

1 CURSO DE CSS

```
.VIEWPORT .TODO_LO_QUE_SE_VE{
  /*todo lo que se puede ver en la pantalla osea la pagina osea no sale ni las url ni los cosos
del
  navegador osea es como si estuviéramos en pantalla completa
}
```

```
.PROPIEDADES_DEL_TEXTO {
  font-size: /*calidad o tamaño*/;
  font-family: /*tipo de fuente se puede traer de google font*/;
  line-height: /*el espacio que ocupa un letra//cuando es un numero cualquiera siempre
busca;
      que el numero puesto ocupe todo el espacio de la letra que se divida por decirlo
asi */;
  font-weight: /*grosor a el texto*/;
  color: /*color de la letra*/;
}
```

```
.NORMALIZE{
  /*buscar en google normalize y copiar la direccion del codigo y nos sirve para normalizar
las
  propiedades que vienen puestas por el navegador*/
}
```

```
.DISTANCIA_CONTENIDO_Y_CAJA{
  /*distancia entre el contenido y el borde de la caja*/
  padding: ; /*el normal se puede usar todos a la vez ==> 10 px lo aplica a todos tambien dos
que son
      top y left que se dan asi ==> 10px 20px = 10px son top y bottom y 20px son left y
right
      o tambien los cuatro de una en este orden ==> 10px 20px 30px 40px = 10px es top,
20px
      es right, 30px es bottom y 40px es left */
  padding-top: /*significa arriba*/
  padding-bottom: /*significa abajo*/
  padding-left: /*significa izquierda*/
  padding-right: /*significa derecha*/
}
```

```
.ANCHURA_Y_ALTURA {
  height: ; /*altura del documento o propiedad de un objeto y modifica al que nos refiramos*/
  width: ; /*anchura del documento o propiedad de un objeto y modifica al que nos
refiramos*/
}
```

```
.BOX-SIZING {
  box-sizing: ;/*nos permite incluir relleno y el borde en el ancho y alto total de un elemento*/
```

```
}
```

```
.MARGENES {
```

```
/*distancia entre las cajas y se usa parecido al padding*/
```

```
margin: ; /*todas*/
```

```
margin-top: ; /*arriba*/
```

```
margin-bottom: ; /*abajo*/
```

```
margin-left: ; /*izquierda*/
```

```
margin-right: ; /*derecha*/
```

```
}
```

```
.BORDES {
```

```
border: ; /*primero se da el tamaño, luego estilo, luego color*/
```

```
/*estilos para personalizar: ;
```

```
solid, dashed, double, groove, inset, outset, ridge hay muchos tipos*/
```

```
border-radius: ; /*redondea*/
```

```
}
```

```
.MODELO_DE_LA_CAJA{
```

```
/*modelo de trabajar las cajas:
```

```
content, padding, border y margin tienen gerarquia
```

```
content ==> line-height
```

```
padding ==> padding
```

```
border ==> border
```

```
margin ==> margin.
```

```
1 va el content, 2 el paddign, 3 el border y el 4 es el margin en este orden se ponen los elementos*/
```

```
}
```

```
.SOMBRAS {
```

```
box-shadow: ;/*permite dar sombra a un caja y uno se mueve box-shadow: 10px este es eje x, 20px es el
```

```
eje y, 30px decimos cual tamaño va a tener el desenfoque, 0px cual borde va a tener y # el color.
```

```
quedaria asi:*/
```

```
box-shadow: 10px 20px 30px 0px #ffffff; /*asi es*/
```

```
/*tambien esta la sobra pero para el text se usa igual*/
```

```
text-shadow: 10px 20px 30px #ffffff; /*diferencia es que el este no tiene la propiedad borde*/
```

```
}
```

```
.OUTLINE {
```

```
/*es una propiedad acortada porque es el conjunto de otras propiedades es como con el padding
```

```
que acorta una en vez de escribir padding, se pone la que queremos modificar padding left o cualquiera
```

```
outline es como un border pero sin afectar a las cajas no ocupa espacio real en el DOM*/
```

```
/*es como un "shorthand" es propiedad abreviada a si se define*/
```

```
outline: ; /*outline no ocupa espacio no afecta a los otros DOM es como para enmarcar
objetos*/
}
```

```
.SHORTHAND{
  /*se le llama shorthand a una propiedad abreviada o acortada*/
}
```

```
.POSITION{
  /*lo que hace es posicionar elementos o sea adquiere nuevas propiedades.
  si un elemento esta significa que afecta el flujo html */
  position: static; /*valor por defecto o sea no esta posicionado*/
  position: relative; /*adquiere la propiedad que esta abajo y adquiere una nueva propiedad
  que se llama
  zindex y aparte hace que el espacio se conserve*/
  position: absolute; /*parecido a relative si no que hay 2 cosas que cambian: no es como el
  relative que conservaba el espacio si no que el elemento de abajo sube porque no esta
  reservado el espacio. si no definimos es como que se quedara en la caja porque es el
  punto de
  referencia. el ancho aunque sea un elemento bloque el ancho se ajusta al contenido*/
  position: fixed; /*exactamente que el absolute pero queda fijado*/
  position: sticky; /*mezcla entre fixed y relative o sea conserva el espacio y se comporta
  como un
  fixed cuando llegamos al punto en el que le definimos*/
  /*se usa con estas propiedades aveces*//*las propiedades top y left son mas importantes
  que las otras*/
  top: ; /*se mueva hacia abajo*/
  bottom: ; /*se mueva hacia arriba*/
  right: ; /*se mueve hacia la izquierda*/
  left: ; /*se mueve hacia la derecha*/
}
```

```
.zindex {
  /*zindex la forma en la que ponemos un elemento por encima de otro o sea en el eje z los
  ordena y los ordena por el valor mas alto es el que va a adelante o sea se pone por
  encima de los otros*/
  /*solamente funciona solo cuando la caja es posicionada*/
  position: relative;
  z-index: 1;
  /*recomiendo que los z index dejen varios espacio por lo menos 10 espacios para cuando
  le toque
  cambiar uno no los cambie todos si se encuentra cerca si no cambie ese si lo quiere
  primero o antes*/
  /*y el z-index tiene conflicto entre los padres e hijos o sea asi hagamos mil z-index el padre
  no va a estar por encima del hijos la unica forma seria que al hijo le pusieramos = -1 con el
  -
  y en el padre no tiene que estar definido*/
  z-index: -1;
```

```
}
```

```
.DISPLAY {
```

```
/*lo que hace es modificar el comportamiento de las cajas*/
```

```
display: block; /*ocupan todo osea es como un h1 que ocupa todo*/
```

```
display: inline; /*solamente lo que el contenido da osea es como distinto al block si no a la vez es como al reves porque los pone solo el contenido y a la vez se ponen en la linea */
```

```
display: inline-block; /*son las mezcla de los anteriores solo que se pueden modificar los elementos de la caja*/
```

```
display: flex; /*se comporta como bloque pero lo que esta dentro de la caja no*/
```

```
display: grid; /*se comporta como bloque pero lo que esta dentro de la caja no*/
```

```
display: inline-flex; /*es como un inline block*/
```

```
display: inline-grid; /*es como un inline block*/
```

```
}
```

```
.OVERFLOW { /*(Y, X) AUTO, HIDDEN, SCROLL*/
```

```
/*es la barra de al lado*/
```

```
/*osea el overflow es que hacemos con lo que sobra si algo sobresale de la pantalla*/
```

```
overflow: hidden; /*hace dejar que se sobresalga osea es como fuera invisible*/
```

```
overflow: auto; /*si se sale da posibilidad de scrollear dentro de la caja*/
```

```
overflow: scroll; /*lo que hace es obligar a poner la barra de scrollear*/
```

```
/*overflow es un shorthand porque tiene estas propiedades*/
```

```
overflow-x: scroll; /*esta agrega el scroll pero el del eje x nada mas*/
```

```
overflow-y: scroll; /*esta agrega el scroll pero el del eje y nada mas*/
```

```
/*y si queremos ocultar alguno porque salen los dos con*/
```

```
overflow-x: hidden; /*oculta el del eje x*/
```

```
overflow-y: hidden; /*oculta el del eje y*/
```

```
}
```

```
.FLOAT { /*(Left, Right, None, Inherit)*/
```

```
/*forma antigua de hacer las paginas la unica cosa para la que se puede usar que puede ser
```

```
necesaria que se puede usar como si el texto envolviera la imagen como un tipo de doc en word para hacer un efecto bonito*/
```

```
float: left;
```

```
float: right;
```

```
float: none;
```

```
float: inherit;
```

```
}
```

```
.PSEUDO-ELEMENTOS {
```

```
/*es un elemento que no es un elemento osea es un elemento pero que no forma parte del DOM
```

```
es como el outline*/
```

```
/*PARA USARSE SE USA EL EJEMPLO ESTA EN LA SIGUIENTE
```

```
/*first-line solo funciona en block ==> cambiar la linea osea sus propiedades cualquiera*/
```

```
/*first-letter solo funciona en block ==> es lo mismo solo que la primera letra*/
```

/*placeholder es lo que aparece adentro = es como una casilla de texto y dice que insertar un nombre alli y lo modificamos mediante ese mismo pseudoelemento*/
/*selection es lo que al seleccionar creamos una etiqueta y la podemos modificar si no que con propiedades reducidas*/

/*los siguientes necesitan si o si la propiedad content y no son elementos
osea crean como un hijo en el elemento y no se puede seleccionar*/
/*after ==> hijos - contenido (necesario) y funciona con inline es es como antes*/
/*before ==> hijos - contenido (necesario) y funciona con inline este es como si no existiera
y este es como si fuera un despues*/

}

/*EJEMPLO DE LA ANTERIOR*/

```
.ejemplo::first-line{  
  /*asi se usa la propiedad anterior*/  
}
```

.PSEUDO-CLASE{

/*son como escuchas para eventos*/
/*hover = escucha el evento cuando el mouse esta encima*/
/*link = cambia las propiedades de un link que todavia no visitamos y si lo abrimos queda normal y
ya no cambia*/
/*visited = es lo contrario de la anterior osea cambia propiedades de los que ya estan visitados*/
/*active = al dejar presionado la caja cambia los estilos o propiedades*/
/*focus = se aplica mas que todo a los inputs es parecido al anterior pero es mas con inputs*/
/*lang = esta es diferente ya que a esta hay que pasarle una abreviacion del lenguaje el ejemplo
despues de hover esta solo es para lo que este con ese lenguaje a eso se le cambia las propiedades
y para eso tenemos que darle un parametro a la etiqueta en html y luego darselo aqui en css*/

/*como se usan en la proxima*/

}

.pseudo-clase:hover {

/*asi se utilizan*/
/*todas menos lang se utilizan asi*/

}

.pseudo-clase:lang(en) { /*el parametro que se da es la abreviacion del idioma en es del english*/

/*aca cambia las propiedades si es correcto*/

}

.OBJECT-FIT { /*se aplica mas que todo para la resoluciones de las imagenes

/*contain = hace que la resolucion se ajuste al contenido*/

```

/*cover = la imagen se ajusta al contenedor*/
/*none = usa las propiedades por defecto lo otro se lo pasa*/
/*scale-down = se queda con la mejor propiedad si cualquiera de contain o none es mas
chiquito
deja el mas pequeño*/
object-fit: ;
}

```

```

.OBJECT-POSITION{ /*tambien se usa como al de arriba solo que cambian cosas*/
/*se puede trabajar con left, right, top y bottom osea con estas se mueve a la direccion
hay planteada*/
/*tambien se puede con estas px, em, % y ETC*/
object-position: right; /*se ubica a la derecha este es un ejemplo*/
}

```

```

.CURSOR {
/*cambiar el cursor al estar encima de un elemento*/
/*lista de cursores esta en la siguiente pagina:
https://www.w3schools.com/cssref/pr\_class\_cursor.asp */
}

```

```

.COLORIZACION {
/* hay varias formas de aplicar color a algo*/

/*1.coloname = blue o asi.

```

2.hexadecimal = se trabaja con los 16 caracteres de hexadecimal se trabaja con # esto y se pueden poner asi 3 unidades: #fff = la primera f es rojo como en rgb la segunda es verde y la tercer f es azul; 4 unidades: #fff9 = como la de 3 unidades pero en este case 9 equivale a la opacidad del elemento; con 6 unidades: #ffffff = es como la de 3 unidades pero las dos primeras ff son rojo y asi sucesivamente; con 8 unidades: #ffffffff = es como la de 4 unidades pero son; de dos en dos y las dos ultimas son la opacidad es como la de 6 si no que se agregan las de opacidad.estas son las unicas que se pueden ni mas ni menos.

3.rgb = mezclar los colores primarios que son red, green y blue.

4.rgba = este tiene cuatro parametros es lo mismo que el de arriba pero se le agrega la opacidad a lo ultimo es una abreviacion de el de arriba.

5.hsl = estas existen pero casi no se usan.

6.hsla = estas existen pero casi no se usan.*/*

background: red; /*ejemplo de la primera solo con poner el nombre del color en inglesya cambia*/

background: #ffffff; /*asi es hexadecimal son 16 numeros.*/

background: rgb(5, 10, 500); /*se tienen que dar los 3 valores y van en orden maximo es 255 de cada uno de los 3;*/

background: rgba(10,50,120,.9); /*el cuarto parametro es la opacidad del color es como el anterior pero se le agrega eso*/

/*las otras las agrego ahorita*/

}

.RESPONSIVE-DESING_Y_MOBILE-FIRS {

/*responsive desing = trabajar con las distintas resoluciones.

mobile firs = adapta resoluciones del un celular a una tablet y despues al escritorio*/

/*el ejemplo en la siguiente*/

}

/*ejemplo*/

div {

width: 50%;

}

@media only screen and (max-width: 800px) {

/*aca van las propiedades que se van a cambiar cuando la resolucion sea lo que pusimos*/

div {

/*propiedades*/

width: 100%;

}

}

.FLEX-BOX {

/*es una nueva forma de enmaquetar*/

/*main axis es como un eje x

y el cross axis es como el y,

el main axis se compone de dos main start y main end y de igual forma el cross axis*/

/*que es*/

/*contenedor y contenido se divide en dos flex container

y flex item

Propiedades:

- flex-direction = -row = vamos a ir como van las filas
- column = se comportan como si fueran columnas
- row-reverse = osea no va a ir de derecha a izquierda si no que al revés
- column-reverse = osea como el de arriba pero columnas empezando desde

abajo

-flex-wrap = no-wrap, wrap y wrap-reverse = osea las cajas o lo que sea que siga para no perder

la propiedad que se vaya para abajo

-flex-flow = flex-direction, flex-wrap = esto es como para no hacer tanto osea en una linea de

codigo

-justify-content = center, space-around, space-between, space-evenly(main axis) = funciona

para alinear el contenido haciendolo mas facil

-align-items = flex-start, center, flex-end, stretch, baseline(cross axis) = solo se usa con una linea osea si se pasan ya no modifica

-align-content = mismas propiedades de la de arriba pero esta si es para todas las lineas

Propiedades de los items:

align-self = (cross axis) para posicionar un solo elemento

Funcionamiento del margin:

es diferente a cuando lo usamos normal

Otras Propiedades:

-flex-grow = agarra espacio sobrante y lo reparte en las cajas que quedan

-flex-basis =

-flex-shrink = para referirnos a que caja va a ceder el espacio

-flex = shorthan de los 3 anteriores

-order = es como un z-index pero en el eje que apunta main axis osea como en (x) y en (y)*/
}

```
.DISPLAY-FLEX {
  display: flex;
}
```

```
.DISPLAY-GRID {
  display: grid;
  /*conceptos basicos
  -grid
  -grid-container = todo el contenedor es como el flex container
  -grid-item = elementos que forman parte del container hijos directos
  -grid-cell = son cada una de las divisiones en el container
  -grid-tracks(column y row) = filas + columnas
  -grid-area = las definimos nosotros son como lo que queremos a hacer en 4 celdas o asi
  -grid-line(column line y row line) = cantidad de lineas de las columnas y filas
```


Propiedades: ;
grid-template-rows(grid-container)
grid-template-columns(grid-container)
grid-row-gap(grid-container)
grid-column-gap(grid-container)
grid-gap(grid-container) = se separan entre ellas

unidades = auto, fr

grid-row-start(grid-item)
grid-row-end(grid-item)
grid-column-start(grid-item)
grid-column-end(grid-item)
grid-row(grid-item)
grid-column(grid-item)

repeat: 3, 150px; dos parametro hacemos que repita 3 veces 150px

```
*/  
}
```

```
.GRID-IMPLICITO-Y-EXPLICITO {  
  /*hacer unas grillas y que una sobre y tenga diferentes cualidades a las otras y abquiere 3  
  propiedades nuevas*/  
  grid-auto-rows: /*igual que el template*/;  
  grid-auto-columns: /*igual que el template*/;  
  grid-auto-flow: column row o dense; /*dense sirve para cuando hay errores en column y  
  row para que los llene*/  

```

```
}
```

```
.GRID-DINAMICO {  
  grid: repeat(3, minmax(300px));  
  grid: repeat(3, min-content);  
  grid: repeat(3, max-content);  
  /*cantidad*/  
  grid: repeat(auto-fill, max-content);  
  grid: repeat(auto-fit, min-content);  

```

```
}
```

```
.LLINEAMIENTO-Y-CONTROL-DE-FLUJO {  
  /*diferencia con flex  
  -la alineacion es adentro  
  de cada celda y no total como en flex  
  -cada celda seria un flex-container  
  propiedades de grid-container
```

```

*/
justify-items: center;
align-items: start;
place-items: end;

/*content*/
justify-content: space-around;
align-content: space-between;
place-content: space-evenly;
/*self*/
justify-self: start;
align-self: end;
place-self: center;
order: 3;
}

.GRID-AREAS {
  grid-template-areas:
    "header header header"
    "aside aside aside"
    "footer footer footer";
  grid-area: header, aside, main, footer;
}

.NOMBRAR-LINEAS {
  /*[LINE-NAME]*/
}

.GRID-SORTHANDS {
  grid-template: ;
  grid: ;
}

.RESPONSIVE-DESING_COMPLETO {
  /*adaptador a diferentes resoluciones
  estructuras=(flexibles, contenedores, imagenes, videos flexibles)
  @media usada en consulta de terminos para aplicar diferentes estilos para diferentes
  tipos(
  dispositivos de medios
  -ancho y alto de ventana grafica
  -ancho y alto de dispositivo
  -orientacion(¿tableta/telefonos modo vertical o horizontal))
  -resolucion

  parametros = all = todos los dispositivos

```

print = materias para imprimir o impreso en modo de vista previa
screen = destinado mas que todo a las pantallas
speech = destinada a sintetizadores de voz
operadores = and | or

orientacion = lanscape = mas ancho que largo escritorio
portrait = mas largo que ancho mobile

opciones del responsive desing
mobile firs = trabaja de pequeño a grande recomendada
content firs = solo trabaja contenido */

```
}  
.TRANSITION {  
  transition: ;  
  transition-property: ;  
  transition-duration: /*duracion/s*/;  
  transition-delay: /*tiempo para que se ejecute*/;  
  transition-timing-function: linear | ease | ease-in | ease-out | ease-in-out | step-start |  
  step-end  
  | steps(int, start, end) | initial | inherit | infinite;  
}
```

```
.ANIMATION {  
  animation-name: ;  
  animation-duration: ;  
  animation-delay: ;  
  animation-direction: ;  
  animation-timing-function: ;  
  animation-iteration-count: 1 | infinite;  
  animation-iteration-count: ;  
  animation-fill-mode: none | forwards | backwards | both | initial | inherit | infinite;
```

/*esto se usa con @key frames acontinuacion
define las animacion
from {
 cuando empieza
}
to {
 cuando termina
}
se puede cambiar por 0% o 100% y lo bueno de cambiar es que podemos usar 50% o los
que sea entre esos dos
ejemplo en los siguiente*/

/*cubic-bezier es una propiedad de animation-timing-function: cubic-bezier(1.23, 23.4, 3.5,
4.56);
los dos valores primeros son eje X y Y pero en el punto 1 y los otros 2 son X y Y pero en el
punto 2

```
    esto es una curva es parecida a las curvas de color de un editor*/  
}
```

```
@keyframes {  
    0% { /*inicio*/  
  
    }
```

```
    100% {  
        /*final*/  
    }  
}
```

```
.TRANSFORM {  
    /*translate*/  
    transform: translate(10px, -20px);  
    transform: translateX(10px);  
    transform: translateY(-20px);  
    transform: translate3d(10px, -20px, 50px);  
    transform: translateZ(50px);  
    /*scale*/  
    transform: scale(3);  
    transform: scaleX(2);  
    transform: scaleY(5);  
    transform: scaleZ(7);  
    transform: scale3d(2, 5, 7);  
    /*skew*/ /*deg es grados*/  
    transform: skew(10deg);  
    transform: skewX(10deg);  
    transform: skewY(20deg);  
    /*css clip-path maker = pagina ayuda a hacer las formas mas facil con skew*/
```

```
    /*RECOMENDACION USAR GENERADORES PARA HACERLO MAS FACIL*/  
}
```

```
.BACKGROUND {  
    background-color: blue;  
    background-image: url();  
    background-size: cover, contain;  
    background-position: left, top, bottom, right | left, top, bottom, right; /*osea dos parametros*/  
    background-attachment: scroll, fixed, inherit;  
    background-clip: border-box, padding-box, content-box;  
    background-origin: ; /*origen de la foto*/  
}
```

```
/*las variables se crean asi para ser globales*/
```

```
.root {  
    --var-color: blue;
```

```
}
```

```
.VARIABLES-EN-CSS {  
  /*aca un ejemplo de como se llaman las variables creadas*/  
  background: var(--var-color);  
}
```

```
.FILTER {  
  /*poner filtros en nuestro documento*/  
  filter: none;  
  filter: blur(5px);  
  filter: brightness(0.5);  
  filter: contrast(0.3 or 30%);  
  filter: drop-shadow(20px 50px);  
  filter: grayscale(60%);  
  filter: hue-rotate(90deg);  
  filter: invert(50%);  
  filter: opacity(50%);  
  filter: saturate(200%);  
  filter: sepia(50%);  
  filter: url("filter.svg#filter-id");  
}
```

```
.OTRAS-PROPIEDADES-FUNCIONES-Y-SELECTORES-EN-CSS {  
  direction: initial | inherit | unset | rtl | ltr; /*es como el text-align*/  
  letter-spacing: ; /*separa las letras*/  
  scroll-behavior: ; /*transicion para cuando uno le mete animacion a la barra o scroll*/  
  user-select: ; /*hacer que el usuario pueda seleccionar o no*/  
  text-shadow: ; /*sombras del texto*/  
}
```

```
.RECOMENDACION {  
  /*  
  -Los selectores se pueden encontrar en google busca selectores de css y sale la  
  documentacion oficial  
  -Estar mirando constantemente lo que salga nuevo en el lenguaje  
  -Practicar lo aprendido  
  */  
}
```