

RESUMEN CSS

Por Javier Peiro Aguado (2DAW)

1- ¿Qué son las hojas de estilo CSS?

Las hojas de estilo en cascada (CSS) son una tecnología que permite controlar la apariencia de la página web, complementando el HTML. HTML define la estructura del contenido y CSS se encarga de su presentación.

Esto permite configurar estilos como colores, fuentes, tamaños, espaciados y la posición de los elementos en la página.

CSS está estandarizado por el W3C,.

Las hojas de estilo pueden aplicarse en el HTML, dentro del encabezado de la página o en archivos externos con extensión *.css.

2-Definición de estilos a nivel de elemento HTML

A pesar de que definir estilos en el atributo style de un elemento HTML es fácil, es poco práctico a la larga. Esta técnica se usa para pruebas rápidas, aplicando estilos unos pocos elementos.

Ej:

```
<h1 style="color:red; background-color:yellow;">  
  Texto rojo con fondo amarillo  
</h1>
```

El problema es, que si necesitas el mismo estilo en varios elementos, tendrás que repetirlo en cada uno... Es mejor usar hojas de estilo externas para simplificar y organizar el diseño.

3-Definición de estilos a nivel de página

La definición de estilos a nivel de página se hace en la cabecera del HTML, dentro de etiquetas <style>. Aquí se pueden agrupar los estilos para varios elementos de la página, evitando repetirlos.

Ej:

```
<head>
  <style>
    h1 {
      color: red;
      background-color: yellow;
    }
  </style>
</head>
```

4-Propiedades relacionadas a fuentes

Las propiedades de las fuentes en CSS son:

- **font-family**: Define la fuente.
- **font-size**: Tamaño de la fuente.
- **font-style**: Estilo.
- **font-weight**: Grosor.
- **font-variant**: Variantes (por ejemplo *small-caps* o *normal*).

Ej:

```
h1 {
  font-family: times new roman;
  font-size: 30px;
  font-style: italic;
  font-weight: bold;
}
```

```
h2 {
  font-family: verdana;
  font-size: 20px;
}
```

Si no se definir alguna propiedad, el navegador asigna los valores por defecto.

5-Agrupación de varios elementos HTML con una misma regla de estilo

Se puede aplicar la misma regla de estilo a varios elementos HTML, se pueden agrupar separados por comas. Esto evita duplicar las reglas.

Ej:

```
h1, h2, h3 {  
  font-family: verdana;  
  color: #0000ff;  
}
```

h1, h2 y h3 compartirán la misma fuente y color.

6-Definicion de varias reglas para un mismo elemento HTML

En CSS, se pueden definir varias reglas para un elemento HTML separando propiedades.

Ej:

```
h1, h2, h3, h4, h5, h6 {  
  font-family: Verdana;  
}
```

```
h1 {  
  font-size: 40px;  
}  
h2 {  
  font-size: 30px;  
}
```

7-Propiedades relaciunadas al texto (letter-spacing, word-spacing, text-spacing, text-indent, text-transform)

Las propiedades de los textos en CSS son:

- **color:** Es el color del texto (ejemplo: color: #ff0000; para rojo o color: rgb(255,0,0);).
- **text-align:** Alinea el texto (valores: left, right, center, justify).
- **text-decoration:** Añade efectos como subrayado o tachado (valores: none, underline, overline, line-through).

8-Más propiedades relacionadas al texto(letter-spacing, word-spacing, text-indent, text-transform)

Otras propiedades son:

- **letter-spacing**: Ajusta el espacio entre caracteres.
- **word-spacing**: Ajusta el espacio entre palabras.
- **text-indent**: Indenta la primera línea de un texto (puede ser negativo para moverlo a la izquierda).
- **text-transform**: Modifica el formato del texto con valores:
 - capitalize: Mayúsculas en la primera letra de cada palabra.
 - lowercase: Todo a minúsculas.
 - uppercase: Todo a mayúsculas.
 - none: Sin cambios.

9-Herencia de propiedades de estilo

Los elementos HTML forman un árbol, con el elemento body en la raíz. De ahí se bifurca en otros elementos como h1, h2, p, div, y estos pueden contener a su vez otros como em, strong, pre, etc.

Muchos estilos se heredan. Por ejemplo, si definimos un color para el h1, el texto dentro de un em dentro de el heredaré ese mismo color. Esto significa que los elementos anidados pueden tomar algunas propiedades de los elementos contenedores.

10-Definición de un estilo en función del contexto

Las hojas de estilo en cascada (CSS) permiten definir un estilo para un elemento HTML solamente cuando está contenido dentro de otro elemento específico. O sea que el estilo solo se aplica si se cumple una determinada condición de contexto a la que nos referimos.

11-Definición de hojas de estilo en un archivo externo

La forma recomendada de definir estilos es a través de un archivo externo con extensión .css. Este archivo contiene las reglas de estilo separadas del archivo HTML.

- Puedes aplicar los mismos estilos a varias páginas del sitio.

- Facilita hacer cambios en el diseño, ya que solo hay que modificar el archivo CSS.
- Mejora la organización del código al separar HTML y CSS.
- Optimiza la carga, ya que el archivo CSS ya que solo se carga 1 vez para todas las páginas.

Ej:

```
<link rel="StyleSheet" href="estilos.css" type="text/css">
```

12- Definición de estilos por medio de clases

Para aplicar un estilo a varios elementos HTML, podemos usar una clase. Una clase se definen con un nombre seguido de un punto, como .resaltar, y se asignan las reglas de estilo dentro de llaves:

```
.resaltar {  
  color: #000000;  
  background-color: #ffff00;  
  font-style: italic;  
}
```

Luego, para aplicar la clase, se usa el atributo class:

```
<h1 class="resaltar">Este elemento h1 aparece con la clase resaltado</h1>
```

13-Definición de estilos por medio de id.

Para definir un estilo con un id, se utiliza el símbolo (#) seguido del nombre del id. A diferencia de las clases, el id solo se aplica a un único elemento dentro de la página.

Ej:

```
#cabecera {  
  font-family: Arial;  
  font-size: 30px;  
  text-align: center;  
  color: #0000ff;  
}
```

Luego, en el HTML...

```
<div id="cabecera">Cabezon</div>
```

14-Propiedades relacionadas al borde de un elemento HTML (border-width, border-style, border-color)

Cada elemento en HTML crea una caja, y podemos acceder a sus bordes a través de las siguientes propiedades CSS:

- **border-width:** Define el grosor del borde.
- **border-style:** Establece el estilo del borde (por ejemplo, sólido, punteado, etc.).
- **border-color:** Especifica el color del borde.

Estas propiedades nos permiten personalizar los bordes de cualquier elemento HTML.

15- Más propiedades relacionadas al borde de un elemento HTML

CSS deja personalizar cada uno de los cuatro bordes de un elemento HTML de manera independiente mediante las siguientes propiedades:

- **border-top-width, border-right-width, border-bottom-width, border-left-width:** Definen el grosor de cada borde.
- **border-top-style, border-right-style, border-bottom-style, border-left-style:** Establecen el estilo de cada borde (sólido, punteado, etc.).
- **border-top-color, border-right-color, border-bottom-color, border-left-color:** Definen el color de cada borde.

16-Propiedades relacionadas al padding de un elemento HTML

El padding crea espacio entre el contenido de un elemento HTML y su borde.

- **padding:** Si todos los lados tienen el mismo valor, esta propiedad se ajusta a todos.
- **padding-top, padding-right, padding-bottom, padding-left:** Permiten configurar el espacio para cada lado del elemento.

17- Propiedades relacionadas al margen de un elemento HTML

Las propiedades del **margen** de este elemento HTML son:

- **margin:** Establece el margen para todos los lados del elemento.
- **margin-top, margin-right, margin-bottom, margin-left:** Permiten ajustar el margen de manera independiente para cada lado.

El margen se encuentra fuera del borde y separa los elementos HTML entre sí dentro de la página.

18- Propiedades relacionadas a listas

Las propiedades para personalizar listas son:

- **list-style-type:** Define el estilo del marcador (punto, número, letra, etc.). Ejemplos: disc, circle, square, decimal, lower-alpha, etc.
- **list-style-position:** Establece la posición del marcador, puede ser inside (dentro del contorno) o outside (fuera del contorno).
- **list-style-image:** Permite usar una imagen como marcador, con valores none o url.

19-Propiedades relacionadas al fondo (background)

Las propiedades CSS del fondo de un elemento HTML son:

- **background-color:** El color de fondo.
- **background-image:** Permite colocar una imagen de fondo.
- **background-repeat:** Define si la imagen de fondo se repite (valores: repeat, no-repeat, repeat-x, repeat-y).
- **background-position:** Ajusta la posición de la imagen de fondo.
- **background-attachment:** Define como se comporta de la imagen de fondo cuando hacemos scroll (valores: scroll, fixed).

20-Propiedades relacionadas a fuentes (FORMATO RESUMIDO)

La propiedad font nos permite agrupar varias propiedades relacionadas con las fuentes en una sola. Su sintaxis incluye: font-weight, font-style, font-variant, seguido de los obligatorios font-size y font-family.

21-Propiedades relacionadas al border (FORMATO RESUMIDO)

Con border se configuran todos los bordes a la vez:

- border-width
- border-style
- border-color

Ej:

```
h1 {  
  border: 1px solid #ff0000;  
}
```

Se pueden configurar los bordes por separado:

- border-top
- border-right
- border-bottom
- border-left

Indicando: width, style, y color.

22-Propiedades relacionadas al padding (FORMATO RESUMIDO)

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

Ej:

padding: 5px 10px 12px 3px;

Ahora con solo un valor, el mismo padding para todo:

padding: 1px;

Dos valores, el primero es arriba y abajo, el segundo los lados:

padding: 5px 10px;

23-Propiedades relacionadas al margin (FORMATO RESUMIDO)

Ej:

```
p {  
  margin: 5px 2px 4px 10px;  
}
```

Un solo valor, aplica el mismo margen a los cuatro lados:

```
margin: 1px;
```

Dos valores, El primero es para arriba y abajo, el segundo para los lados:

```
margin: 5px 10px;
```

24-Propiedades relacionadas al fondo (background)(FORMATO RESUMIDO)

Atributos background:

- background-color
- background-image
- background-repeat
- background-attachment
- background-position

El orden es indistinto y se pueden incluir solo las necesarias.

Ej:

```
p {  
  background: #cccccc url(fondo.jpg) repeat;  
}
```

25-Selector universal *

Aplica propiedades a todos los elementos HTML. Se utiliza el carácter *.

Ej:

```
* {  
  margin: 0;
```

```
padding: 0;  
color: #ff0000;  
}
```

Esto elimina márgenes y rellenos predeterminados de todos los elementos y establece el color del texto en rojo.

26-Pseudoclases

Son clases especiales que representan estados de un elemento HTML. Se usa un colon (:) para separar el elemento de la pseudoclase.

```
a:pseudoclase {  
  propiedad: valor;  
}
```

Pseudoclases comunes para <a>:

1. **link:** Enlace sin visitar.
2. **visited:** Enlace ya visitado.
3. **hover:** Enlace cuando el cursor está encima.
4. **active:** Enlace con el foco activo (ej. usando Tab).

El orden en que definimos las pseudoclases es fundamental para su funcionamiento:

link → visited → hover → active.

27-Eliminar el subrayado a un enlace “a” por medio de las pseudoclases

Quitamos el subrayado de los enlaces para mejorar la estética.

Ej:

```
a:link {  
  text-decoration: none;  
}
```

```
a:visited {  
  text-decoration: none;  
}
```

28-Creación de un menú vertical configurando las pseudoclasas

Creamos un menú fácil con cambiando el estilo al pasar el cursor sobre los enlaces.

```
<div class="menu">
  <p><a href="#link1">Enlace 1</a></p>
  <p><a href="#link2">Enlace 2</a></p>
  <p><a href="#link3">Enlace 3</a></p>
</div>
```

29-Creación de un menú horizontal con una lista

Un menú común es el horizontal, donde las opciones están alineadas una al lado de otra. Utilizamos una lista no ordenada con elementos de lista que contienen enlaces. Para que los elementos se ubiquen horizontalmente, se usa la propiedad float: left.

HTML:

```
<ul class="menu-horizontal">
  <li><a href="#opcion1">Opción 1</a></li>
  <li><a href="#opcion2">Opción 2</a></li>
  <li><a href="#opcion3">Opción 3</a></li>
</ul>
```

CSS:

```
.menu-horizontal {
  list-style: none; /* Quitar viñetas */
  padding: 0;
  margin: 0;
}

.menu-horizontal li {
  float: left; /* Colocar los ítems uno al lado del otro */
  margin-right: 15px; /* Separación entre ítems */
}

.menu-horizontal a {
  text-decoration: none; /* Quitar subrayado */
  padding: 10px 15px; /* Espaciado interno */
  color: black;
```

```
background-color: #f0f0f0;
display: block;      /* Asegurar el área clicable */
}

.menu-horizontal a:hover {
background-color: #007BFF; /* Cambia el fondo */
color: white;           /* Cambia el color del texto */
}
```

30-Propiedades relacionadas a la dimensión de un objeto en la página

“width” fija el ancho de un elemento y “height” fija el alto de un elemento. Si no se establece un valor, el navegador usa auto y calcula el tamaño en base al contenido y el espacio disponible.

Otras propiedades:

- min-width: Define el ancho mínimo.
- max-width: Define el ancho máximo.
- min-height: Define la altura mínima.
- max-height: Define la altura máxima.

31- unidades de medida (px,rem,em,cm,mm,etc)

Existen varias unidades de medida para definir tamaños en CSS, cada una adecuada para diferentes situaciones:

- px (píxeles): La unidad más común para definir tamaños.
- in (pulgadas), cm (centímetros), mm (milímetros): Unidades físicas.
- pt (puntos): 1 punto = 1/72 pulgadas.
- pc (picas): 1 pica = 12 puntos.
- rem (root em): Tamaño basado en la fuente del elemento raíz del documento.
- em (em): Basada en la altura de la fuente del elemento actual.
- ex (ex): Basada en la altura de la letra "x" minúscula.
- % (porcentaje): Relativo al tamaño del elemento contenedor.

32-Formas para indicar el color.

-Hexadecimal:

background-color: #ff0000;

-rgb:

Utiliza valores decimales entre 0 y 255 para rojo, verde y azul:

background-color: rgb(255, 0, 0);

-rgb con porcentaje:

background-color: rgb(100%, 0%, 0%);

-Hexadecimal abreviada:

background-color: #faf;

33-Definir un cursor para un elemento HTML

La propiedad cursor permite cambiar el cursor que se muestra cuando el ratón está sobre el elemento HTML. Algunos valores son:

- crosshair
- default
- pointer (generalmente usado para enlaces)
- move
- text
- wait
- help
- auto
- e-resize
- ne-resize
- nw-resize
- n-resize
- se-resize
- sw-resize
- s-resize
- w-resize

34-Aplicación de hojas de estilo para un formulario.

Al formulario se le pueden aplicar hojas de estilo también.

Ej:

```
#contenedorform {  
width:500px;  
margin-left:20px;  
margin-top:10px;  
background-color:#ffe;  
border:1px solid #CCC;  
padding:10px 0 10px 0;  
}  
#contenedorform form label {  
width:120px;  
float:left;  
font-family:verdana;  
font-size:14px;  
}  
.botonsubmit {  
color:#f00;  
background-color:#bbb;  
border: 1px solid #fff;  
}
```

35-Definiendo reglas de estilo a una tabla.

Las tablas son elementos comunes en los sitios web, y se pueden aplicar varias propiedades a la tabla, filas y celdas. La propiedad border-collapse del elemento <table> puede tomar dos valores:

- collapse: Une las líneas de los bordes de las celdas, mejorando la estética en muchos casos.
- separate: Mantiene los bordes separados.

36-Posicionamiento relativo (position:relative)

La propiedad position define la ubicación de un elemento HTML. Por defecto, tiene el valor static..

Cuando se usa position: relative, se puede mover el elemento respecto a su posición original usando las propiedades left y top, con valores positivos o negativos. Sigue ocupando su espacio original, dejando un área vacía donde estaba

37-Posicionamiento absoluto (position:absolute)

Con position: absolute, un elemento se coloca fuera del flujo normal de la página, sin reservar espacio para él. La posición se calcula respecto a la página, y se utiliza junto con las propiedades top y left para ubicarlo. Al contrario que el posicionamiento relativo, no deja espacio vacío en el lugar donde debería estar.

38-Posicionamiento absoluto y propiedad z-index

Cuando varios elementos con position: absolute se superponen, utilizamos la propiedad z-index para definir cuál se muestra encima de los demás. El valor más alto de z-index tiene mayor prioridad para ser visible. El navegador renderiza los elementos en orden, de menor a mayor valor de z-index.

39-Posicionamiento fijo (position:fixed)

osition: fixed; Coloca un elemento en una posición fija con respecto a la ventana del navegador, no a la página. A diferencia del posicionamiento absoluto, el elemento permanece en el mismo lugar incluso al hacer scroll en la página.

40-Disposición de 2 columnas (position:absolute)

Para crear una página con dos columnas sin usar tablas HTML, podemos utilizar position: absolute; para posicionar los elementos. Esto permite organizar el contenido de forma más flexible y es compatible con navegadores antiguos.

41-Propiedad float aplicada a una imagen.

La propiedad float saca un elemento del flujo normal de la página. Con los valores left, right o none, podemos hacer que el texto rodee una imagen (), posicionándola a la izquierda o derecha del contenido.

42-Propiedad float aplicada a otros elementos HTML

La propiedad float permite colocar cualquier elemento HTML a la izquierda o derecha, haciendo que otros elementos fluyan alrededor de él. Al usar float, se debe establecer el width para definir el ancho. El comportamiento de la altura del elemento flotante afecta la disposición de los demás elementos, igual que el texto que lo rodea o un div.

43-Propiedad clear

La propiedad clear se usa junto a elementos flotantes (float) para evitar que otros elementos se alineen junto a los flotantes. Puede especificar que el elemento no flote a la izquierda, derecha o en ambos lados.

- both
- left
- right
- none

44- Disposición de 2 columnas (propiedad float)

Para crear una página con dos columnas usando float, se flotan dos div. El primer div se coloca a la izquierda con un ancho fijo, y el segundo se ajusta a la derecha con un margin-left un poco mas largo para evitar que se envuelva alrededor del primero si es más largo.

45- Disposición de 2 columnas (cabecera y pie)

Una estructura común es tener una cabecera, dos columnas y un pie de página. Para lograr un diseño fluido, donde el tamaño del contenido se adapte al tamaño de la ventana del navegador, se utiliza width: 100%. Se hace similar con la segunda columna con margin en vez de float.

46- Disposición de 3 columnas (cabecera y pie)

Para añadir una tercera columna flotante a la derecha, los div de las columnas 1 y 3 deben aparecer primero en el HTML, ya que son las que se flotan. La columna 2, que es la central, tirnr que ir al final.

47- Diseño de ancho fijo

En un diseño de ancho fijo, los elementos HTML no cambia cuando el tamaño de la ventana varía. Si el contenido no cabe, aparecerá una barra de desplazamiento. Las medidas se indican en píxeles.

48- Diseño de ancho líquido

En un diseño de ancho líquido, los elementos se ajustan automáticamente al tamaño de la ventana expandiéndose o contrayéndose. Se utilizan porcentajes para definir el ancho de los div, haciendo que el diseño se adapte al espacio disponible.

49- Selector de hijos

Los selectores permiten aplicar estilos a elementos más específicos. Un selector de tipo de elemento aplica un estilo los elementos de ese tipo.

Ej:

```
h1 { font-family: Arial; }.
```

También puede combinarse con clases:

```
h1.principal { font-family: Arial; }
```

50- Selector de hermano adyacente y hermano general.

Selecciona un elemento que sigue inmediatamente a otro con el mismo padre. Por ejemplo, en el código:

```
<h1>Título 1</h1>
<p>Esto es el primer párrafo.</p>
<p>Esto es el segundo párrafo.</p>
```

`h1 + p { color: #f00; }` aplica el color rojo al segundo párrafo.

51-Selector de atributo

El selector de atributo permite seleccionar elementos HTML basándose en los atributos. Ej:

```
<a href="http://www.google.com.ar" target="_blank">Titulo Principal</a>
```

52-Pseudo-clases: first-child y last-child

Las pseudo-clases first-child y last-child permiten seleccionar el primer y el

último hijo del elemento padre. Se usan para aplicar estilos a esos elementos dentro del contenedor.

53-Pseudo-clases: nth-child y nth-last-child

Las pseudo-clases nth-child y nth-last-child seleccionan elementos según la posición en el contenedor. Al usar odd (impar) y even (par), aplicamos estilos a elementos en posiciones impares o pares.

Ej: `tr:nth-child(odd) { background-color: red; }`

54-Pseudo-clases: nth-of-type, nth-last-of-type, first-of-type y last-of type

nth-of-type selecciona el enésimo elemento, mientras que nth-child cuenta todos los elementos, sin importar su tipo. Las otras seleccionan el primer, último o enésimo desde el final del tipo de elemento.

55-Pseudo-clases: only-child y only of type

only-child se aplica cuando un elemento es el único hijo de su padre.

Ej:

```
<ul id="fechaInicio"> <li>10/10/2015</li> </ul>
```

56-Pseudo-clases: empty

La pseudo-clase empty se aplica a un elemento HTML que no tenga ni hijos ni texto.

Ej:

```
<div></div>
```

```
div:empty{
```

```
...
```

```
}
```

57-Pseudo-elementos: first-letter y first-line

Estos pseudo-elementos afectan solo una parte de un elemento HTML. El pseudo-elemento first-letter selecciona la primera letra de un elemento,

permitiendo aplicar un estilo específico. Por ejemplo, para hacer que la primera letra de un párrafo sea más grande y de color rojo, se usa:

```
p::first-letter {  
  font-size: 1.5rem;  
  color: red;  
}
```

58-Pseudo-elementos: before y after

Los pseudo-elementos before y after se utilizan para insertar contenido antes o después del contenido del propio elemento. Se debe usar la propiedad content para definir el contenido que se agrega.

Ej:

```
strong::before {  
  content: "¡";  
}
```

```
strong::after {  
  content: "!";  
}
```

59-CSS media queries

Las media queries son una funcionalidad de CSS que permite aplicar reglas de estilo según el dispositivo o del medio (como pantalla, impresora, TV, etc.). Se introdujeron en CSS2 y se mejoraron con CSS3.

Son útiles para crear diseños adaptables, ajustandose el estilo según el dispositivo (móvil, tablet, PC).

Por ejemplo, cambiar un div si el ancho de la pantalla es pequeño.

60-Flexbox

Flexbox facilita la maquetación de páginas web, simplifica el diseño en comparación con métodos anteriores. Es ampliamente soportado por navegadores y utilizado en frameworks como Bootstrap 4.

Permite crear diseños adaptables a diferentes tamaños de pantalla.

61-Flexbox (flex-direction aplicada al contenedor)

La propiedad flex-direction define la dirección de los elementos dentro de un contenedor flexible. Los valores que tiene son:

- row (predeterminado)
- column
- row-reverse
- column-reverse.

62-Flexbox (justify-content aplicada al contenedor)

La propiedad justify-content controla la distribución de los elementos dentro de un contenedor flex. Por defecto, el valor es flex-start. Los valores son:

- flex-start
- flex-end
- center
- space-between
- space-around.

63-Flexbox (align-items aplicada al contenedor)

align-items controla la alineación de los elementos a lo largo del eje transversal del contenedor flex. Los valores que podemos asignarle son:

- stretch
- flex-start
- flex-end
- center
- baseline

64-Flexbox (flex-wrap aplicada al contenedor)

La propiedad flex-wrap controla si los elementos se ajustan en varias líneas. Por defecto, el valor es nowrap. Los valores que podemos asignarle son:

- nowrap
- wrap
- wrap-reverse

65-Flexbox (align-content aplicada al contenedor)

La propiedad align-content la usamos cuando hay varias líneas en un Flexbox (con flex-wrap: wrap). Los valores posibles son:

- stretch
- flex-start
- flex-end
- center
- space-between
- space-around

66-Flexbox (flex-flow aplicada al contenedor)

flex-flow es un atajo para flex-direction y flex-wrap.

Ej:

flex-flow: "valor de flex-direction" "valor de flex-wrap".

67-Flexbox (flex-grow aplicada a los items)

flex-grow controla el crecimiento de un item dentro de un contenedor, rellenando el espacio vacío. Su valor por defecto es cero, y no tiene unidad de medida, ya que indica un factor de crecimiento.

68-Flexbox (flex-shrink aplicada a los items)

La propiedad flex-shrink controla el factor de encogimiento de un item cuando el espacio es justo. Su valor por defecto es 1, lo que hace que los items se encojan en proporción. Si queremos que algún item no se encoja, le asignamos 0.

69-Flexbox (flex-basis aplicada a los items)

flex-basis define el tamaño inicial de un item en el eje principal (width en contenedores horizontales, height en verticales). Su valor por defecto es auto, esto hace que el tamaño se calcule automáticamente.

70-Flexbox (flex aplicada a los items)

flex es una forma resumida de flex-grow, flex-shrink y flex-basis, en ese orden.

Ej:

flex: 1 0 60%;

71-Flexbox (order aplicada a los items)

order permite cambiar el orden de los items en el contenedor. Por defecto es 0, lo que mantiene el orden original del HTML. Podemos asignar valores enteros a partir de 0 para modificar el flujo de los items.

72-Flexbox (align-self aplicada a los items)

align-self permite alinear un item dentro de el contenedor flexible en el eje transversal. Los valores son:

- auto
- flex-start
- flex-end
- center
- stretch
- baseline