

FLEXBOX

Se coloca la propiedad **display** en flex en el contenedor.

Existen dos ejes el main axis (eje principal) y cross axis (eje secundario).

Inline-flex se comporta como un elemento en inline.

La dirección por defecto es **row**.

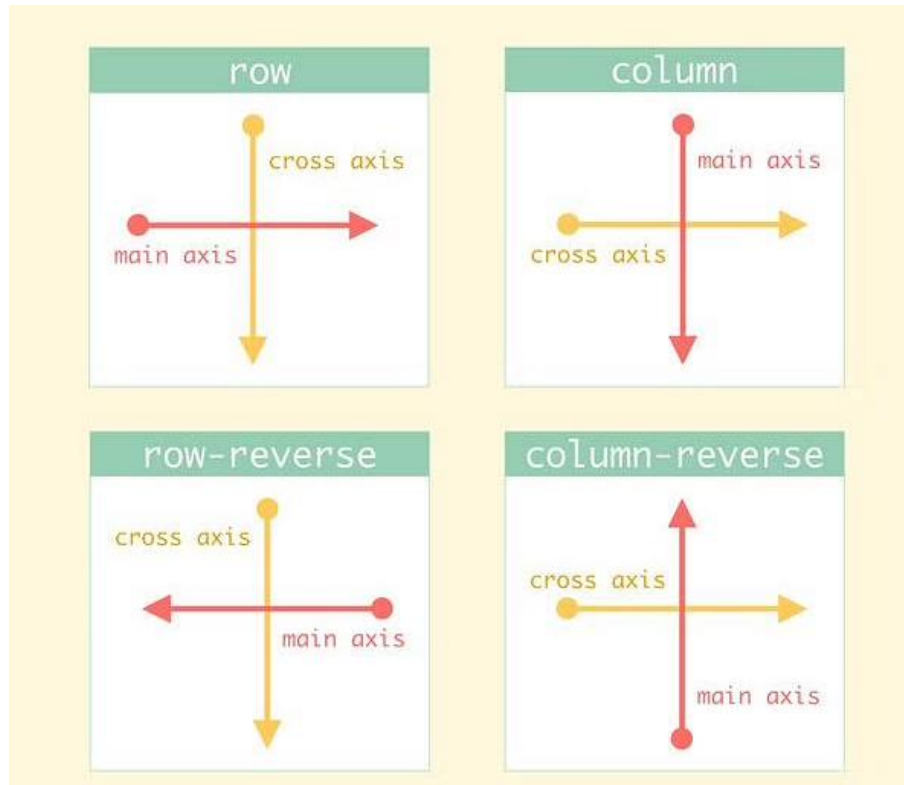
Flex-direction: sirve para cambiar la dirección del eje principal, puede tener los valores: **row**, **column**, **row-reverse**, **column-reverse**.

Para centrar con margin:

Width definido

Display block

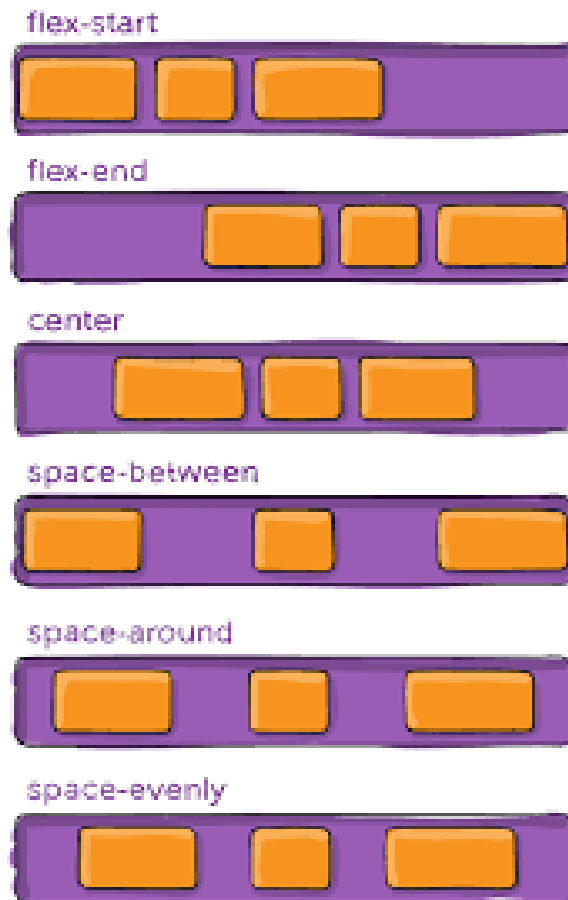
Margin right and left auto



Flex-wrap: permite que se creen nuevas columnas o filas cuando los elementos no quepan en el main axis por el ancho del contenedor. Tiene los valores: **nowrap**, **wrap**, **wrap-reverse**.

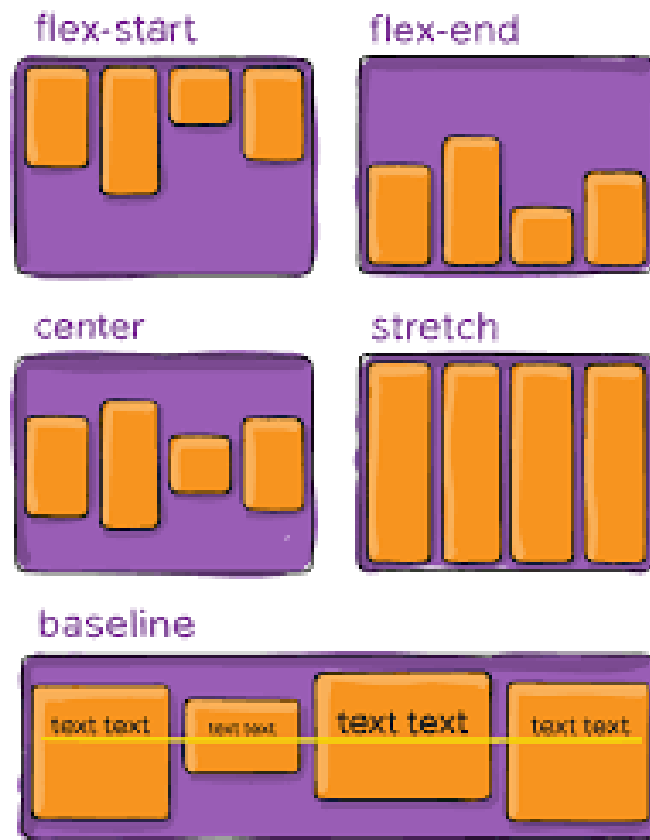
Flex-flow: es un shorthand de **flex-direction** y **flex-wrap**.

Para que las alineaciones sirvan debe haber espacio en el eje donde se quiere alinear. Si queremos alinear en el **main-axis** se utiliza **justify-content** que puede tener los valores: **flex-start**, **flex-end**, **center**, **space-between**, **space around**, **space-evenly**.

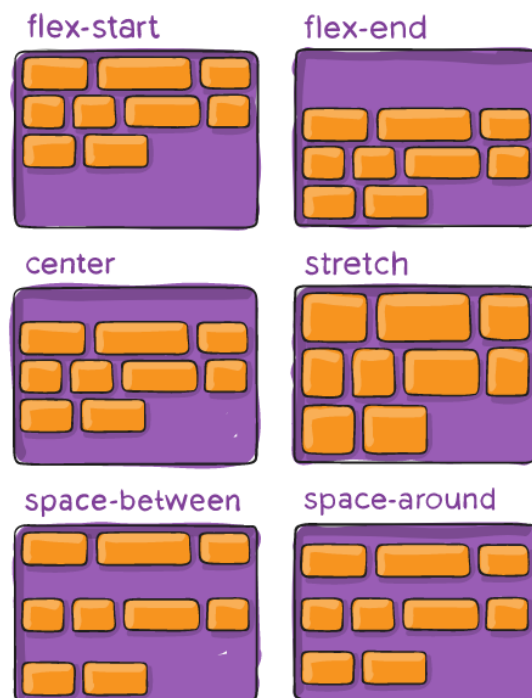


Align-items: permite alinear los elementos en el eje secundario o cross-axis. Los valores pueden ser:

Flex-start, **flex-end**, **baseline**, **stretch**, **center**

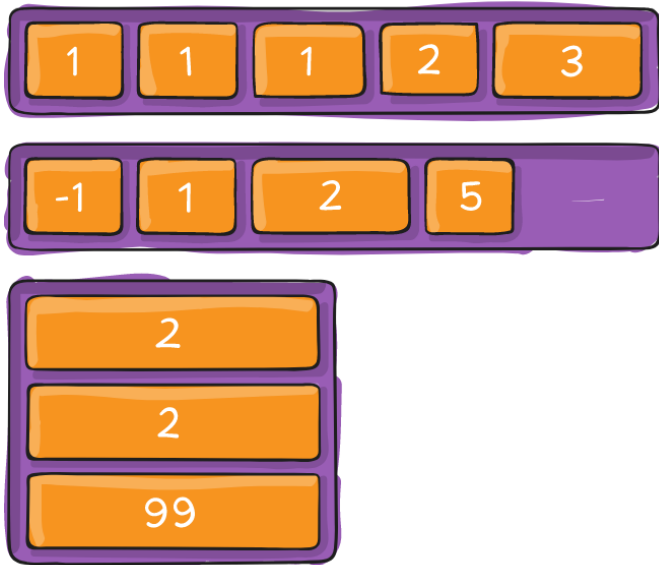


Cuando se coloca flex-wrap, en wrap se desbloquea una propiedad de alineacion ***align-content***, tiene los mismos valores que justify-content.



Propiedades de los elementos del container

Order: permite cambiar el orden de un elemento dentro del contenedor, por defecto todos poseen order de 0. Se ordenan de menor a mayor, permite números enteros.



Para que una imagen ocupe todo su contenedor.

Cambiar su display a block

Flex-grow: es la propiedad que tiene un elemento de crecer dependiendo del espacio sobrante. Por defecto esta desactivado en 0.

Flex-shrink: es la propiedad que tiene un elemento de estrecharse dependiendo del espacio faltante. Por defecto esta desactivado en 0.

Flex-basic: establece el tamaño base de los elementos del contenedor, se coloca por defecto en 0 para que cada elemento se coloque del mismo tamaño con el flex-grow

Flex: es el shorthand de flex-grow, flex-shrink y flex-basic. Se compone de la siguiente manera: primero va el valor de **grow**, luego el de **shrink** y finalmente el de **basic**. Si se coloca en **auto** el grow se coloca en 1, el shrink en 1 y el basic en auto, si se coloca en **1** el grow se coloca en 1, el shrink en 1 y el basic en 0. Si se colocan dos valores estos representan los dos primeros que son grow y shrink y coloca basic en 0.

Align-self: permite alinear un elemento del contenedor y solo ese elemento es afectado por la propiedad.

TRUCO

Los márgenes pueden ayudar a posicionar los elementos colocando dichos márgenes en auto.

Gap: permite dar espacio entre las columnas y las filas, es el shorthand de **row-gap** y **column-gap**. Primero es row-gap y luego column-gap

GRID

Permite manejar tanto columnas como filas.

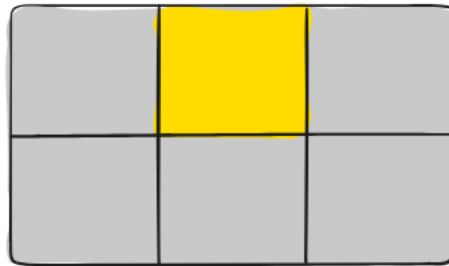
Términos básicos:

Grid container: Para poder utilizar grid se necesita un contenedor.

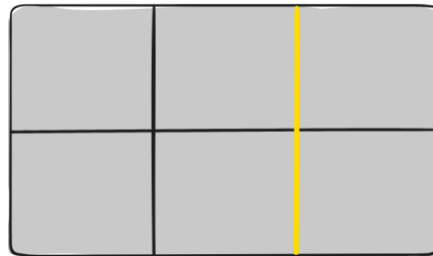
Grid ítem: Son los hijos del contenedor.

Filas y columnas: Las filas son row y las columnas son llamadas column. Cuando las columnas y filas se intersectan se forma la grid.

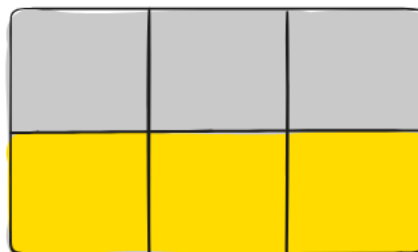
Cell(celda): por default todos los hijos ocupan una celda.



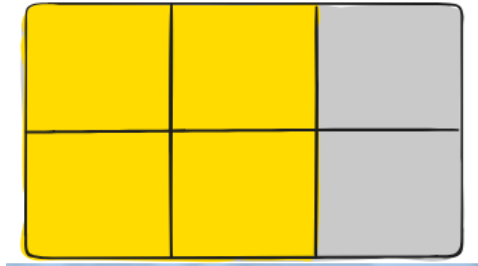
Grid lines: Las cuadrículas están compuestas por líneas verticales y horizontales. Las líneas horizontales se conocen como **row-grid-lines** y las líneas verticales como **column-grid-lines**.



Tracks: Se conoce al espacio entre dos grid lines adyacentes. Término para llamar a las filas y columnas.



Grid área: es el espacio entre entre cuatro grid lines, debe ser cuadrada o rectangular.



Para utilizar grid se debe colocar al contenedor con un **display : grid**, también existe **inline-grid** que actúa como un elemento **inline**.

Su comportamiento por defecto es colocar cada elemento uno debajo del otro.

Grid-auto-flow: comportamiento por defecto para colocar los elementos. Y creara filas o columnas automáticamente si los elementos no caben en las filas o columnas que hemos establecido. Valores: **row**, **column**.

Para definir la cuadrícula de la grid se utiliza las propiedades:

Grid-template-columns: permite definir las columnas, acepta cualquier unidad de medida.

Grid-template-rows: permite definir las filas, acepta cualquier unidad de medida.

Repeat(numero(filas o columnas), medida): sirve para definir de mejor manera el tamaño de las columnas y filas sin tener que repetir varias veces las medidas.

Ejemplo:

Repeat(10,10%) crea 10 columnas o filas del 10%.

Repeat(10, 20% 10%) crea 10 filas intercaladamente una de 20% otra de 10% otra de 20% otra de 10%....

Repeat(5, 20%) 45% creara 5 columnas del 20 % y una columna de 45%

1fr: permite dividir el espacio disponible en fracciones, de manera equitativa.

Auto: el valor auto primero le da el tamaño minimo para que el track exista o celda.

Propiedades de los grid ítems

Para posicionar a los elementos en la grid con las líneas se utilizan las siguientes propiedades:

Grid-column-start: establece la linea inicial vertical o de la columna donde debe posicionarse el ítem.

Grid-column-end: establece la linea final vertical o de la columna donde debe posicionarse el ítem.

Grid-row-start: establece la linea inicial horizontal o de la fila donde debe posicionarse el ítem.

Grid-row-end: establece la linea final horizontal o de la fila donde debe posicionarse el ítem.

Grid-column: es el shorthand de grid-column-start y grid-column-end. Se define de la siguiente manera:

Grid-column-start / grid-column-end

Grid-row: es el shorthand de grid-row-start y grid-row-end. Se define de la siguiente manera:

Grid-row-start / grid-row-end

Grid-area: es el shorthand de grid-column y grid-row, se define de la siguiente manera:

Grid-row-start / grid-column-start / grid-row-end / grid-column-end

Span: sirve para especificar cuantas celdas o tracks debe ocupar un elemento.

Gap: sirve para dar espacio entre filas o columnas, es un shorthand de **row-gap** y **column-gap**.

Grid-auto-flow: permite posicionar los elementos por defecto ya sea en columnas, filas. Cuando los elementos no caben en la filas y columnas establecidas se crean por defecto columnas o filas. Existe un valor **dense** que sirve para llenar celdas que queden vacias por posicionar a los ítems.

Otras propiedades sirven para la creación de filas o columnas implícitamente.

Grid-auto-row: define el tamaño que tendrán las filas que se crean automáticamente. Por defecto esta en auto

Grid-auto-column: define el tamaño que tendrán las columnas que se crean automáticamente. Por defecto esta en auto.

Oder: funciona igual que en flexbox.

GRID AREAS

Grid-template-areas: sirve para definir el nombre y las areas de la grid. Cada fila esta separa por "" y cada columna se escribe dentro de las comillas separadas por un espacio.

GRID AVANZADO

Para darle nombre a las líneas se deben agregar los nombres cuando se crea el template estos encerrados en corchetes, ejemplo:

Grid-template-column: **[line-start] 1fr [line-medium] 1fr [line-end]**

Si se crean areas tambien se crean de forma implicita las lineas de la grid.

VALORES DINAMICOS

Max-content y **min-content**.(grid-template-column, auto-column)

Max-content: se adapta al máximo contenido del elemento.

Min-content: se adapta al minimo contenido del elemento.

Esta función permite crear valores dinámicos **fit-content()**. Es una combinación entre max-content y min-content. Permite que la columna actua como su máximo contenido pero si rebasa el tamaño del fit-content entonces toma el tamaño de la función. Grid-template y gri-auto

Minmax() permite dar un valor minimo y un tamaño máximo del track. El primer valor define el minimo y el otro valor define el máximo.

Auto-fit: permite crear las filas o columnas que vayamos a necesitar.

Auto-fill: permite crear tantos track como columnas o filas tanto el espacio lo permita. Template.

Alinear los ítems

Funciona siempre que exista espacio para poder hacerlo.

Justify-items: Alinear los ítems en las filas. Por defecto es stretch.

Align-items: Alinear los ítems en las columnas.

Place-items: es el shorthand de justify-items y align-items. El primer valor es para align y el otro es para el justify.

Cuando el tamaño de la grid es menor que el contenedor permite utilizar las propiedades para alinear a la grid (NO LOS ELEMENTOS O ITEMS)

Justify-content: permite alinear al eje horizontalmente. **Start, end, center, space-around, space-between, space-evenly**

Align-content: permite alinear al eje vertical.

Place-content: shorthand de justify-content y align-content. El primer valor es de align y el Segundo de justify.

Propiedades que permiten alinear solo a un ítem o elemento

Justify-self: permite alinear al ítem de manera horizontal.

Align-self, permite alinear al ítem de manera vertical.

Place-self: es el shorthand de justify-self y align-self

Shorthand para crear filas y columnas

Grid-template: shorthand de grid template row/ grid template column

Grid: es un shorthand parecido a **grid-template**.

SELECTORES

Selectores de tipo: Un selector de tipo, de nombre, o de etiqueta, es conocido así porque selecciona una etiqueta directamente con el nombre de la etiqueta HTML. El selector es el nombre de la etiqueta tal cuál, sin ninguna modificación.

A code editor window with a dark background and three colored window control buttons (red, yellow, green) in the top left corner. It contains CSS code for the 'h2' selector.

```
h2{  
    color: red;  
    border: 1px solid #000;  
}
```

Selectores de clase: Selecciona todos los elementos que tienen el atributo de class especificado.

Sintaxis: .classname

Selectores de id: Selecciona un elemento basándose en el valor de su atributo id. Solo puede haber un elemento con un determinado ID dentro de un documento.

Sintaxis: #idname

Selector Universal: Permite seleccionar todos los elementos de un html. Tiene especificidad de cero. (*)

Selector de descendientes: El combinador selecciona los elementos que son descendientes del primer elemento.

Sintaxis: A B

Selectores de descendientes directos (>): El combinador > selecciona los elementos que son hijos directos del primer elemento.

Sintaxis: A > B

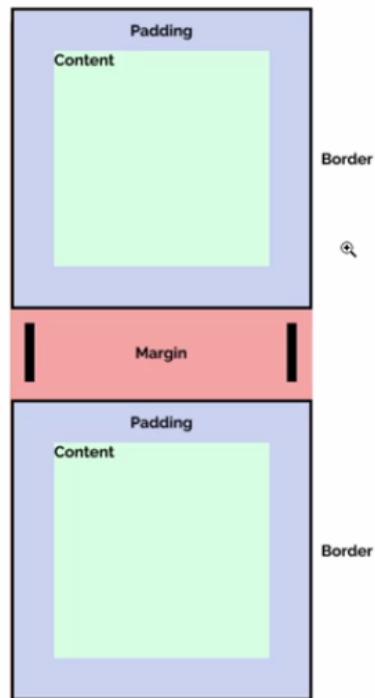
Especificidad

1.Selectores de tipo, pseudoelementos::

10.Selectores de clase, pseudoclases:, []

100. Selectores de ID

BOXMODEL



Padding: Se encuentra entre el borde del elemento y su contenido. Se establece con la propiedad *padding*.

Padding: *k m n l* se agrega *k* arriba, *m* al lado derecho, *n* abajo y *l* al lado izquierdo

Padding: *k m n* se agrega *k* arriba, *m* a los lados y *n* abajo

Padding: *k m* se agrega *k* arriba y abajo y *m* a los lados

Padding: *k* se agrega *k* relleno a cada lado

Border: Permite agregar la línea del border. Esto modifica el color, el ancho y el estilo.

Border-color: Modifica el color-

Border-width: Define el ancho del border, permite colocar thin, médium thick

Border width es un shorthand de border-top-width, border-right-width, border-bottom-width y border-left-width.

Border-style: Dotted, dashed, solid, double, ridge, groove. También es un shorthand.

Margin: Permite establecer márgenes entre elementos.

El colapso de márgenes solo sucede con margin top y margin bottom. Cuando dos márgenes se encuentran no se suman se coloca el margen del mayor.

OTROS ELEMENTOS

Background: Permite establecer el color de fondo de un elemento.

Color: Permite establecer el color de un elemento

Los colores se pueden agregar con **rgb, hexadeciman, nombre, hsl**

Rgb: acepta tres valores el r-rojo, g-verde, b-azul, estos van desde 0-255, rgb(0,0,0). Tambien se puede agregar un cuarto valor que va de 0-1 y establece la opacidad del color.

Hexadecimal: establece el color a partir del lenguaje hexadecimal, siempre empieza con #, los números van de 0-9 y las letras de A-F.

Hsl: h-tono matiz, s-saturacion, l-iluminacion. Acepta tres valores y se puede agregar un nuevo valor para la opacidad. En la saturación y iluminación son porcentajes

Background-color: Sirve para colocar color en el fondo de la imagen.

Background-image: Sirve para colocar una imagen de fondo. Utiliza la función url para establecer a imagen.

Background-size: Permite establecer el tamaño de la imagen de fondo. Tambien sirve: cover, contain,

Background-repeat: Establece si la imagen se debe repetir. Puede tener el valor de repeat-x, repeat-y, no-repeat, repeat

Background-position: Sirve para establecer la posición de la imagen, top, left, bottom, right, center.

Background-attachmen:

Con background podemos colocar gradientes con **linear-gradient()**

TEXTO

Letter-spacing: permite cambiar el espacio entre las letras de un texto.

Word-spacing: permite definir un espacio entre cada palabra de un texto.

Hyphens: nos ayuda a que, si una palabra es muy grande para su contenedor, puede separar con un guion medio a la palabra creando un salto de línea.

Overflow-wrap: se aplica a los texto y se crea un salto de línea si es necesario.

Text-transform: permite transformar el texto entre minúscula y mayúscula.

Text-align: permite alinear al texto.

Font-family: permite definir una familia de lfuentes para la pagina.

Font-weight: permite definir el peso de la fuente.

Font-style: Permite defnir el estilo de la letra.

Font-size: permite modificar el tamaño de la letra de la fuente.

Line-height: permite definir el alto de la línea de los textos. Su valor por defecto es 1.2.

MEDIDAS

Existen dos tipos de unidades las absolutas y las relativas.

Las absolutas son absolutas porque tienen un valor definido, sus valores no dependen de nada más. Ya que son definidas previamente.

Ejemplo: px, pt, mm.

Las relativas dependen de algo externo a ellas.

Rem: es relativa al tamaño de fuente o al Font-size de la raíz.

Em: es relativa al tamaño del padre más cercano con una fuente declarada.

Vw: se utilizan valores de 0 a 100, es relativa al viewport view

Vh: es relativa a la parte visible de alto del viewport view.

POSITION

STATIC: es el que tiene por defecto.

RELATIVE: desbloquea top, left, bottom, right. Permite desplazar al element en base de la posicion original del element. Deja el espacio reservado a pesar de que se desplace a otro lugar.

ABSOLUTE: Es similar al anterior. Se remueve del flujo natural del elemento. También desbloquea top, left, right, bottom. Es relativo al contenedor padre que tenga una posición diferente a **static**.

FIXED: es bastante similar absolute. Hace que el elemento salga del flujo, y se posiciona con respecto al viewport, y se pega.

STICKY: es una mezcla entre relative y fixed. Primero es relative hasta que toque a su contenedor padre y se queda pegado. Para que funcione el contenedor padre debe tener una altura definida.

Z-INDEX

Todo elemento que tenga una position distinta a static desbloquea z-index. Funciona para saber que elemento debe posicionarse encima de un elemento. Por defecto es auto.

MEDIA QUERIES

Son esenciales para adaptar la página dependiendo de las condiciones que se presenten. Se declaran con **@media(condicion){**

}

Max-width: define el tamaño máximo del cual se ejecutará el query. Se coloca de mayor a menor

Min-width: define el tamaño mínimo del cual se ejecutará el query. Se coloca de menor a mayor

Siempre van al final del css.

CUSTOM PROPERTIES

Sirven para crear variables. Se declaran en

:root{

--nombre_de_la_variable:

}

BOX-SHADOW, BORDER RADIUS

Box-shadow: sirve para dar una sombra al elemento, mínimo necesita dos valores, el primer valor es el desplazamiento en x, el segundo valor es el desplazamiento en y, el tercer valor opcional es el difuminado del elemento, el cuarto valor es el esparcimiento del elemento, también permite otro valor que sería el color de la sombra. Se puede colocar más de una sombra.

Border-radius: permite redondear las esquinas de un elemento, es un shorthand.

SELECTORES COMBINADORES

Selector de hermano adyacente(+): Selecciona al hermano siguiente y solo al hermano siguiente del primer selector.

Selector de hermanos(~): Selecciona a los hermanos de un elemento ya sea el siguiente inmediato o no.

PSEUDOCLASSES

Son un tipo de selector en CSS, permite generar interacciones en la pagina. Empieza con `:`. Algunas pseudo clases son:

`:hover`: se activa cuando el usuario pasa el mouse sobre el elemento.

`:active`: representa un elemento (como un botón) que el usuario está activando. Cuando se usa un mouse, la "activación" generalmente comienza cuando el usuario presiona el botón primario del mouse y termina cuando se suelta.

`:checked`: representa cualquier radio, checkbox u option que está marcado o conmutado a un estado on

`:default`: representa cualquier elemento de formulario que sea el predeterminado entre un grupo de elementos relacionados.

`:disabled`: representa a cualquier elemento deshabilitado. Un elemento se encuentra deshabilitado si no puede ser activado (por ejemplo ser seleccionado, hacer click en él o aceptar texto de entrada) ni aceptar el foco. El elemento tiene además un estado habilitado en el cual puede ser activado o recibir foco.

`:empty`: representa cualquier elemento que no tenga hijos. Los hijos pueden ser nodos de elemento o texto (incluido el espacio en blanco). Los comentarios o las instrucciones de procesamiento no afectan si un elemento se considera vacío.

`:enable`: representa cualquier elemento habilitado. Un elemento está habilitado si puede ser activado (es decir seleccionado, se puede hacer click en él, acepta que se le introduzca texto, etc.) o si acepta el foco. El elemento también tiene un estado deshabilitado en el cual, no puede ser activado ni acepta el foco.

`:first-child`: representa el primer elemento entre un grupo de elementos hermanos.

`:first-of-type`: representa el primer elemento de su tipo entre un grupo de elementos hermanos.

`:focus`: representa un elemento (como una entrada de formulario) que ha recibido el foco. Generalmente se activa cuando el usuario hace clic, toca un elemento o lo selecciona con la tecla "Tab" del teclado.

`:in-range`: representa un elemento cuyo valor actual se encuentra dentro de los límites de rango especificados por los atributos `min` y `max`.

`:invalid`: Esta pseudo-clase es útil para resaltar errores de campo para el usuario.

`:last-child`: representa el último elemento entre un grupo de elementos hermanos.

`:last-of-type`: representa el último elemento de su tipo entre un grupo de elementos hermanos.

`:link`: representa un elemento que aún no se ha visitado.

`:not()`: representa elementos que no coinciden con una lista de selectores. Como evita que se seleccionen elementos específicos, se lo conoce como la *pseudoclase de negación*.

`:nth-child`: coincide con uno o más elementos en función de su posición entre un grupo de hermanos. Even or odd.

:nth-last-child: selecciona uno o más elementos en función de su posición entre un grupo de hermanos, contando desde el final.

:nth-last-of-type: coincide con uno o más elementos de un tipo dado, en función de su posición entre un grupo de hermanos, contando desde el final.

:nth-of-type: selecciona uno o más elementos de un tipo dado, en función de su posición entre un grupo de hermanos.

:only-child: representa un elemento sin hermanos

:only-of-type: representa un elemento que no tiene hermanos del mismo tipo.

:optional: representa cualquier elemento `<input>`, `<select>`, o `<textarea>` que no tenga el atributo `required` establecido en él.

:out-of-range: representa un elemento `<input>` cuyo valor actual está fuera de los límites de rango especificados por los atributos `min` y `max`.

:required: representa cualquier elemento `<input>`, `<select>`, o `<textarea>` que tenga el atributo `required` establecido en él.

:target: representa un elemento único (el *elemento objetivo*) con un `id` que coincide con el fragmento de la URL.

:visited: representa enlaces que el usuario ya ha visitado. Por motivos de privacidad, los estilos que se pueden modificar con este selector son muy limitados.

CLIP-PATH

Permite realizar un recorte con una forma concreta, ocultando toda la región externa del recorte. Su utilización es muy sencilla y permite realizar formas muy flexibles y versátiles.

TRANSICIONES

Es una de las formas de animar un elemento en CSS, lo cual nos permite animar el cambio de un estado a otro.

No todas las propiedades son animables en CSS. Debe existir un disparador para que la animación suceda.

Las animaciones consumen recursos del sistema. Procura usar transform y opacity para animar. El navegador tiene un proceso de renderización.

Transition: es un shorthan. Se puede colocar en diferente orden, pero el primer tiempo es para la duración.

Transition-property: especifica la propiedad o propiedades que sufrirán una transición.

Transition-duration: define la duración de las transiciones.

Transition-timing-function: establece la curva de transición. **Ease, ease-in, ease-out, ease-in-out, linear.**

Transition-delay: define el tiempo de espera para que una transición se ejecute.

TRANSFORMACIONES

Translate(): permite trasladar al elemento en las coordenadas. Es un shorthand de **translateX, translateY.**

Rotate(): permite rotar el element, permite valores angulares.

Scale(): permite escalar un elemento.

Transform-origin: permite indica desde que punto se ejecuta la transformación.

SkewX(), skewY(): utiliza valores angulares.

Will-change: permite optimizar el navegador solo en caso de que haya problemas de rendimiento.

ANIMACIONES

Se utiliza la propiedad animation. Se crea un **@keyframes nombre de la animación**, contiene dos elementos **from**, y **to**.

Animation-name: referencia al keyframes que queremos agregar al elemento.

Animation-duration: es la duración de la animación.

Animation-delay: es el tiempo de espera para que se ejecute la animación

Animation-iteration-count: define cuantas veces se repitera la animación, existe el valor infinite.

Animation-direction: define la dirección de la animación, **reverse**, **normal**, **alternate**, **alternate-reverse**.

Animation-time-function:

Animation-fill-mode: define lo que sucede antes o después de la animación. **None**, **forwards** hacer que una vez que finalice la animación hace que el elemento mantenga las características del keyframes aplicados en el 100%. **Backwards** define que los estilos del primer bloque los tenga el elemento. **both**

Animation-olay-state: pause, running.

Animation: es el shorthand.