Diseño Web Responsive HTML5 y CSS3

Unidad 4: Maquetación con Flexbox

Contacto: tecnologia@elearning-total.com
Web: www.elearning-total.com

Indice

Objetivos

Que el alumno logre:

- Logre aplicar los atributos de estilos CSS a la estructura del sitio a través de la utilización de cajas.

Flexbox

Flexbox es un módulo completo de layout disponible en la especificación de CSS3. Define cómo se muestran los elementos y cómo se relacionan con el resto.

Como concepto, podemos entender Flexbox como un modelo para la creación de layouts, que pretende mejorar los anteriores, aunque sin ser excluyente. Todas las técnicas disponibles antes de Flexbox tenían diversos problemas y limitaciones que se pretenden solucionar con esta especificación de CSS3.

Flexbox es una herramienta muy avanzada para poder crear layouts de características mejoradas y necesarias en el día de hoy, donde es tan importante una estética cuidada y una gran adaptabilidad a distintos formatos de pantalla.

En la práctica, Flexbox agrega un nuevo tipo de "display CSS", con una completa gama de nuevas propiedades aplicables a ese tipo de display, a partir de los que podemos conseguir cosas extraordinarias.

En Flexbox diferenciamos dos elementos principales: **la caja contenedora y los elementos que situamos dentro**.

Al aplicar un ***display flex*** o ***display inline-flex*** hacemos que una caja se comporte mediante este nuevo estándar y eso produce que los elementos que tiene como contenido se puedan distribuir con las propiedades de este estándar de maquetación.

El contenedor va a poder modificar las dimensiones y el orden de los items, para acomodarlos de distintas maneras controladas por el desarrollador. Podremos repartir el espacio entre ellos de diversas formas, para distribuirlo a nuestro antojo, permitir que los items se estiren para ocupar todo el contenedor, o se encojan para que quepan en él sin desbordar, de colocarse distribuidos en filas o en columnas, etc.

Qué soluciona Flexbox

Flexbox permite que se puedan posicionar elementos de una manera más concisa y distribuir los espacios entre ellos de forma más flexible. Permite gran cantidad de comportamientos:

- **Alineación vertical**

- **Columnas de igual altura:** Conseguir que todos los items de un listado tengan la misma altura, independientemente de su contenido.

- **Cambiar el orden de los elementos:** Sin tener que cambiar el orden de los elementos en el HTML, con Flexbox conseguimos que se ordenen de maneras distintas al visualizarse en la página.

Con esto se consigue que los diseñadores, maquetadores y desarrolladores frontend en general tengan mucho mayor control sobre los elementos en la página y puedan, de una manera más detallada, especificar su apariencia y colocación, limitándose solamente a modificar los atributos CSS.

Flexbox y compatibilidad con navegadores

Flexbox hoy ya es una realidad, puesto que los navegadores modernos lo soportan y lo interpretan correctamente. Por tanto, está listo para usar en cualquier tipo de proyecto.

El único navegador que no lo soporta es Internet Explorer en versiones antiguas, como IE8 o IE9 (versiones obsoletas que no soportan la mayoría de los nuevos elementos de HTML5 y CSS3).

Aunque no suele ser la mejor solución, existen fallbacks o polyfills que aportan un soporte parcial a Flexbox, instalando una librería Javascript adicional. Son una opción cuando necesitamos soportar Flexbox en algunos navegadores antiguos, aunque también hay que advertir que tendrá un costo en términos de rendimiento / peso, y quizás no todo se vea exactamente igual que en navegadores con soporte nativo. Un ejemplo de polyfill lo encontramos en Flexibility.

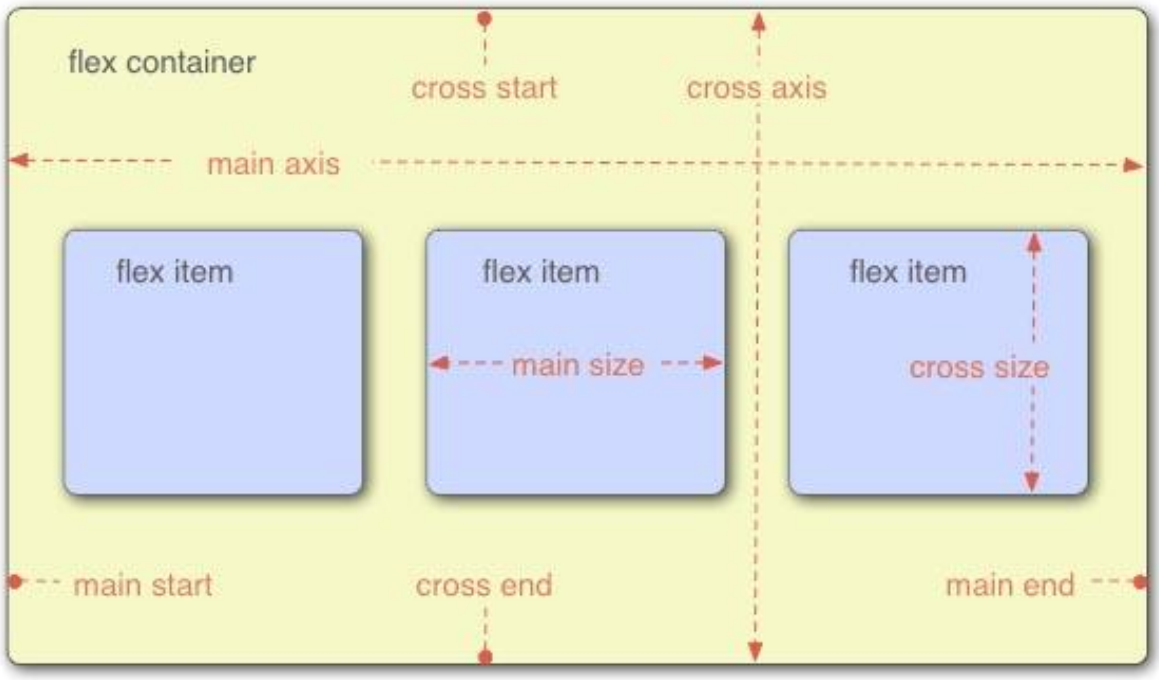
El concepto de "cajas flexibles"

Lo que caracteriza un diseño flexible es su habilidad para alterar el ancho y alto de sus elementos para ajustarse lo mejor posible al espacio disponible en cualquier dispositivo. Un contenedor flexible expande sus elementos para rellenar el espacio libre, o los comprime para evitar que rebasen el área prevista.

El algoritmo del modelo de diseño de "**cajas flexibles**" no parte de ninguna dirección predeterminada, al contrario de lo que ocurre con el modelo "**bloque**", que asume una disposición vertical de los elementos, o lo que pasa con el modelo "**en línea**", que asume una disposición horizontal. Mientras que el modelo "bloque" funciona bien para páginas, se queda muy corto cuando se trata de aplicaciones en las que hay que tener en cuenta el cambio de orientación del dispositivo o los cambios de tamaño realizados por los gestos del usuario. El modelo de "cajas flexibles" es más apropiado para diseños de pequeña escala, mientras que el modelo "rejilla" es adecuado para diseños de gran escala.

Ambos son parte del gran esfuerzo que el "CSS Working Group" está realizando para proveer de mayor interorperabilidad a las aplicaciones web con todo tipo de usuarios, distintos modos de escritura, y otras necesidades de flexibilidad.

En el siguiente diagrama se muestra un contenedor flexible que tiene una **flex-direction** de tipo **row**, es decir, que los elementos flexibles se muestran uno a continuación del otro horizontalmente a lo largo del **eje principal** (main axis) de acuerdo con el modo de escritura preestablecido, y en este caso, la dirección en que el texto de los elementos fluye es de izquierda-a-derecha.



Contenedor flexible (Flex container)

El elemento "padre" que contiene los elementos flexibles. Un contenedor flexible se define usando los valores flex o inline-flex en la propiedad ***display***.

Elemento flexible (Flex item)

Cada hijo de un contenedor flex se convierte en un elemento flexible. Si hay texto directamente incluido en el contenedor flexible, se envuelve automáticamente en un elemento flexible anónimo.

Ejes

Cada diseño de "caja flexible" sigue dos ejes. El **eje principal** es el eje a lo largo del cual los elementos flexibles se suceden unos a otros. El **eje secundario** es el eje perpendicular al eje principal.

- La propiedad ***flex-direction*** establece el eje principal.
- La propiedad ***justify-content*** define cómo los elementos flexibles se disponen a lo largo del eje principal en la línea en curso.

- La propiedad ***align-items*** define cómo los elementos flexibles se disponen a lo largo del eje secundario de la línea en curso.

- La propiedad ***align-self*** define cómo cada elemento flexible se alinea respecto al eje secundario, y sustituye al valor por defecto establecido por align-items.

Direcciones

Los lados inicio principal / fin principal (main start / main end) e inicio secundario / fin secundario (cross start/cross end) del contenedor flexible describen el origen y final del flujo de los elementos flexibles. Estos siguen los eje principal y secundario según el vector establecido por **writing-mode** (izquierda-a-derecha, derecha-a-izquierda, etc.).

- La propiedad ***order*** asigna elementos a grupos ordinales y determina qué elementos aparecen primero.

- La propiedad ***flex-flow*** property combina las propiedades ***flex-direction*** y ***flex-wrap*** para colocar los elementos flexibles.

Líneas

Los elementos flexibles pueden disponerse en una sola o varias líneas de acuerdo con la propiedad ***flex-wrap***, que controla la dirección del eje secundario y la dirección en la que las nuevas líneas se apilan.

Dimensiones

Los términos equivalentes a "altura" y "anchura" usados en los elementos flexibles son tamaño principal (main size) y tamaño secundario (cross size), que respectivamente siguen al eje principal y al eje secundario del contenedor flexible.

Las propiedades ***min-height*** y ***min-width*** tienen un nuevo valor:auto que establece el tamaño mínimo de un elemento flexible.

La propiedad ***flex*** combina las propiedades ***flex-basis***, ***flex-grow***, y ***flex-shrink*** para establecer el grado de flexibilidad de los elementos flexibles.

Consideraciones de los elementos flexibles

El texto que se encuentre directamente dentro de un contenedor flexible, será automáticamente envuelto en un elemento flexible anónimo. Sin embargo, si un elemento flexible contiene solamente espacios en blanco no será mostrado, como si tuviera la propiedad **display:none**.

Los "*hijos*" de un contenedor flexible que tengan un posicionamiento absoluto, se situarán de manera que su posición estática se determine en referencia a la esquina del inicio principal (main start) de su contenedor flexible.

Si asignamos **visibility:collapse** a un elemento flexible causamos que sea tratado como si fuera **display:none** en vez de lo que se supone que debería ocurrir, es decir, como si fuera visibility:hidden.

La alternativa mientras se resuelve este problema es usar visibility:hidden para elementos flexibles que deban comportarse como visibility:collapse.

Los márgenes de elementos flexibles adyacentes no se colapsan. Usando márgenes auto se absorbe el espacio extra vertical y horizontalmente y puede ser utilizado para alinear o separar elementos flexibles adyacentes.

Para asegurarnos un tamaño mínimo por defecto de los elementos flexibles, debemos usar min-width:auto y/o min-height:auto.

Para los elementos flexibles, el valor de atributo auto calcula la mínima anchura/altura del elemento para que no sea menor que la anchura/altura de su contenido, garantizando que el elemento es mostrado suficientemente grande como para que se vea su contenido.

Las propiedades de alineación de "cajas flexibles" realizan un "verdadero" centrado en CSS. Esto significa que los elementos flexibles permanecerán centrados, incluso si estos rebasan su contenedor flexible. Esto puede llegar a ser un problema, ya que si sobrepasan el tope superior de la página o el izquierdo (en escritura LTR de izquierda-a-derecha) o el derecho (en escritura RTL de derecha-a-izquierda), no se puede desplazar hacia ese área, incluso habiendo contenido allí.

En el futuro, las propiedades de alineación se ampliarán para que tengan una opción "safe" (seguro) para controlar esta situación. De momento, podemos usar los márgenes para conseguir el centrado. Los márgenes automáticos se adaptarán asumiendo el espacio sobrante, centrando los elementos flexibles donde sobre espacio, y cambiando a alineación normal donde no sobre espacio.

Mientras el orden en que se muestran los elementos es independiente de su orden en el código fuente, esta independencia afecta solamente a la representación visual, y no al orden de locución y navegación que seguirán el orden establecido en el código fuente. Incluso la propiedad ***order*** no afectará a la secuencia de locución ni de navegación. Así que los desarrolladores debemos preocuparnos del orden de los elementos adecuadamente en el código fuente para que no se deteriore la accesibilidad del documento.

Propiedades de las "cajas flexibles"

Propiedades que no afectan a las "cajas flexibles"

Como las "cajas flexibles" emplean un algoritmo diferente, alguna propiedades no tienen sentido para un contenedor flexible.

- Propiedades column-* del ***Módulo Multicol*** no tienen ningún efecto en un elemento flexible.

- ***float*** y ***clear*** no tienen ningún efecto en un elemento flexible. Usar float causa que la propiedad display del elemento se comporte como block.

- ***vertical-align*** no tiene efecto en la alineación de los elementos flexibles.

Ejemplo de flexbox

Este ejemplo básico muestra cómo aplicar "flexibilidad" a un elemento y como sus "hijos" se comportan flexiblemente:

HTML

```
<!DOCTYPE html>
<html>
<head>
<title> Ejemplo flexbox </title>
<meta charset="utf-8"/>
<link href="style.css" rel="stylesheet"/>
</head>
<body>
 <p>Flexbox ejemplo</p>
 <div class="flexible">
  <div>primero</div>
  <div>segundo</div>
  <div>tercero</div>
 </div>
</body>
</html>
```

Archivo CSS

```
.flexible
  {
                /* estilo básico */
    width: 350px;
    height: 200px;
    border: 1px solid #555;
    font: 14px Arial;

                /* flexbox */
    display: flex;
    flex-direction: row;
  }

  .flexible > div
  {
    flex: 1 1 auto;
    width: 30px;
    transition: width 0.7s ease-out;
  }

            /* colores */
  .flexible > div:nth-child(1){ background : #c137ce; }
  .flexible > div:nth-child(2){ background : #cec739; }
  .flexible > div:nth-child(3){ background : #3947cc; }
```

```
.flexible > div:hover
{
    width: 200px;
}
```

Propiedades de flexbox

flex

La propiedad **CSS flex** es una propiedad resumida que especifica la capacidad de un elemento flexible para alterar sus dimensiones para llenar el espacio disponible. Los elementos flexibles pueden ser estirados para utilizar el espacio disponible proporcional a su factor de crecimiento flexible o su factor de contracción flexible para evitar desbordamiento.

Valores

<'flex-grow'>

Define el *flex-grow* del elemento flexible. Ver { { Xref_cssnumber ( ) } } para obtener más detalles . Los valores negativos no se consideran válidos . El valor predeterminado es 1 cuando se omite.

<'flex-shrink'>

Define el *flex-shrink* del elemento flexible. Ver { { Xref_cssnumber ( ) } } para obtener más detalles . Los valores negativos no se consideran válidos . El valor predeterminado es 1 cuando se omite.

<'flex-basis'>

Define el *flex-basis* del elemento flexible. Se acepta cualquier valor válido para las propiedades width y height . Un tamaño preferente de 0 debe tener una unidad para evitar ser interpretado como flexible. El valor predeterminado es 0% cuando se omite.

none

Esta palabra clave se computa a 0 0 auto.

Sintaxis

```
        /* 0 0 auto */
flex: none;
```

```
        /* Un valor, número sin unidades: flex-grow */
flex: 2;

        /* Un valor, width/height: flex-basis */
flex: 10em;

flex: 30px;

flex: auto;

flex: content;

        /* Dos valores: flex-grow | flex-basis */
flex: 1 30px;

        /* Dos valores: flex-grow | flex-shrink */
flex: 2 2;

        /* Tres valores: flex-grow | flex-shrink | flex-basis */
flex: 2 2 10%;

        /* Valores globales */
flex: inherit;

flex: initial;

flex: unset;
```

flex-basis

La propiedad de CSS *flex-basis* especifica la base flexible, la cual es el tamaño inicial de un elemento flexible. Ésta propiedad determina el tamaño de una caja de contenidos a no ser que se haya especificado de otra forma usando *box-sizing.*

Valores

width

Definido por un número seguido de una unidad absoluta tal como px, mm o pt, o un porcentaje del tamaño principal de un contenedor flexible padre. Los valores negativos no son válidos.

content

Indica el dimensionamiento automático, basado en el contenido del elemento flexible.

Sintaxis

```
/* Especificar <'width'> */
flex-basis: 10em;
flex-basis: 3px;
flex-basis: auto;

/* Palabras clave de dimensionamiento intrínseco */
flex-basis: fill;
flex-basis: max-content;
flex-basis: min-content;
flex-basis: fit-content;

/* Tamaño automático basado en el contenido del elemento flexible */
flex-basis: content;

/* Global values */
flex-basis: inherit;
```

flex-basis: initial;

flex-basis: unset;

flex-direction

Especifica cómo colocar los objetos flexibles en el contenedor flexible definiendo el eje principal y la dirección (normal o invertida).

El valor de row y row-reverse se verán afectados por la direccionalidad del contenedor flexible.

Si su atributo dir es ltr, row representa el eje horizontal orientado de izquierda a derecha y, row-reverse desde la derecha hacia la izquierda;

Si el atributo dir es rtl, row representa el eje orientado de derecha a izquierda, y row-reverse de izquierda a derecha.

Valores

Se aceptan los siguientes valores:

row

El eje principal del contenedor flexible está definido para ser el mismo que la dirección del texto. Los puntos principales de inicio y final son los mismos que la dirección del contenido.

row-reverse

Se comporta igual que row pero los puntos principales de inicio y final son intercambiados.

column

El eje principal del contenedor flexible es el mismo que el eje del bloque. Los puntos principales de inicio y final son los mismos que los puntos de antes y después del modo escritura.

column-reverse

Se comporta igual que row pero los puntos principales de inicio y final son intercambiados.

Sintaxis

flex-direction: row

flex-direction: row-reverse

flex-direction: column

flex-direction: column-reverse

flex-direction: inherit

flex-flow

Es una propiedad taquinográfica para las propiedades individuales flex-direction y flex-wrap.

Sintaxis

```
/* flex-flow: <'flex-direction'> */
flex-flow: row;
flex-flow: row-reverse;
flex-flow: column;
flex-flow: column-reverse;

/* flex-flow: <'flex-wrap'> */
flex-flow: nowrap;
flex-flow: wrap;
flex-flow: wrap-reverse;

/* flex-flow: <'flex-direction'> and <'flex-wrap'> */
flex-flow: row nowrap;
flex-flow: column wrap;
flex-flow: column-reverse wrap-reverse;

/* valores globales */
flex-flow: inherit;
flex-flow: initial;
flex-flow: unset;
```

flex-grow

Especifica el factor de crecimiento de un elemento flexible. Se especifica qué cantidad de espacio debe ocupar el elemento dentro del contenedor flexible.

Sintaxis

```
/* <number> valores */
```

flex-grow: 3;

flex-grow: 0.6;

```
/* Valores globales */
```

flex-grow: inherit;

flex-grow: initial;

flex-grow: unset;

flex-shrink

Si el espacio disponible es negativo (el tamaño del contenedor es menor a la suma de los tamaños de los items), de forma predeterminada los items se encogen en proporciones iguales para caber en una sola línea (en un próximo artículo trabajaremos flexbox con varias líneas de items). Con la propiedad flex-shrink se controla cómo se encogerán los elementos.

Sintaxis

flex-shrink: 2;

flex-shrink: 0.6;

```
/* Global values */
```

flex-shrink: inherit;

flex-shrink: initial;

flex-shrink: unset;

flex-wrap

Especifica si los elementos "hijos" son obligados a permanecer en una misma línea o pueden fluir en varias líneas. Si la cobertura (wrap) está permitida, esta propiedad también te permite controlar la dirección en la cual serán apilados los elementos.

Valores

Se aceptan los siguientes valores:

nowrap

Los elementos flex son distribuidos en una sola línea, lo cual puede llevar a que se desborde el contenedor flex. El valor cross-start es equivalente a start o before según el valor de *flex-direction*.

wrap

Los elementos flex son colocados en varias líneas. El valor cross-start equivale a start o before dependiendo del valor flex-direction y cross-end implicaría lo opuesto a lo especificado por cross-start.

wrap-reverse

Actúa como wrap pero cross-start y cross-end están intercambiados.

Sintaxis

flex-wrap: nowrap

flex-wrap: wrap

flex-wrap: wrap-reverse

flex-wrap: inherit

align-content

Ajusta las líneas dentro de un contenedor flex cuando hay espacio extra en el eje transversal.

Esta propiedad no tiene efecto en cajas flexibles de una sola línea.

Valores

flex-start

Las líneas son ajustadas a partir del inicio del eje transversal. El borde transversal de inicio de la primera línea y el del contenedor flexible quedan unidos. Cada línea siguiente es unida a su predecesora.

flex-end

Las líneas son ajustadas a partir del final del eje transversal. El borde transversal final de la última línea y el del contenedor flexible quedan unidos. Cada línea que precede es unida a la línea siguiente.

center

Las líneas son ajustadas hacia el centro del contenedor flexible. Las líneas son unidas entre sí, y centradas dentro del contenedor. El espacio entre el borde transversal de inicio y la primera línea, y el que hay entre el borde transversal final y la última línea es el mismo.

space-between

Las líneas son distribuidas de manera uniforma en el contenedor flexible. El espaciado se hace de modo que la separación entre cualquier par de elementos adyacentes sea el mismo. Los bordes transversales de inicio y de fin del contenedor son unidos a los bordes de la primera y última línea, respectivamente.

space-around

Las líneas son distribuidas uniformemente de modo que el espacio entre cualquier par de elementos adyacentes sea el mismo. El espacio vacío antes de la primera línea y el espacio después de la última es igual a la mitad del espacio entre cualquier par de líneas adyacentes.

stretch

Las líneas son estiradas para usar el espacio sobrante. El espacio libre en el contenedor es dividido por igual entre todas las líneas.

Sintaxis

```
/* Ajusta las líneas desde el inicio del eje transversal */
align-content: flex-start;

/* Ajusta las líneas desde el final del eje transversal */
align-content: flex-end;

/* Ajusta las líneas al rededor del centro del eje transversal */
align-content: center;
```

```
/* Distribuye las líneas a lo largo del eje transversal, de principio a fin
*/
align-content: space-between;

/* Distribuye las líneas a lo largo del eje transversal, igualmente
espaciados */
align-content: space-around;

/* Estira las líneas para que ocupen el eje transversal completo */
align-content: stretch;

/* Valores globales */
align-content: inherit;
align-content: initial;
align-content: unset;
```

align-items

Alinea los elementos de la línea flexible actual de la misma forma que ***justify-content***, pero en dirección perpendicular.

Valores

flex-start

El límite del margen transversal inicial del elemento flexible es unido al borde transversal final de la línea.

flex-end

El límite del margen transversal final del elemento flexible es unido al borde transversal final de la línea.

center

Los márgenes del elemento flexible son centrados dentro de la línea sobre su eje transversal. Si el tamaño transvesal del elemento es mayor al del contenedor, se excederá por igual hacia ambas direcciones.

baseline

Todos los elementos flexibles son ajustados de modo que sus bases queden alienadas. El elemento con la distancia mayor entre su límite transversal inicial y su base es combinado con el borde transversal de la línea.

stretch

Los elementos flexibles son estirados de modo que el tamaño transversal de sus límites sea el mismo de la línea, manteniendo sus restricciones de anchura y altura.

Sintaxis

```
/* Alinea los elementos al borde de inicio */
align-items: flex-start;

/* Alinea los elementos al borde de fin */
align-items: flex-end;

/* Centra los elementos en el eje transversal */
align-items: center;

/* Alinea la base de los elementos */
align-items: baseline;

/* Estira los elementos para ajustarlos */
align-items: stretch;

/* Valores globales */
align-items: inherit;

align-items: initial;

align-items: unset;
```

align-self

Alinea los elementos flexibles de la línea flexible actual, reemplazando el valor de ***align-items***. Si el límite transversal de alguno de los elementos está definido como auto, el valor de align-self es ignorado.

Valores

auto

Se calcula acorde al valor de align-items del padre, o al de stretch si el elemento no tiene padre.

flex-start

El límite transversal inicial del elemento flexible es unido al borde transversal inicial de la línea.

flex-end

El límite transversal final del elemento flexible es unido al borde transversal final de la línea.

center

Los límites del elemento flexible son centrados dentro de la línea en su eje transversal. Si el tamaño transversal del elemento es superior al del contenedor, se excederá por igual hacia ambas direcciones.

baseline

Todos los elementos flexibles son ajustados de modo que sus bases estén alineadas. El elemento con la distancia mayor entre su límite transversal inicial y su base es combinado con el borde transversal de la línea.

stretch

Los elementos flexibles son estirados de modo que el tamaño transversal de sus límites sea el mismo de la línea, manteniendo sus restricciones de anchura y altura.

Sintaxis

```
/* Valores clave */
align-self: auto;
align-self: flex-start;
align-self: flex-end;
align-self: center;
align-self: baseline;
align-self: stretch;
```

```
/* Valores globales */
```

align-self: inherit;

align-self: initial;

align-self: unset;

justify-content

Define cómo el navegador distribuye el espacio entre y alrededor de los items flex, a lo largo del eje x principal de su contenedor.

El alineamiento se produce luego de que las longitudes y márgenes automáticos son aplicados, lo que significa que, si existe al menos un elemento flexible con *flex-grow* diferente a 0, no tendrá efecto ya que no habrá espacio disponible.

Valores

flex-start

Los items flex se colocan comenzando desde el comienzo principal. El margen del primer item es alineado al ras con el borde del comienzo principal de la línea y cada item siguiente es alineado al ras con el precedente.

flex-end

Los items flex se colocan comenzando desde el final principal. El margen del último item es alineado al ras con el borde del final principal de la línea y cada item precedente es alineado al ras con el siguiente.

center

Los items flex son colocados hacia el centro de la línea. Los items flex se alinean al ras entre sí y en torno al centro de la línea. El espacio entre el borde del comienzo principal de la línea y el primer item es el mismo que el espacio entre el borde del final principal y el último item de la línea.

space-between

Los items flex se distribuyen uniformemente sobre la línea. El espaciamiento se hace de tal manera que el espacio adyacente entre dos items es el mismo. El borde del comienzo principal y el borde del final principal se alinean al ras con el borde del primer y último item respectivamente.

space-around

Los items flex se alinean uniformemente de tal manera que el espacio entre dos items adyacentes es el mismo. El espacio vacío anterior al primer item y posterior al último item equivale a la mitad del espacio entre dos items adyacentes.

Sintaxis

```
/* Alinear items flex desde el comienzo */
justify-content: flex-start;

/* Alinear items desde el final */
justify-content: flex-end;

/* Alinear items en el centro */
justify-content: center;

/* Distribuir items uniformemente
El primer item al inicio, el último al final */
justify-content: space-between;

/* Distribuir items uniformemente
Los items tienen el mismo espacio a su alrededor */
justify-content: space-around;

/* Valores globales */
justify-content: inherit;
justify-content: initial;
justify-content: unset;
```

order

Especifica el orden utilizado para disponer los elementos en su contenedor flexible. Los elementos estarán dispuestos en orden ascendente según el valor de order. Los elementos con el mismo valor de order se dispondrán en el orden en el cual aparecen en el código fuente.

Valores

<integer>

Representa el grupo ordinal al que el elemento flexible ha sido asignado.

Sintaxis

```
/* Valor numérico incluyendo números negativos */
order: 5;
order: -5;

/* Valores Globales */
order: inherit;
order: initial;
order: unset;
```

Resumen

En esta Unidad…

Trabajamos con el Modelo de cajas, aplicado al a maquetación.

En la próxima Unidad…

En la próxima unidad vamos a comenzar a trabajar con las nuevas incorporaciones de CSS3.