.....	22
0.....	22
1.....	22
2.....	22
Justify-content.....	22
Flex-start.....	22
Center.....	22

flex-end.....	22
space-between.....	22
space-around.....	22
space-evenly.....	22
Align-Items.....	22
Stretch.....	22
Flex-Start.....	22
Center.....	22
Flex-end.....	22
Baseline.....	22
Grid.....	23
Display.....	23
Grid.....	23
Grid-gap,row-gap,column-gap.....	23
px.....	23
Grid-template-columns,Grid-template-rows.....	23
px.....	23
%.....	23
fr.....	23
auto.....	23
repeat().....	23
minmax().....	23
Grid-column.....	23
Inicio/final.....	23

Estructura Documento HTML



Link CSS y ICON

```
<link rel="stylesheet" href="proyecto.css">  
<link rel="icon" type="image/x-icon" href="img/img/logo.png">
```

Fuentes Link

Anclar fuentes por enlace, Pegar enlace en html

```
<link rel="preconnect" href="https://fonts.googleapis.com">  
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>  
<link href="https://fonts.googleapis.com/css2?family=Boldonse&display=swap" rel="stylesheet">
```

En CSS llamar a la fuente

```
font-family: "boldonse"
```

[Enlace Fuentes](#)

@Font-face

Sirve para cuando hemos descargado una fuente y queremos anclarla desde css

```
@font-face {  
  font-family: boldose;  
  src: url(fuentes/Boldonse-Regular.ttf);  
}
```

[Descargar Fuentes](#)

Overflow

Se utiliza para cuando el contenido es demasiado grande y sobrepasa los elementos.

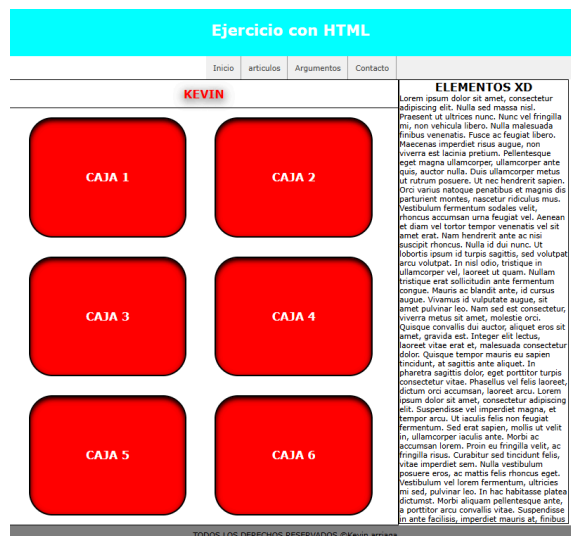
- **Visible**

Es el predeterminado, no recorta el contenido y lo deja feo sin respetar los demás elementos



- **Hidden**

Borra todo lo que se sale del caja



- **Scroll**

Crea una barra lateral y horizontal para poder ver todo el contenido de una manera más agradable

En caso de que no queramos una barra horizontal o vertical podemos usar overflow-x o -y para los ejes

Ejercicio con HTML

InicioartículosArgumentosContacto

KEVIN

CAJA 1

CAJA 2

CAJA 3

CAJA 4

CAJA 5

CAJA 6

TODOS LOS DERECHOS RESERVADOS ©Kevin arraga

ELEMENTOS XD
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla sed massa nisi. Praesent ut ultrices nunc. Nunc vel fringilla mi, non vehicula libero. Nulla malesuada finibus venenatis. Fusce ac feugiat libero. Maecenas imperdiet risus augue, non viverra est lacina pretum. Pellentesque eget magna ullamcorper, ullamcorper ante quis, auctor nulla. Duis ullamcorper metus ut rutrum posuere. Ut nec hendrent sapien. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Vestibulum fermentum sodales velit, rhoncus accumsan urna feugiat vel. Aenean et diam vel tortor tempor venenatis vel sit amet erat. Nam hendrent ante ac nisi suscipit rhoncus. Nulla id dui nunc. Ut lobortis ipsum id turpis sagittis, sed volutpat arcu volutpat. In nisl odio, tristique in ullamcorper vel, laoreet ut quam. Nullam tristique erat sollicitudin ante fermentum congue. Mauris ac blandit ante, id cursus augue. Vivamus id vulputate augue, sit amet pulvinar leo. Nam sed est consectetur, viverra metus sit amet, molestie orci. Quisque convallis dui auctor, aliquet eros sit amet, gravida est. Integer elit lectus, laoreet vitae erat et, malesuada consectetur dolor. Quisque tempor mauris eu sapien tincidunt, at sagittis ante aliquet. In pharetra sagittis dolor, eget porttitor turpis consectetur vitae. Phasellus vel felis laoreet, dictum orci accumsan, laoreet arcu. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse vel imperdiet magna, et tempor arcu. Ut iaculis felis non feugiat fermentum. Sed erat sapien, mollis ut velit in, ullamcorper iaculis ante. Morbi ac accumsan lorem. Proin eu fringilla velit, ac fringilla risus. Curabitur sed tincidunt felis, vitae imperdiet sem. Nulla vestibulum posuere eros, ac mattis felis rhoncus eod. Vestibulum vel lorem

- **Auto**

Es como el scroll, solo que este se crea si es necesario

Ejercicio con HTML

InicioartículosArgumentosContacto

KEVIN

CAJA 1

CAJA 2

CAJA 3

CAJA 4

CAJA 5

CAJA 6

TODOS LOS DERECHOS RESERVADOS ©Kevin arraga

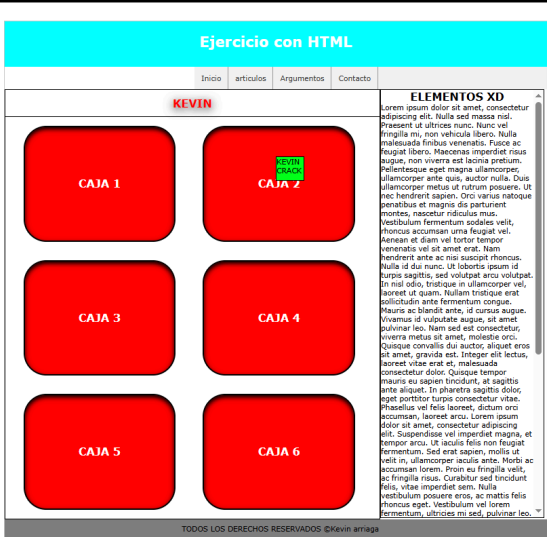
ELEMENTOS XD
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla sed massa nisi. Praesent ut ultrices nunc. Nunc vel fringilla mi, non vehicula libero. Nulla malesuada finibus venenatis. Fusce ac feugiat libero. Maecenas imperdiet risus augue, non viverra est lacina pretum. Pellentesque eget magna ullamcorper, ullamcorper ante quis, auctor nulla. Duis ullamcorper metus ut rutrum posuere. Ut nec hendrent sapien. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Vestibulum fermentum sodales velit, rhoncus accumsan urna feugiat vel. Aenean et diam vel tortor tempor venenatis vel sit amet erat. Nam hendrent ante ac nisi suscipit rhoncus. Nulla id dui nunc. Ut lobortis ipsum id turpis sagittis, sed volutpat arcu volutpat. In nisl odio, tristique in ullamcorper vel, laoreet ut quam. Nullam tristique erat sollicitudin ante fermentum congue. Mauris ac blandit ante, id cursus augue. Vivamus id vulputate augue, sit amet pulvinar leo. Nam sed est consectetur, viverra metus sit amet, molestie orci. Quisque convallis dui auctor, aliquet eros sit amet, gravida est. Integer elit lectus, laoreet vitae erat et, malesuada consectetur dolor. Quisque tempor mauris eu sapien tincidunt, at sagittis ante aliquet. In pharetra sagittis dolor, eget porttitor turpis consectetur vitae. Phasellus vel felis laoreet, dictum orci accumsan, laoreet arcu. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse vel imperdiet magna, et tempor arcu. Ut iaculis felis non feugiat fermentum. Sed erat sapien, mollis ut velit in, ullamcorper iaculis ante. Morbi ac accumsan lorem. Proin eu fringilla velit, ac fringilla risus. Curabitur sed tincidunt felis, vitae imperdiet sem. Nulla vestibulum posuere eros, ac mattis felis rhoncus eod. Vestibulum vel lorem fermentum, ultrices mi sed, pulvinar leo.

Position

Especifica el tipo de posicionamiento que contiene el elemento.(fijarse caja verde)

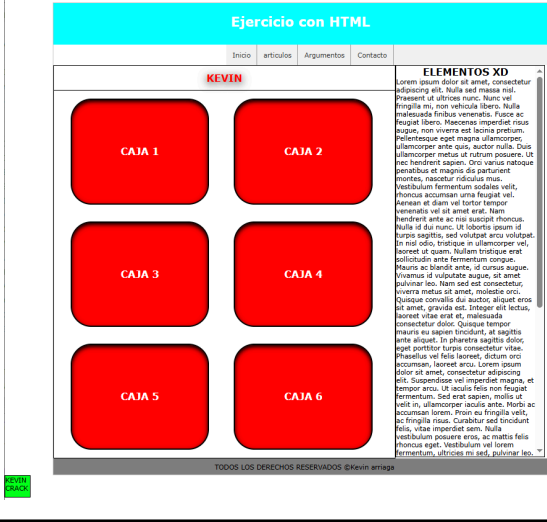
- Absolute

El elemento se postpone por encima de los demás elementos sin respetarlos



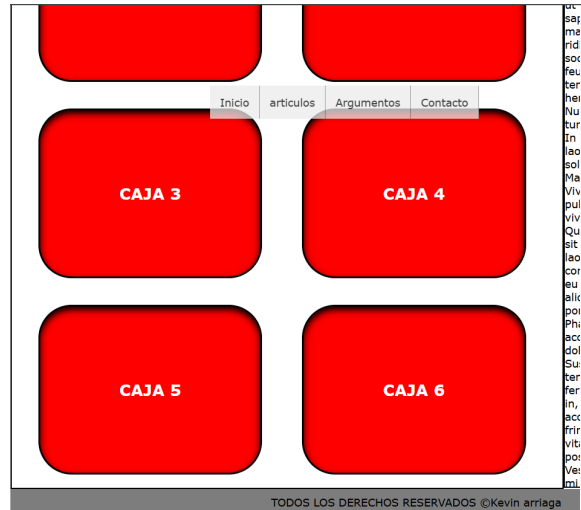
- Static

Es el predeterminado, no sirve pa na ZZZ



- **Fixed**

Es como el absolute en el momento de no respetar los demás elementos, pero añade que a la hora de hacer scroll hacia abajo nos sigue, útil para crear menús o índices que se mantengan (fijarse en el barra del menú como nos sigue al hacer scroll)



Z-Index

Sirve para indicar en qué capa se encuentra el elemento útil para posponer o sobreponer sobre los elementos que tenemos.

- -1
- 0
- 1

(**z-index** solo funciona en elementos posicionados (absoluta, relativa y fija) y elementos flex (elementos que son hijos directos de elementos [display:flex](#)).

Selectores Hijos

selector > selector → **Hijo directo**

Este selector aplica estilos **solo a los hijos directos** (inmediatos) de un elemento padre.

```
<ul id="menu">
  <li> <a href="#">Inicio</a></li>
  <li> <a href="#">Temario</a></li>
  <li> <a href="#">Reseñas</a>
    <ul>
      <li><a href="">HTML</a></li>
      <li><a href="">CSS</a></li>
      <li><a href="">JAVA</a></li>
    </ul>
  </li>
  <li> <a href="#">Contacta</a></li>
</ul>
```

```
#menu > li > a{
  font-size: 18px;
  color: red;
  text-decoration: none;
}
```

[Inicio](#)
[Temario](#)
[Reseñas](#)
[HTML](#)
[CSS](#)
[JAVA](#)
[Contacta](#)

selector selector → **Descendiente (cualquier nivel)**

```
<footer>
  Fin de la web, Saludos <a href="http://">LINK</a>
</footer>
```

```
footer a {
  font-size: 20px;
  color: red;
}
```

Fin de la web, Saludos **LINK**

Selectores de Hijos Específicos

:first-child	Selecciona un elemento que es el primer hijo de su padre	Ejemplo
:last-child	Selecciona un elemento que es el último hijo de su padre.	
:nth-child(n)	Selecciona un hijo según su posición numérica exacta dentro	

	del padre.	<pre>ul li:first-child{ color: purple; font-weight: bold; } ul li:nth-child(2){ color: rgb(0, 128, 81); font-weight: bold; } ul li:last-child{ color: rgb(0, 117, 128); font-weight: bold; }</pre> <ul style="list-style-type: none"> • HTML • JAVA • CSS • PHP • SQL <p>Rockstargames.com</p> <input type="text"/>
:nth-of-type(n)	Similar a :nth-child(n), pero solo cuenta los elementos del mismo tipo (mismo nombre de etiqueta).	
:first-of-type	Selecciona el primer hijo de un tipo específico dentro de su padre.	
:last-of-type	Selecciona el último hijo de un tipo específico.	
:only-child	Selecciona un elemento si y sólo si es el único hijo de su padre.	
:only-of-type	Selecciona un elemento si es el único de su tipo dentro de su padre.	

Pseudo-clases de estado

:hover	Aplica estilos cuando el usuario pasa el puntero (mouse) sobre un elemento.
:active	Se aplica mientras el elemento está siendo presionado (clic sostenido).
:focus	Se aplica cuando el elemento recibe el foco (por teclado o clic).
:focus-visible	Igual que :focus, pero solo cuando el foco es visible (usado por teclado, no por clic).
:focus-within	Se aplica a un contenedor si alguno de sus elementos hijos tiene el foco

Pseudo-clases de Enlaces

:link	Se aplica a enlaces que aún no han sido visitados.
:visited	Se aplica a enlaces que ya fueron visitados.

@Keyframes

Los **keyframes** en CSS se usan para crear **animaciones** que cambian con el tiempo. Básicamente, definen una secuencia de estilos que deben aplicarse a un elemento en ciertos puntos durante una animación.

Sintaxis Básica

```
@keyframes Ejemplo {  
  from{  
    /* Estilos iniciales */  
  }  
  to{  
    /* Estilos finales */  
  }  
}
```

Sintaxis más detallada

```
@keyframes Ejemplo {  
  0%{  
    /* Estilos iniciales */  
  }  
  50%{  
    /* Estilos Intermedios */  
  }  
  100%{  
    /* Estilos finales */  
  }  
}
```

Para usar una animación y aplicarla a un elemento, se utiliza la propiedad **animation** o sus propiedades individuales:

```
.animacion {  
  animation-name: nombre-animacion;  
  animation-duration: 2s;  
  animation-timing-function: ease;  
  animation-delay: 0s;  
  animation-iteration-count: infinite;  
  animation-direction: alternate;  
}
```

Elementos Individuales:

animation-name	Especifica el nombre de la animación definida con @keyframes .
Animation-duration	Define la duración total de la animación.
	Tiempo en segundos (s) o milisegundos (ms)

animation-timing-function	Determina la curva de velocidad de la animación, es decir, cómo varía la velocidad durante el tiempo .	
	Linear	Velocidad constante
	ease	Comienza lento, acelera, termina lento.
	ease-in	Comienza Lento
	ease-out	termina lento
	ease-in-out	comienza y termina lento.
	steps(n)	avanza en n saltos abruptos.
	cubic-bezier(x1,y1,x2,y2)	Define una curva de tiempo personalizada.
animation-delay	Establece un retraso antes de que comience la animación.	
	Tiempo en segundos (s) o milisegundos (ms)	
animation-iteration-count	Indica cuántas veces se debe repetir la animación.	
	número → Cuántas veces repetir (ej: 1, 3, 10) infinite → Repetición sin fin	
animation-direction	Determina en qué dirección se reproduce la animación:	
	normal	La animación se reproduce de principio a fin (0% → 100%).
	reverse	Se reproduce al revés (100% → 0%).
	alternate	Alterna entre normal y reverse en cada ciclo.
	alternate-reverse	Alterna como alternate, pero empieza con la animación en reversa.
animation-fill-mode	Controla si los estilos definidos por la animación se mantienen antes o después de que termine la animación.	
	none	(Por defecto) No se conserva ningún estilo de la animación.
	forwards	Aplica los estilos finales (100% o to) después de finalizar.
	backwards	Aplica los estilos iniciales (0% o from) durante el retraso (delay).
	both	Combina forwards y backwards.

animation-play-state	Controla si una animación está en curso o pausada.
	running (por defecto)/ paused
Sintaxis	animation: rebotar 1.5s ease-in-out 0s infinite alternate both;

Propiedades Animar:

No todas las propiedades de CSS pueden ser animadas, pero muchas sí. A continuación, te explico **cuáles son, cómo se comportan al animarse y ejemplos concretos.**

transform	Una de las propiedades más potentes y versátiles. Se utiliza para aplicar transformaciones visuales.	
	translateX(n),translateY(n), translateZ(n)	Desplaza el elemento en el eje X, Y o Z.
	scale(n) / scaleX(n) / scaleY(n)	Escala el tamaño del elemento en una o más direcciones.
	rotate(deg) / rotateX(deg) / rotateY(deg)	Rota el elemento en 2D o 3D.
	skewX(deg), skewY(deg)	Inclina el elemento en el eje correspondiente.
opacity	0 , 1	0 (totalmente transparente) 1 (completamente opaco)
width, height	Se pueden usar con valores absolutos (px, em) o relativos (% ,vw, etc.).	Permiten animar el tamaño del elemento .
left, top, right, bottom	Se pueden usar con valores absolutos (px, em) o relativos (% ,vw, etc.).	Controlan la posición del elemento, útil para moverlo.(Solo funcionan si el elemento tiene position: relative, absolute o fixed.)
margin, padding	Se pueden usar con valores absolutos (px, em) o relativos (% ,vw, etc.).	Controlan el espacio interno (padding) y externo (margin) de un elemento.
background-color, color, border-color , box-shadow	Colores se pueden animar entre cualquier valor (por nombre, hex, rgb, etc.).	box-shadow puede usarse para generar efectos visuales de profundidad.

SASS

Sass (Syntactically Awesome Stylesheets) es un preprocesador de CSS, es decir, una herramienta que extiende el CSS con características que no tiene por defecto, como:

- Variables (\$color: red;)
- Anidación (como en HTML, puedes anidar selectores)
- Mixins (reutilizar bloques de código)
- Funciones y operadores (realizar cálculos)

Escribes tu estilo en un archivo .scss o .sass, y luego se compila a CSS normal que entiende el navegador

instalar SASS

Debemos instalar [Node.JS](#) para poder hacer funcionar SASS.



Terminada la instalación del programa, para comprobar que se ha instalado correctamente deberemos de abrir “CMD”.

Para ver la versión “Node -v”

versión de paquetes “npm -v”

```
Símbolo del sistema
Microsoft Windows [Versión 10.0.19045.5854]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\kevin.arriaga>node -v
v22.15.1

C:\Users\kevin.arriaga>
```

```
C:\Users\kevin.arriaga>npm -v
10.9.2

C:\Users\kevin.arriaga>
```

instalar paquetes “npm install -g sass”

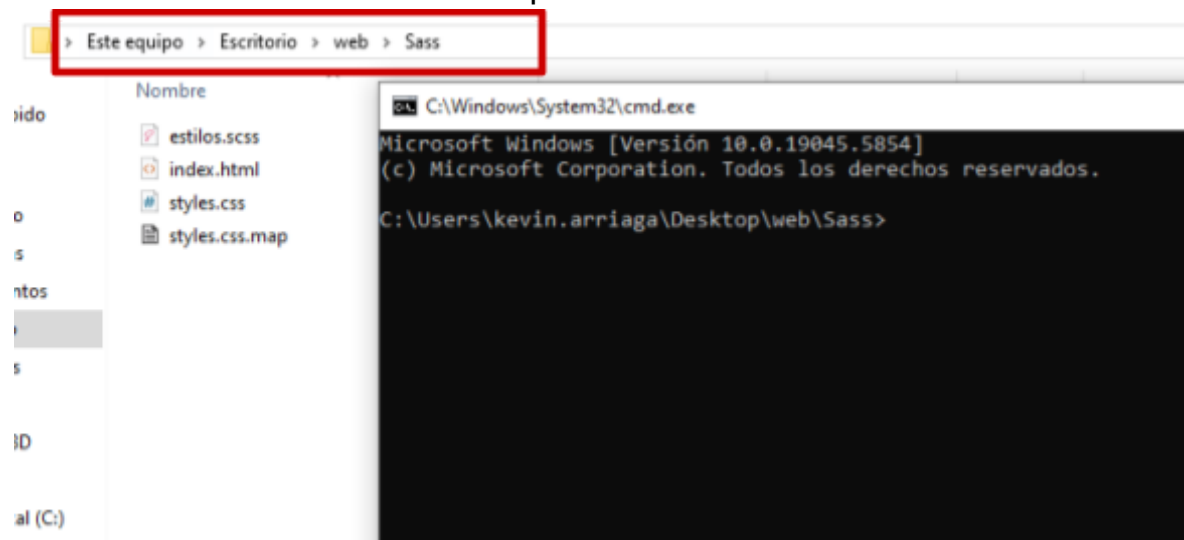
```
Símbolo del sistema
C:\Users\kevin.arriaga>npm install -g sass
npm warn cleanup Failed to remove some directories [
npm warn cleanup   'C:\Users\kevin.arriaga\AppData\Roaming\npm\node_modules\sass-1hEclZsB',
npm warn cleanup   [Error: EPERM: operation not permitted, unlink 'C:\Users\kevin.arriaga\AppData\Roaming\npm\node_modules\sass-1hEclZsB\node_modules\@parcel\watcher-win32-x64\watcher.node'] {
npm warn cleanup     errno: -4048,
npm warn cleanup     code: 'EPERM',
npm warn cleanup     syscall: 'unlink',
npm warn cleanup     path: 'C:\Users\kevin.arriaga\AppData\Roaming\npm\node_modules\sass-1hEclZsB\node_modules\@parcel\watcher-win32-x64\watcher.node'
npm warn cleanup   }
npm warn cleanup ]
npm warn cleanup ]

changed 17 packages in 6s

5 packages are looking for funding
  run `npm fund` for details
```

Con esto ya tendríamos las comprobaciones completas, ahora haremos que las configuraciones del sass se pasen a un archivo css.
Para compilar un archivo scss a un css deberemos de realizar los siguientes pasos.
crear un archivo .scss y uno css.

Debemos abrir nuestra carpeta donde tenemos los ficheros y abrir cmd desde la barra de enlace, borramos toda dirección de carpetas y ponemos “CMD” y se abrirá cmd con la dirección de la carpeta.



y para compilar el archivo scss a css debemos escribir lo siguiente “sass –watch estilos.scss styles.css” con esto le estamos diciendo que pase lo que tenemos escrito en sass a css y con este ya estaría funcionando correctamente.

```
C:\WINDOWS\system32\cmd.exe - "node" "C:\Users\kevin.arriaga\AppData\Roaming\npm\node_modules\sass\sass.js" --watch estilos.scss styles.css
C:\Users\kevin.arriaga\Desktop\web\Sass>sass --watch estilos.scss styles.css
[2025-05-20 12:49] Compiled estilos.scss to styles.css.
Sass is watching for changes. Press Ctrl-C to stop.

Error: Missing argument $color.

49 | @mixin crearborders($color, $zise, $shadow){
    |                      ^^^^^^ declaration
...
56 |     @include crearborders();
    |     ^^^^^^^^^^^^^^^^^^^^^ invocation

estilos.scss 56:5  crearborders()
estilos.scss 56:5  root stylesheet
Error: Missing argument $color.

49 | @mixin crearborders($color, $zise, $shadow){
    |                      ^^^^^^ declaration
...
56 |     @include crearborders();
    |     ^^^^^^^^^^^^^^^^^^^^^ invocation

estilos.scss 56:5  crearborders()
estilos.scss 56:5  root stylesheet
Error: Undefined variable.

50 |     border: $size 5px solid red;
    |            ^^^^^
estilos.scss 50:13  crearborders()
```

Variables

Sirven para guardar valores reutilizables, como colores o tamaños.

SCSS

```
// variables
$fuente-tipca: sans-serif, helvetica, arial;
$fondo-color: #f6f28a;
$letra: #00b3ff;
$fondo2: #c28af6;
```

HTML

```
// uso de variables

body{
  background-color: $fondo-color;
}

h1{
  font-family: $fuente-tipca;
  color: $letra;
}
```

Resultado

Aprendiendo SASS

Sass es un preprocesador css muy popular y potente

Anidación

Permite escribir estilos dentro de otros, como si siguieras la estructura del HTML.

HTML

```
<div class="caja">
  <h1>Caja</h1>
  <p>Soy un parrafo</p>
  <div class="info">
    16/05/2025
  </div>
  <span class="author">
    Kevin Arriaga
  </span>
</div>
```

SCSS

```
.caja{
  border: 1px solid black;
  width: 500px;
  margin: 0px auto;
  text-align: center;
  background-color:$fondo2 ;
  padding: 10px;

  h1{
    color: blue;
    text-decoration:underline red;
  }
  p{
    color: green;
  }
  .info{
    font-size: 50px;
    .author{
      font-size: 15px;
      color: rgb(234, 157, 101);
    }
  }
}
```

Mixins

Sirven para **reutilizar bloques de código** con o sin parámetros.

html

```
<div class="caja">
  <h1>Caja</h1>
  <p>Soy un parrafo</p>
  <div class="info">
    16/05/2025
  </div>
  <span class="author">
    Kevin Arriaga
  </span>
</div>

<!-- <div class="caja">...

<div class="listado">
  <ul>
    <li>SASS</li>
    <li>LESS</li>
    <li>STYLUS</li>
  </ul>
</div>
```

SCSS

```
@mixin crearborders($color, $size, $shadow){
  border: $size solid $color;
  border-radius: $size ;
  box-shadow:0px 0px $size $shadow;
}

.caja{
  @include crearborders(red,10px,black);
}

.listado{
  @include crearborders(green,20px,yellow);
}
```

Resultado



Operaciones

Puedes **hacer cálculos** y usar funciones propias o de Sass.

```
// // operadores
.clearfix{
  clear: both;
}

h1{
  font-size: 20px + 5px - 1px;
}

.cajas{
  width: 900px;
}

.caja{
  width: 500px / 900px * 100%;
  float: left;
}
```

Condiciones (@if, @else if, @else)

Sirven para aplicar estilos solo si se cumple cierta condición.

HTML

```
<body>
<h1>Aprendiendo SASS</h1>
<P>Sass es un preprocesador css muy popular y potente</P>
```

SCSS

```
// condiciones -if
h1{
  @if $fondo-color == #ccc{
    color: black;
  }@else{
    color: lightcoral;
  }
}
```

Ejemplo

Aprendiendo SASS

Aprendiendo SASS

HTML

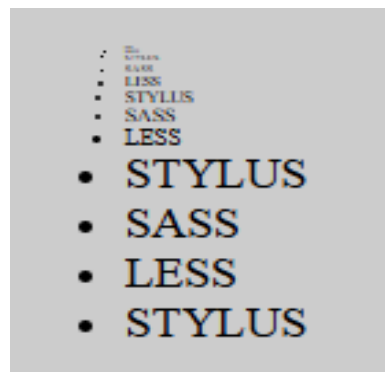
```
<div class="listado">
  <ul>
    <li>SASS</li>
    <li>LESS</li>
    <li>STYLUS</li>
    <li>SASS</li>
    <li>LESS</li>
    <li>STYLUS</li>
    <li>SASS</li>
    <li>LESS</li>
    <li>STYLUS</li>
  </ul>
</div>
```

SCSS

```
// condiciones -for
$contador: 1;

@for $contador from 1 through 8{
  .listado ul li:nth-child(#{ $contador }){
    font-size:#{ $contador }px;
  }
}
```

Resultado



FLEXBOX

Cuando aplicas **display: flex** a un **contenedor**, lo conviertes en un **contenedor flexible**. Esto cambia la forma en que se distribuyen sus **elementos hijos**, que ahora se llaman **ítems flexibles** o **flex items**.

Display

Flex	Coloca todos los elementos en horizontal (contenido sobresalen)
Inline-Flex	Adapta el contenido interno al div o sin que se sobresalga

Flex-direction

Perfecto, vamos a centrarnos en flex-direction, una de las propiedades más importantes de Flexbox. Esta propiedad define la dirección del eje principal: es decir, cómo se colocan los elementos dentro de un contenedor con display: flex

Column	Posiciona los elementos que se hayan seleccionado el columnas
Row-Reverse	Invierte el orden de y mueve el contenido a la inversa
Row	es la que viene por defecto

Flex-Wrap

Por defecto, los elementos en un contenedor display: flex se colocan todos en una sola línea, aunque no quepan. Esto puede hacer que se reduzcan de tamaño o que se desborden. La propiedad flex-wrap te permite controlar si los ítems pueden "envolverse" (wrap) a la siguiente línea cuando no hay suficiente espacio en una sola fila o columna.

nowrap	Deja que los elementos sobresalgan
wrap	Hace que los elementos mantengan la posición

Order

La propiedad order cambia el orden visual de los elementos flexibles (los hijos de un contenedor con display: flex) sin cambiar el HTML.

1,2,3,4,5	Esto es especialmente útil si necesitas reorganizar elementos para distintas resoluciones o estilos sin tocar la estructura del documento.
------------------	--

Flex-grow

La propiedad **flex-grow** determina **cuánto puede crecer un ítem flexible** para ocupar el espacio disponible **dentro del contenedor**. Es decir, cuando hay **espacio sobrante**, los elementos con **flex-grow** se agrandan proporcionalmente según su valor.

1	el elemento puede crecer para llenar espacio
2	crecerá el doble que uno con flex-grow: 1
3	no crecerá , aunque haya espacio disponible.

Flex-shrink

Determina cuánto se reducirá un elemento en proporción a los demás cuando no haya suficiente espacio disponible en el contenedor con `display: flex`.

0	no se encoge nunca , aunque no quepa
1	se encoge normalmente.
2	se encoge el doble que un elemento con valor 1.

Justify-content

La propiedad `justify-content` se aplica al **contenedor** con `display: flex` y sirve para **alinear los ítems a lo largo del eje principal** (horizontal si `flex-direction: row`, o vertical si `flex-direction: column`).

Flex-start	Los ítems se alinean al inicio del eje principal.
Center	Los ítems se alinean en el centro .
flex-end	Los ítems se alinean al final del eje principal.
space-between	Espacio entre los ítems (el primero y último pegados a los bordes).
space-around	Espacio alrededor de cada ítem (mitad al principio y final).
space-evenly	Espacio igual en todos lados (antes, entre y después de cada ítem).

Align-Items

La propiedad `align-items` se aplica al contenedor flex y sirve para alinear los ítems hijos en el eje perpendicular al eje principal

Stretch	(valor por defecto) Estira los ítems para que ocupen todo el eje cruzado.
Flex-Start	Alinea los ítems al inicio del eje cruzado .
Center	Alinea los ítems en el centro del eje cruzado.
Flex-end	Alinea los ítems al final del eje cruzado .
Baseline	Alinea los ítems según su línea base del texto .

Grid

CSS Grid Layout (o simplemente Grid) es un sistema de diseño en CSS que te permite crear estructuras de diseño de dos dimensiones (filas y columnas) de una manera sencilla y poderosa.

A diferencia de Flexbox (que se enfoca en una sola dimensión), Grid puede controlar tanto filas como columnas al mismo tiempo, lo que lo hace ideal para construir diseños más complejos.

Display

Grid	Se define el grid layout
-------------	--------------------------

Grid-gap,row-gap,column-gap

px	Es una propiedad abreviada en CSS Grid que se usa para establecer el espacio entre las filas y columnas del grid.
-----------	--

Grid-template-columns,Grid-template-rows

define cuántas columnas tendrá el grid y cuánto espacio ocupará cada una.

px % fr auto	<ul style="list-style-type: none">• Tamaño fijo en píxeles,• Porcentaje del ancho del contenedor• Fracción del espacio disponible (muy útil)• El ancho se ajusta al contenido
repeat()	Repetir columnas automáticamente
minmax()	Cada columna será mínimo 200px, pero puede crecer.

Grid-column

Definir en qué columna empieza y en cuál termina un elemento del grid. Se puede usar para hacer que un elemento ocupe varias columnas. nos ahorra el escribir esta línea grid-column-start/grid-column-end

Inicio/final	Cada columna será mínimo 200px, pero puede crecer.
---------------------	--