Contenido

[CSS 3](#_Toc111551349)

[Referencia 3](#_Toc111551350)

[Referencia Colores 3](#_Toc111551351)

[Sintaxis 3](#_Toc111551352)

[Style 3](#_Toc111551353)

[Estilo dentro de una etiqueta span 3](#_Toc111551354)

[Estilo dentro de una etiqueta particular 3](#_Toc111551355)

[Práctica CSS 1 4](#_Toc111551356)

[Head - Style 4](#_Toc111551357)

[Ejemplo 4](#_Toc111551358)

[Estilo dentro del encabezado 4](#_Toc111551359)

[Estilo haciendo uso de un archivo externo 5](#_Toc111551360)

[Importar el archivo CSS 5](#_Toc111551361)

[Estilo haciendo uso de la sentencia import de CSS 5](#_Toc111551362)

[Práctica 2 CSS 6](#_Toc111551363)

[Jerarquía CSS 6](#_Toc111551364)

[Práctica 3 css 6](#_Toc111551365)

[Tamaños de fuentes 7](#_Toc111551366)

[Unidades relativas 7](#_Toc111551367)

[Práctica 4 css 8](#_Toc111551368)

[Unidades absolutas 8](#_Toc111551369)

[Otros ejemplos 8](#_Toc111551370)

[Práctica 5 css 8](#_Toc111551371)

[Margin, Border, padding 8](#_Toc111551372)

[Práctica 6 css 9](#_Toc111551373)

[Práctica 7 css 9](#_Toc111551374)

[Selectores 9](#_Toc111551375)

[Etiqueta HTML 9](#_Toc111551376)

[ID 9](#_Toc111551377)

[Clase 9](#_Toc111551378)

[Práctica 8 css 9](#_Toc111551379)

[CSS, selector “>” 10](#_Toc111551380)

[Práctica 9 css 10](#_Toc111551381)

[CSS Flexbox – Flex container 10](#_Toc111551382)

[El contenedor flex 11](#_Toc111551383)

[Práctica 10 12](#_Toc111551384)

[Práctica 11 12](#_Toc111551385)

[Práctica 12 12](#_Toc111551386)

[Propiedades aplicadas a los ítems 13](#_Toc111551387)

[Espacio disponible 13](#_Toc111551388)

[Flex-basis 13](#_Toc111551389)

[Flex-grow 13](#_Toc111551390)

[Práctica 13 13](#_Toc111551391)

[flex-shrink 14](#_Toc111551392)

[Práctica 14 14](#_Toc111551393)

[Abreviaciones. 14](#_Toc111551394)

[align-items 14](#_Toc111551395)

[Práctica 15 15](#_Toc111551396)

[justify-content 15](#_Toc111551397)

[Práctica 16 15](#_Toc111551398)

[Propiedades 16](#_Toc111551399)

[Flex items 17](#_Toc111551400)

[Propiedades 17](#_Toc111551401)

[@media 19](#_Toc111551402)

[Flex Responsivo 19](#_Toc111551403)

[Ver ejemplo 21](#_Toc111551404)

[Ver otro ejemplo. 21](#_Toc111551405)

# CSS

Cascading Style Sheets (Hojas de estilo en cascada)

Al igual que HTML, CSS se define en base a un estándar publicado por W3C.

* Se propagan en cascada a los elementos

## Referencia

<https://www.w3schools.com/css/default.asp>

Lenguaje de diseño gráfico

Su función principal es separar el contenido de la presentación.

Otra ventaja de CSS es la capacidad para especificar las medidas con diversas unidades:

* Pixeles (px)
* Porcentaje (%)
* Puntos (pt)
* Pulgadas (in)
* Centímetros (cm)
* Entre otras..

## Referencia Colores

<https://htmlcolorcodes.com/es/>

# Sintaxis

## Style

## Estilo dentro de una etiqueta span

<p>Esto es un parrafo <span style="color:green;"> esto va en color verde</span> </p>

## Estilo dentro de una etiqueta particular

<h1 style="color:#21B4DC;">Título con color azul.. </h1>

<p style="font-weight:bold;">

Parrafo en negritas.

</p>

<div style="background-color:#BCE93A; color:#F82005">

<p>

Lorem ipsum

</p>

<hr>

<p>

Lorem ipsum dolor sit amet,

</p>

</div>





## Práctica CSS 1

1. Crear archivo con 2 etiquetas H1 y H2 con colores diferentes.
2. Crear un párrafo con el texto de color #21B4DC y letra “bold”
3. Crear un div con fondo de color #BCE93A y color de texto #F82005
   1. Dentro del div, crear 2 párrafos sin Ningún estilo

## Head - Style

A picture containing text

Description automatically generated

## Ejemplo

|  |  |
| --- | --- |
| A picture containing graphical user interface  Description automatically generated | * p, es el selector em CSS (Elemento al cual se quiere colocar el “estilo” * color, es la propiedad * red, es el valor de la propiedad |

## Estilo dentro del encabezado

<style type="text/css">

p {

color:red;

}

h1 {

color:blue;

background-color:red;

}

ul {

border: 3px solid #cccccc;

}

li {

color: blue;

font-size:18pt;

}

</style>

# Estilo haciendo uso de un archivo externo

<link rel="stylesheet" type="text/css" href="estilos\_hoja3.css">

p {

font-size: 18pt;

background-color: #F82005;

color:aqua;

}

h1 {

background-color: black;

color:blanchedalmond;

}

ul {

border: 3px solid black;

}

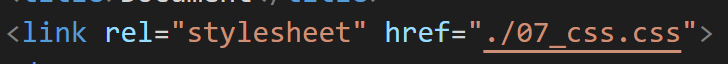
body {

background-color: chartreuse;

font-family: Verdana, Geneva, Tahoma, sans-serif;

}

## Importar el archivo CSS



## Estilo haciendo uso de la sentencia import de CSS

<style type="text/css">

@import url("estilos\_hoja3.css");

</style>

## Práctica 2 CSS

1. Crear los estilos dentro del tag <style> dentro del encabezado y verificar su funcionamiento
2. Crear un archivo .css e incluirlo con la notación
   1. <link rel="stylesheet" type="text/css" href="nombre\_archivo.css">
3. Verificar el funcionamiento ahora con la notación
   1. @import url("nombre\_archivo.css");

# Jerarquía CSS

De Menor a mayor prioridad se muestra la jerarquía.

* Archivo Externo
* Etiqueta Style dentro de la cabecera de la página
* Definidos en una etiqueta en particular
* Declarados dentro de la etiqueta span

En caso de existir 2 o más archivos y exista conflicto, tiene mayor prioridad el último de ellos.

Ejemplo:

Graphical user interface, application

Description automatically generated A screenshot of a computer

Description automatically generated with low confidence

A picture containing graphical user interface

Description automatically generated

## Práctica 3 css

1. Crear 2 archivos css, incluirlos en el html
   1. Modificar la misma etiqueta, por ejemplo el color de H1
   2. Verificar su funcionamiento.
2. Modificar el HTML y colocar un style, dentro del H1, modificando el color.
   1. Verificar su funcionamiento, mediante el inspector del navegador

Graphical user interface, text

Description automatically generated

# Tamaños de fuentes

### Unidades relativas

* em (Relativa a la fuente actual)
  + La **unidad em** tiene como referencia de medida el font-size del elemento actual.
  + Por ejemplo si la fuente actual es de 12 puntos, entonces 2em = 24 puntos.
* rem
  + La **unidad rem** tiene como referencia de medida el font-size del elemento root, la etiqueta <html> (en inglés sería como "root em").
* ex (1 ex es igual a la altura de la letra x, según la fuente actual del usuario)
  + Generalmente es la mitad de la fuente normal.
* px (Un pixel = punto en la pantalla)
  + Depende de la resolución de la pantalla del dispositivo.

A picture containing diagram

Description automatically generated Text

Description automatically generated

## Práctica 4 css

1. Crear un archivo html y dentro del style cambiar el Font-size dentro de los tags
   1. Html (rem)
   2. Body (em)
2. Crear 2 parrafos, uno con tamaño de fuente 1.2 em y el otro 1.2rem

### Unidades absolutas

* pt = 1/72 pulgadas
* cm = centimetro
* in = pulgada
* mm = milímetros
* pc = 12 puntos (pica) = 1/6 pulgada

# Otros ejemplos

p{

margin-top:10px;

backgorund-color:rgb(248, 32, 5);

}

body {

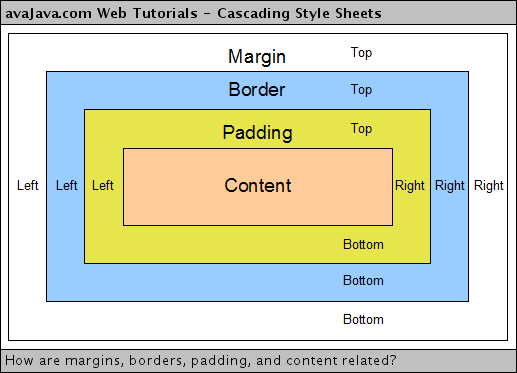
background-image: url("carita1.jpg");

}

## Práctica 5 css

1. Crear un html para verificar las propiedades margin-top y background-image

## Margin, Border, padding



## Práctica 6 css

1. Crear un archivo html
2. Colocar un contenedor div con fondo naranja
3. Crear un div dentro del div anterior
   1. Crear un párrafo dentro del div
   2. Modificar las propiedades del margin, border y padding.

## Práctica 7 css

1. Crear un archivo html
2. Verificar las siguientes propiedades
   1. text-decoration
      * underline
      * overline
      * line-through

# Selectores

## Etiqueta HTML

* Text

  Description automatically generated Graphical user interface

  Description automatically generated

## ID

* Selección de un elemento específico mediante su “id”
* Para escribir el estilo de un id especifico se utiliza el carácter “#”
* Graphical user interface, text

  Description automatically generated Graphical user interface, text, application

  Description automatically generated

## Clase

* Para escribir la clase se utiliza el “.” y para realizar la selección se realiza utilizando la palabra class
* Text

  Description automatically generated Text

  Description automatically generated with medium confidence

## Práctica 8 css

1. Crear un archivo html
2. Verificar el funcionamiento de los selectores Etiqueta, ID y clase.

# CSS, selector “>”

Indica el “hijo” específico de la etiqueta que lo precede.

Por ejemplo:

Text

Description automatically generated Text

Description automatically generated

## Práctica 9 css

1. En el mismo archivo html
2. Verificar el selector “>”

# CSS Flexbox – Flex container

<https://developer.mozilla.org/es/docs/Web/CSS/CSS_Flexible_Box_Layout/Basic_Concepts_of_Flexbox>

Fue diseñado como un modelo unidimensional de layout, y como un método que pueda ayudar a distribuir el espacio entre los ítems de una interfaz y mejorar las capacidades de alineación.

Unidimensional -> Flexbox maneja el layout en una sola dimensión a la vez, ya sea como **fila** o como **columna**.

Antes del diseño “Flexbox” existían:

* Block, para secciones en una página
* Inline, para texto
* Table, para tablas de 2 dimensiones
* Positioned, para posición explicita

Flexbox hace uso de 2 ejes, el **eje principal** esta definido por flex-direction, y el eje cruzado es perpendicular a este.

**El eje principal** está definido por flex-direction, que posee cuatro posibles valores:

* row
* row-reverse
* column
* column-reverse

Si elegimos row o row-reverse, el eje principal correrá a lo largo de la fila según la dirección de la línea .

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Al elegir column o column-reverse el eje principal correrá desde el borde superior de la página hasta el final — según la **dirección del bloque**.

Gráfico, Gráfico de barras

Descripción generada automáticamente

El eje cruzado va perpendicular al eje principal, y por lo tanto si flex-direction (del eje principal) es row o row-reverse el eje cruzado irá por las columnas.

Gráfico, Gráfico de barras

Descripción generada automáticamente

Entender cuál eje es cuál es importante cuando empezamos a mirar la alineación y justificación flexible de los ítems; flexbox posee propiedades que permiten alinear y justificar el contenido sobre un eje o el otro.

Pensar en inicial y final en vez de izquierda y derecha se hará natural, y será útil cuando interactúe con otros métodos de layout tales como el CSS Grid Layout que sigue los mismos patrones.

### [El contenedor flex](https://developer.mozilla.org/es/docs/Web/CSS/CSS_Flexible_Box_Layout/Basic_Concepts_of_Flexbox#el_contenedor_flex)

Un área del documento que contiene un flexbox es llamada **contendedor flex**. Para crear un contenedor flex, establecemos la propiedad del área del contenedor **[display](https://developer.mozilla.org/es/docs/Web/CSS/display)** como flex o inline-flex. Tan pronto como hacemos esto, **los hijos directos** de este contenedor se vuelven **ítems flex**.

Como con todas las propiedades de CSS, se definen algunos valores iniciales, así que cuando creemos un contenedor flex todos los ítems flex contenidos se comportarán de la siguiente manera.

* Los ítems se despliegan sobre una fila (la propiedad flex-direction por defecto es row).
* Los ítems empiezan desde el margen inicial sobre el eje principal.
* Los ítems no se ajustan en la dimensión principal, pero se pueden contraer.
* Los ítems se ajustarán para llenar el tamaño del eje cruzado.
* La propiedad [flex-basis](https://developer.mozilla.org/es/docs/Web/CSS/flex-basis) es definida como auto.
* La propiedad [flex-wrap](https://developer.mozilla.org/es/docs/Web/CSS/flex-wrap) es definida como nowrap.

El resultado es que todos los ítems se alinearán en una solo fila, usando el tamaño del contenedor como su tamaño en el eje principal. Si hay más ítems de los que caben en el contenedor, estos no pasarán más abajo si no que sobrepasarán el margen.

Si hay ítems más altos que otros, todos los ítems serán ajustados en el eje cruzado para alcanzar al mayor.

## Práctica 10

Texto

Descripción generada automáticamente Texto

Descripción generada automáticamente con confianza media Tabla

Descripción generada automáticamente

## Práctica 11

1. Cambiar la propiedad “flex-direction” a row-reverse
   1. Interfaz de usuario gráfica, Texto

      Descripción generada automáticamente Tabla

      Descripción generada automáticamente
2. Cambiar la propiedad “flex-direction” a columna
   1. Imagen que contiene Tabla

      Descripción generada automáticamente

## Práctica 12

1. Contenedores multilínea
2. Cambiar la propiedad flex-wrap
   1. Texto

      Descripción generada automáticamente Imagen que contiene Texto

      Descripción generada automáticamente
3. Verificar abreviatura flex-flow
   1. Texto

      Descripción generada automáticamente

## Propiedades aplicadas a los ítems

* flex-grow
* flex-shrink
* flex-basis

### Espacio disponible

Cuando se cambia el valor de una de las propiedades de los ítems, se cambia la forma en la que se distribuye el espacio disponible entre los ítems.

Por ejemplo, si se tienen 3 items de 100 pixeles de ancho en un contenedor de 500px de ancho, entonces, se requieren 300px para los ítems. Esto deja 200px disponibles. Si no se cambian los valores iniciales, entonces flexbox colocará ese espacio después del último item.

Gráfico, Gráfico en cascada

Descripción generada automáticamente

## Flex-basis

Se define el tamaño de un ítem en términos del espacio que deja como espacio disponible. El valor inicial de esta propiedad es auto — en este caso el navegador revisa si los ítems definen un tamaño. En el ejemplo de arriba, ***todos los ítems tienen un ancho de 100 pixeles así que este es usado como flex-basis.***

Si los ítems no tienen un tamaño entonces el tamaño de su contenido es usado como flex-basis. Y por eso, apenas declarado display: flex en el padre a fin de crear ítems flex, todos estos ítems se ubicaron en una sola fila y tomaron ***solo el espacio necesario para desplegar su contenido***.

## Flex-grow

Con la propiedad flex-grow definida como un entero positivo, los ítems flex pueden crecer en el eje principal a partir de flex-basis. Esto hará que el ítem se ajuste y tome todo el espacio disponible del eje, o una proporción del espacio disponible si otro ítem también puede crecer.

Si le damos a todos los ítems del ejemplo anterior ***un valor flex-grow de 1 entonces*** el espacio disponible en el contenedor **flex será compartido igualitariamente** entre estos ítems y se ajustarán para llenar el contenedor sobre el eje principal.

Podemos usar flex-grow apropiadamente para distribuir el espacio en proporciones. Si otorgamos al primer ítem un valor flex-grow de 2 y a los otros un valor de 1, entonces 2 partes serán dadas al primer ítem (100px de 200px en el caso del ejemplo de arriba) y 1 parte para cada uno de los restantes (cada uno con 50px de los 200px en total).

## Práctica 13

Verificar la propiedad flex-grow

Texto

Descripción generada automáticamente Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

## flex-shrink

Así como la propiedad flex-grow se encarga de añadir espacio sobre el eje principal, la propiedad flex-shrink controla como se contrae. Si no contamos con suficiente espacio en el contenedor para colocar los ítems y flex-shrink posee un valor entero positivo, el ítem puede contraerse a partir de flex-basis. Así como podemos asignar diferentes valores de flex-grow con el fin que un ítem se expanda más rápido que otros — un ítem con un valor más alto de flex-shrink se contraerá más rápido que sus hermanos que poseen valores menores.

## Práctica 14

Verificar la propiedad flex-grow y flex-shrink

Texto

Descripción generada automáticamente Patrón de fondo

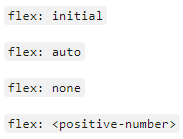
Descripción generada automáticamente con confianza media

## Abreviaciones.

Recuerde que el primer valor es flex-grow. Dándole un valor positivo significa que el ítem puede crecer. El segundo es flex-shrink — con un valor positivo los ítems pueden contraerse. El valor final es flex-basis; este es el valor que los ítems usan como valor base para crecer y contraerse.

Texto, Carta

Descripción generada automáticamente Texto

Descripción generada automáticamente 

## align-items

La propiedad [align-items](https://developer.mozilla.org/es/docs/Web/CSS/align-items) alineará los ítems sobre el eje cruzado. El valor inicial para esta propiedad es stretch razón por la cual los ítems se ajustan por defecto a la altura de aquel más alto. En efecto se ajustan para llenar el contenedor flex — el ítem más alto define la altura de este.

En cambio definimos align-items como flex-start para que los ítems se alineen al comienzo del contenedor flex, flex-end para alinearlos al final, o center para alinearlos al centro. Intente esto en el ejemplo en vivo — He definido en el contenedor flex una altura para que se aprecie que se pueden mover libremente dentro del contenedor. Vea lo que sucede si se coloca el valor align-items como:

* stretch
* flex-start
* flex-end
* center

## Práctica 15

Verificar la propiedad align-items

Texto

Descripción generada automáticamente Tabla

Descripción generada automáticamente

\* Verifique los valores flex-end, flex-center

## [justify-content](https://developer.mozilla.org/es/docs/Web/CSS/CSS_Flexible_Box_Layout/Basic_Concepts_of_Flexbox#justify-content)

La propiedad [justify-content](https://developer.mozilla.org/es/docs/Web/CSS/justify-content) es usada para alinear los ítems en el eje principal, cuyo flex-direction define la dirección del flujo. El valor inicial es flex-start que alineará los ítems al inicio del margen del contenedor, pero también se podría definir como flex-end para alinearlos al final, o center para alinearlos al centro.

También podemos usar space-between para tomar todo el espacio sobrante después de que los ítems hayan sido colocados, y distribuir de forma pareja los ítems para que haya un espacio equitativo entre cada ítem. O bien, usamos el valor space-around para crear un espacio equitativo a la derecha e izquierda de cada ítem.

## Práctica 16

Verifique los siguientes valores dejustify-content:

* space-evenly
* flex-start
* flex-end
* center
* space-around
* space-between

Texto

Descripción generada automáticamente Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Verificar

Text

Description automatically generated Text

Description automatically generated

## Propiedades

* flex-direction
  + Graphical user interface, text, application

    Description automatically generated
* flex-wrap
  + Indica si los elementos deben “ajustarse” o no.
  + Graphical user interface, text, application, chat or text message

    Description automatically generated A picture containing diagram

    Description automatically generated
* flex-flow
  + Escribe en una sola línea las dos propiedades (flex-direction y flex-wrap)
  + 
* justify-content
  + Text

    Description automatically generated
  + Text

    Description automatically generated
  + 
* align-items
  + Alinea los elementos “flex”
  + Graphical user interface, text, application

    Description automatically generated
  + Text

    Description automatically generated Text

    Description automatically generated with medium confidence
* align-content
  + Graphical user interface, text, application, chat or text message

    Description automatically generated
  + Text

    Description automatically generated Application

    Description automatically generated with medium confidence

# Flex items

## Propiedades

* order
  + Text

    Description automatically generated Icon

    Description automatically generated
* flex-grow
  + Text

    Description automatically generated A picture containing rectangle

    Description automatically generated
* flex-shrink
  + Text

    Description automatically generated with medium confidence Icon

    Description automatically generated
* flex-basis
  + Text

    Description automatically generated Shape, rectangle

    Description automatically generated
* flex
  + atajo para flex-grow, flex-shirk y flex-basis

Text

Description automatically generated with low confidence

* + Graphical user interface, text, application, chat or text message

    Description automatically generated
* align-self
  + Text

    Description automatically generated Graphical user interface

    Description automatically generated

# @media

Asocia un grupo de declaraciones anidadas, en un bloque CSS delimitado por llaves, con una condición definida por un “media query”

Ver:

<https://developer.mozilla.org/es/docs/Web/CSS/@media>

# Flex Responsivo

Ejemplo:

Chart, funnel chart

Description automatically generated

Diagram

Description automatically generated with low confidence

.flex-container {

display: flex;

flex-direction: row;

font-size: 30px;

text-align: center;

}

.flex-item-left {

background-color: #f1f1f1;

padding: 10px;

flex: 50%;

}

.flex-item-right {

background-color: dodgerblue;

padding: 10px;

flex: 50%;

}

/\* Responsive layout - makes a one

column-layout instead of two-column layout \*/

@media (max-width: 800px) {

.flex-container {

flex-direction: column;

}

}

<div class="flex-container">

<div class="flex-item-left">1</div>

<div class="flex-item-right">2</div>

</div>

## Ver ejemplo

A picture containing text, different, various, same

Description automatically generated A picture containing text, outdoor, different

Description automatically generated

<https://www.w3schools.com/css/tryit.asp?filename=trycss3_flexbox_image_gallery>

## Ver otro ejemplo.

<https://www.w3schools.com/css/tryit.asp?filename=trycss3_flexbox_website2>