**permite realizar ejercicios en línea:**

[**https://developer.mozilla.org/es/docs/Learn/CSS/Introduction\_to\_CSS/Pseudo-clases\_y\_pseudo-elementos**](https://developer.mozilla.org/es/docs/Learn/CSS/Introduction_to_CSS/Pseudo-clases_y_pseudo-elementos)

**Mapa de caracteres ascii**

**https://elcodigoascii.com.ar/codigos-ascii-extendidos/simbolo-marca-registrada-codigo-ascii-169.html**

**HOJA DE ESTILOS EN CASCADA CSS (**Cascading Style Sheets)

<http://www.htmlhelp.com/es/reference/css/>

Es un Lenguaje que sirve para definir el estilo a un documento con el fin de definir su presentación y aspecto de una página web, conjunto de reglas que determinan como luce los elementos del sitio web

HTML = Definir La Estructura

CSSS = Aplicar Formato Grafico

Css define la forma como deben renderisarse los diferentes elementos de un documento html en pantallas, impresos o los diferentes dispositivos

* Selectores avanzados
* Relación con tipografias web
* Propiedades de accesibilidad
* Manipular elementos html en 2 y 3 dimenciones
* Herramientas para generar animación sin usar lenguajes sript
* Modos de color avanzado
* Integración con la mayoría de los navegadores

**Los estándares son definidos por la W3C consocio:** organismo, formado por varias organizaciones, tiene como misión establecer estándares para la web, publicar directrices para crear webs usables y accesibles

**Principio**: una web para todos con acceso desde cualquier dispositivo

**Responsive Web Design** (Diseño Web Adaptativo): busca que el contenido web se vea correctamente en cualquier dispositivo a través del uso de media-queries (modulo CSS3) que permite controlar la presentación de HTML a partir de condiciones basadas en ancho, alto y color de los dispositivos de salida.

**Mobile first**: metodología de diseño web adaptativo que se usa para Desarrollar sitios web que se rendericen óptimamente en dispositivos mobiles y luego según el tamaño en pantallas de computador.

**FrameWorks**: entornos de desarrollo (Bpptstrap, Less, Framework, Skeleton y Foundation)

**Librerías CSS**: son sistemas grid con decenas de componentes pre-configurados, tipografías web entre otras opciones

La W3C provee una herramienta para realizar las validaciones de HTML y CSS <http://validator.w3.org/>

Validate by URI: Es posible validar el HTML y CSS de una URI que ya exista utilizando la opción

Validate by file Upload: Si aún no se encuentra el recurso que quieres validar en internet, es posible cargarlo y aplicarle la validación haciendo uso de la opción

Validate by Direct input: En esta opción es posible validar una porción de código.

**MÉTODOS PARA APLICAR ESTILOS A LOS ELEMENTOS**

**Atributos a nivel de elemento HTML** (estilos en línea): se aplican como atributo dentro de una etiqueta ejemplo:

<P style= “color:red;Font-family:helvética”> aplica tipo de fuente y color a un párrafo

**Metodo a traves de Etiqueta style** : se ubica dentro del head antes del cierre, y se aplican a todos los elementos que se declaren dentro de la estructura de la etiqueta ejemplo:

<style>

H1{

Color:red;

Font-family: helvética;

Fon-size: 5px;

}

P{

Color: blue;

Font-family: helvética;

}

</style>

**Método para vincular un archivo CSS**

1. Agregar al directorio carpeta CSS donde se guardaran las hojas estilos
2. Crear el archivo con extensión CSS
3. Trasladar todos los estilos a el archivo
4. Vincular el archivo a la hoja de estilo usando la etiqueta link dentrol del head antes del cierre: <link rel= “stylesheet” href= “css/estilos.css” >

**Herencias de hojas de estilos:**

**Padres**: HTML

**hijos y Hermanos**: head y body (padres de elementos que los componen: header, nav, aside,footer)

Todo lo que se declare en elemento padre se heredan a todos los elementos que se encuentren en su interior, es decir heredan todas sus propiedades a menos que se especifique una propiedad diferente teniendo encuenta la jerarquía

**Hojas de Estilos en Cascadas**: posibilidad de definir cual estilo tiene prioridad jerárquica

**Orden de prioridad de mayor a menor**: ¡important¡, atributo, etiqueta y hoja de estilo

**Hoja de estilos del usuario:** Hojadeestilosdeusuario.css : cargar hojas de estilos en la configuración del navegador y tendrán prioridad sobre los estilos predeterminados del navegador

**Hoja de estilos del autor**: Estilos.css si hay varios vinculados en un mismo archivo tiene prioridad el ultimo

**Estilo prioritario** ¡important¡ ejemplo

<body> {

Font-family = verdana; ¡important¡

}

**SELECTORES DE ELEMENTOS EN CSS**

**Selectores de etiqueta**: body, footer, img, p ,article, div

**Selectores id**: id = “índice” - #indice{ } (busca y aplica estilo a un único elemento)

**Selector de clase**: class= “ItemIndice” - .itemIndice { } (aplica el mismo estilo a multiples elementos)

**Pseudo-Selectores**: permiten seleccionar cierta parte del contenido o seleccionar contenido cuando este se encuentre en un determinado estado <https://s3.amazonaws.com/nextu-content-production/Desarrollador_Web/02_CSS_Diseno_Web_Responsive/CheatSheet/WEB16S_C2U1L4_CheetSheet_Pseudoclases_V1.pdf>

**Pseudo-Clases**: permiten seleccionar un elemento cuando se encuentre en un estado determinado :focus (permite seleccionar el elemento que tenga el foco

**a:hover,**

Permite colocar de un color rojo oscuro cuando se pase el puntero por encima, cuando este activado o cuando tenga el foco

**a:active,**

**a:focus {**

**color: darkred;**

**text-decoration: none;**

**}**

Pseudo-Elemento: Permite seleccionar una parte de un elemento se emplean utilizando :: ejemplo ::after después de

* [::after](https://developer.mozilla.org/es/docs/Web/CSS/::after)
* [::before](https://developer.mozilla.org/es/docs/Web/CSS/::before)
* [::first-letter](https://developer.mozilla.org/es/docs/Web/CSS/::first-letter)
* [::first-line](https://developer.mozilla.org/es/docs/Web/CSS/::first-line)
* [::selection](https://developer.mozilla.org/es/docs/Web/CSS/::selection)
* [::backdrop](https://developer.mozilla.org/es/docs/Web/CSS/::backdrop)

**SELECTORES AVANZADOS**

**selectores de atributos:** realizan búsquedas a través de todos los atributos html5

a[href]selecciona todas las etiquetas **a** que tengan href

a[href=“http://www.google.com”] selecciona todas las etiquetas **a** que tengan la dirección google

img[src]

img[src$= ‘svg’ ] buscara svg al final del valor del atributo

img[src^= ‘svg’ ] buscara svg al principio del valor del atributo

img[src= ‘svg’ ] buscara svg al principio del valor del atributo

a[href^= ‘#’ ] buscara todos los vínculos a anclas o dentro de la misma pagina

\*{ } aplicara los cambios a todos los elementos del documento html

H1 + h2 { } aplicara los cambios a todos los elementos h2 que esten precedidos por un elemento

H2 ~ p { } aplica los cambio a todos los hijos directos es decir a toso los parrafos que tengan un titulo h2

Article> h2 { } aplica los cambios a todos los h2 des pues del articulo aunque se saklga es el punto de partida article

Ul li { } aplica los cambios a todos los li contenidos en el ul

**PROPIEDADES CSS**

<http://www.htmlhelp.com/es/reference/css/properties.html> (Listado de propiedades con descripción)

Font-family: Helvetica(tipo de fuente)

Font-size: 10px (tamaño de fuente)

Background-color: gray (color de fondo del body)

Margin-top: 20px (margen superior)

Border-radius: 10px(efecto redondeado a las cajas)

Display: block

Margin: 0 (aplica a todos los márgenes arriba, abajo, derecha o izq)

Margin: auto; centra el elemento de forma horizontal

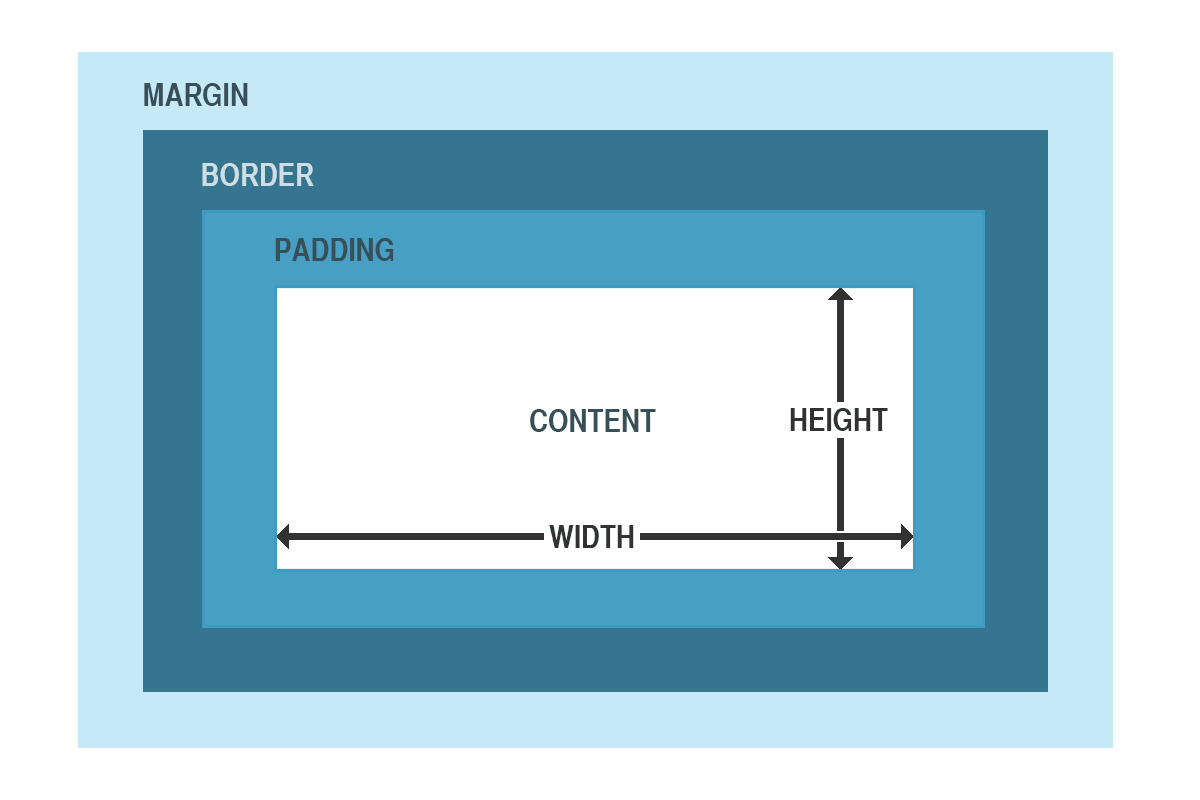
Margin-top: 10px (aplica margen a un lado especifico

border-top:solid 1px black (borde superior solido con línea de 1 pixel y color negro

padding-top: 10px; (margen del texto o contenido con relación a un área especifica fotter, nav, div )

text-align: center; (alineación del texto)

**Modelo de caja en CSS**



Marging: Es el espacio que hay entre la caja y otros elementos.

Border: es la línea que encierra la caja

Padding: es el espacio interior entre el contenido y el borde

Width: es el hancho de la caja

Height: es el alto de la caja

**Nota**: el tamaño de un elemento es la suma de margin + border + padding + width o height

Propiedades del modelo de caja:

Border 4px solid black (tamaño del borde, tipo, color)

**UNIDADES DE MEDIDAS**

**Unidades absolutas**: están completamente definidad y no dependen de otros valores:

in: pulgada 2.54cm

cm: centímetro

mm: milímetros

pt: puntos – equivale 0.35 mm

pc:picas equivale a 12pt - 4.23mm

Ventajas siempre tenemos valores fijos, desventajas no son eficientes para trabajar con el modelo mobile first o responsive desing

**Unidades Relativas**: dependen de otros valores de referencia para adaptarse y se expresa en %

Unidades Relativas a la Tipografia: dependen de tamaño de la tipografía (fuente o letra)

* em: se basa en el contenedor del elemento

Ejemplo:

<div style=”**font-size: 30px**”> <p id=”miParrafo”>Lorem ipsum…. </p> </div>

#miParrafo { width: 2em; // 1em = 30px, **2em = 60px** }

* rem: se basa en el tamaño de la raíz del documento HTML es decir el Body
* ex: relativa con respecto a la altura de la letra “x” minúscula del tipo padre.

Unidades relativas al viewpor(área visible del navegador)

vw: relativo al ancho del wiew port si es 1280px entones 1vw = a 1280px

vh: relativo al alto del wiew port si es 900px entones 1vh = a 900px

vmin: entre vw y vh toma el que tenga el menor valor

vmax: entre vw y vh toma el que tenga el mayor valor

**Propiedad Position**: permite ubicar un elemento con respecto a la pantalla estableciendo nuevos puntos de arranque relative, absolute, fixed, sticky ; static ( responsive design no se acompañan)

**Relative**: es la que está en la base atrás

**Absolute**: es la que se está en el frente y se basa al contenedor se acompaña con las posiciones top, left, right, botton;

**Propiedad Display**: Permite manejar el comportamiento de los elementos con respecto a su posición y como en caja con los otros elementos además de permitir ocultarlos o ponerlos opacos.

Display: inline; inserta un salto de línea

Display: inline-block; inserta un salto de línea antes y después del elemento

Display: none;

Opacity: 0;

Display: flex; permite distribuir las cajas de manera fluida y dinámica, de tal manera que los contenidos se ajusten a los cambios en el tamaño de la ventana; involucra un elemento padre o contenedor y los elementos hijos o ítem.

Flex-direction: row, row-reverse(horizontal), colum, column-reverse ( vertical)

Flex-wrap: wrap; permite usar varias línea

Flex-flow: row, wrap (distribución en filas y varias líneas)

Flex-grow: 2; determina como crecerán los elementos para ocupar el espacio del contenedor

Flex-basis: 40%

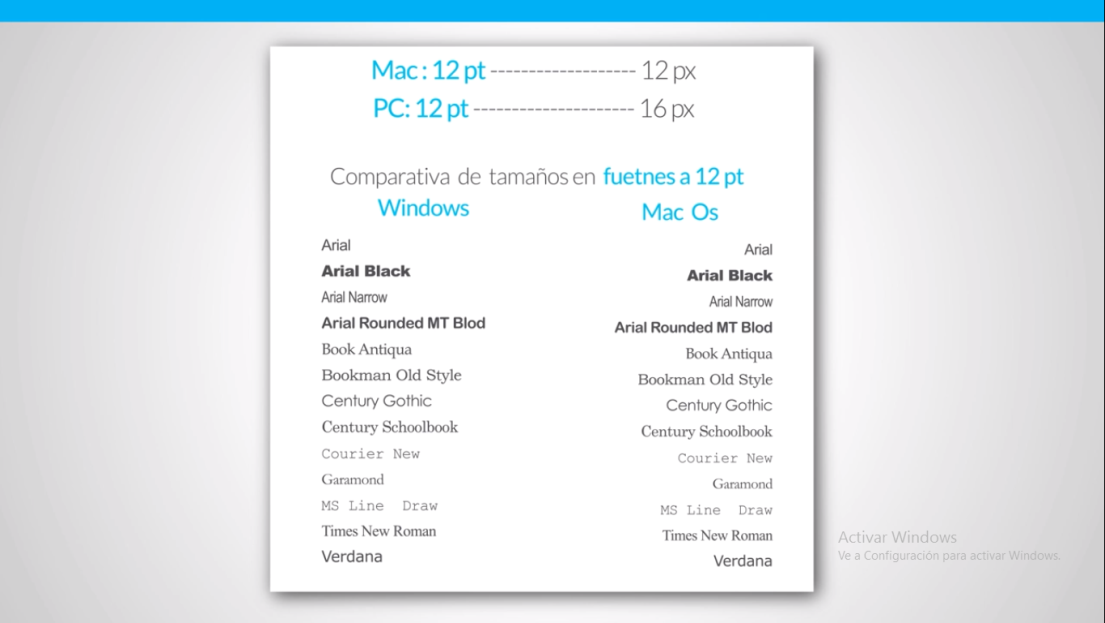
Flex: 1 2 30% (ajuste a toda l apantalla cada elemento 30% del ancho del contenedor ) combina tipo de distribución vertical u horizontla, si usara varias línea, y el % que tendrá cada elemento para ocupar el ancho de la pantalla row, wrap, basis.

Order: 1; permite colocar el número del orden del elemento

FORMATO DE TEXTO

El navegador utiliza las fuentes del computador donde se renderiza la pagina web y se corre el riesgo que el navegador no encuentre la fuente definida por lo que el navegador puede cambiar la fuente alterando el diseño de la pagina, existe un conjunto de tipografías estándares para internet arrial, Georgia y verdana, las fuentes para pc y para mac son diferentes a continuación una tabla equivalente para los dos sistemas operativos:

d**Font-family:** Para que se redenricen adecuadamente las fuentes minimizando la diferencia de renderizacion se utiliza la propiedad Font-family incluyendo una familia tipografía de pc y su equivalente en mac y además una fuente de respaldo por si ninguna funciona

**Font-size:**Permite cambiar el tamaño de las fuentes utilizando las diferentes medidas, lo mas conveniente es usar la media **em** que eqivale al tamaño por defecto de la fuente del navegador que usalmente es de 16px

Font-variant: smalls-caps (todo en mayus prim + gande)

Font-style: cursiva

Font-weight: en negrita

Las mayorías de fuentes tienen derechos de autor de tal manera que no se pueden usar libremente, existen proyectos de tipografía de uso libre

* open type: (otf)
* web open format (woff)
* open sans(google fonts)

**Fuentes Descargadas y agregadas al proyecto**: @font-face permite referenciar fuentes que se colocan en una carpeta del proyecto, las fuentes pueden ser tipo otf, woff, las cuales se pueden descargar de alguno de los repositorios disponibles como: http//ftp.gnu.org/gnu/freefont, se recomienda guardar las fuentes en la carpeta css y se aplica el estilo en el archivo css de la siguiente manera:

@font-face{

Font-family: ‘freesans’;

Src:url(‘FreeSans.otf’);

} Nota: *esta acción se debe realizar por cada fuente que se quiera usar*

Luego se aplica a la etiqueta deseada utilizando la propiedad font-family: “freesans”;

**Fuentes alojadas en un servidor**: se pueden referenciar fuentes que estén alojadas en un servidor a travez de un código que estos repositorios brindan, el mas popular es <https://fonts.google.com/> donde podemos escoger varias fuentes y aagregarlas a una sección donde posteriormente podemos analizarlas y general el código necesario para agregarlo al documento html en el head

Y en css asociamos la tipografía al selector que queramos definir esa propiedad. Ejemplo

HTML: <link href="https://fonts.googleapis.com/css?family=**Allerta+Stencil**" rel="stylesheet*"> google genra el código automatico cuando es una o mas fuentes al mismo tiempo*

CSS: font-family: 'Allerta Stencil', sans-serif;

Propiedades de texto y fuente

Text-aling: left, regth. Center. Justify

Text-indent: 2cm espacio de la primera línea

Text-decoration: under-line, over-line,line-through, none

Text-transform, opercase, lowercase, capitalize (mayúscula. Minuscala, primera en mayúscula)

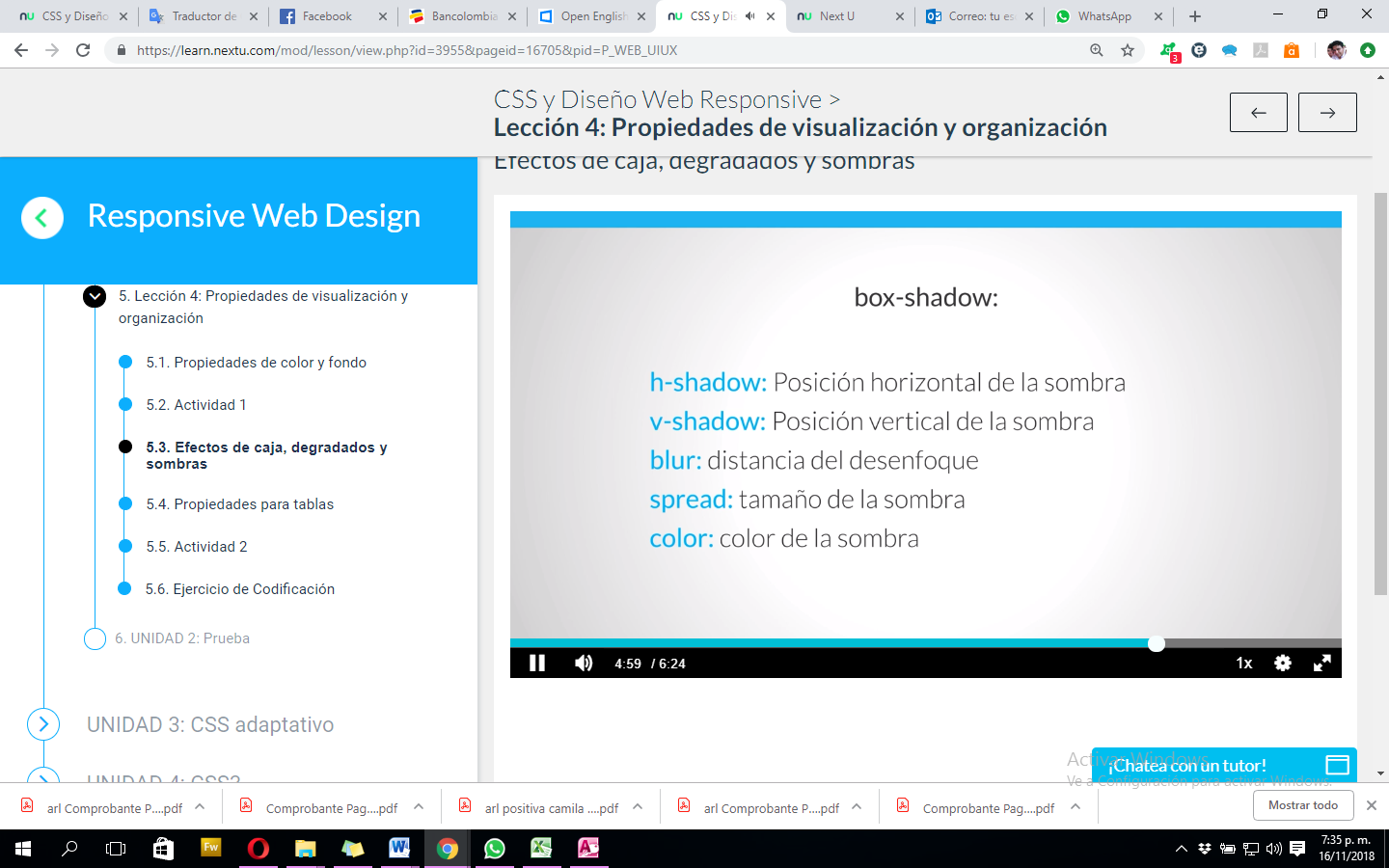
Letter-spacing: 5px; espacio entre letras

Word-spacing: 30px; espacio entre palabras

Color: red; color de texto

**Sombra en texto**

Text-shadow: 1px 2px 3px red; derecha, abajo, difuminacion y color (podemos usar varias sombras al tiempo colocando una coma después de cada grupo de valores de la sombra)

 Ejemplo:

P::selection{

Backgraund:green;

Color:blck;

Text-shadow:1px 1px #fff;

}

Varias sombras al tiempo se usa para dar efecto iluminado ej:

text-shadow: 1px 1px 2px black, 0 0 1em blue, 0 0 0.2em blue;

color: white;

COLORES DE TEXTO Y DE FONDO: se pueden definir usando el método rgb o hsl o hexadecimal

Color: rgb(51, 153, 204, 0.5); rojo, verde, azul, transparencia; números entre 0 menor intensidad y 255 mayor intensidad;

Background: hsl(0,0%,21%,0.5) matis(entre 0 y 360), saturación(%) e intensidad(%) transparencia

Exadecimal: #ffee24

**Degrades**:

background: linear-gradient(to bottom, #33ccff 0%, #ff99cc 100%);

background: radial-gradient(to bottom, #33ccff 0%, #ff99cc 100%);

Aplicación hsl colorpicker convierte un color exadecimal a hsla o rgba

<https://www.w3schools.com/colors/colors_picker.asp>

**Imágenes de fondo:**

background: url(imagenes/imagen.jpg) left top repeat,

url(imagenes/imagen.jpg) left top repeat

background-size: 150px, 300px; aquí se define el tamaño de las dos imagenes se definen segun el orden en que fueron creadas

Bordes de caja: efecto de bordes redondeados : este efecto no es compatible con todos los navegadores por eso hay que definir para cada navegador

Border-radius: 10px(efecto redondeado a las cajas)

(efecto redondeado a las cajas)

-moz-border-radius: 10px(mozilla)

-webkit-border-radius: 10px(zafary)

Se pueden definir un efectoi para cada borde en la misma línea: Border-radius: 10px 5px 15px 3px;

**box-sizing: border-box; (el tamaño definido para la caja incluye el borde y el margen)**

**DISEÑO WEB ADAPTATIVO**

para lograr que un proyecto web sea adaptativo debemos determinar el viewport o el área de la pagina web que será visible por el usuario y definir un conjunto de condiciones (media queries) y reglas css3 de acuerdo con cada tamaño de pantalla, esto estableciendo las características de cada elemento para que puedan verse correctamente.

VIEWPORT : hace referencia al área disponible para renderizar una pagina web es decir el área visible del navegador.

Se configura dentro de la etiqueta meta y se pueden configurar los siguientes valores:

* nivel de escalado que se puede realizar
* zoom inicial que debe realizar el navegador
* el área disponible para renderizar

<meta name="viewport" content="width=device-width, user-scalable=no,initial-scale=1.0>

width=device-width: Área disponible para renderizar, en este caso todo el ancho que tenga el viewport del dispositivo.

user-scalable=no: El usuario no puede hacer escala.

initial-scale=1.0: Nivel de zoom/escala Inicial.

Esta es la definición oficial para que tu página web se adapte de forma correcta a los móviles.

RESOLUCIONES DE PANTALLAS SEGÚN LOS DISPOOSITIVOS



Etiqueta <Picture>

Permite seleccionar una imagen según una condición de tamaño de display o dispositivo

<picture>

<source media = “(mind-width:768px) srcset = “img/banner.png”>

<source media = “(mind-width:480px) srcset = “img/gato.png”>

<img src= “img/pajaro.jpg”/>

</picture>

**MEDIA QUERIES**

Son porciones de código que se ejecutan cuando se cumple una condicion basada en tamaños de pantallas o orientación o dispositivos de visualización los cuales podemos agrupar en:

print - Vista de Impresión.

min-width - Tamaño mínimo en horizontal.

max-width - Tamaño máximo en horizontal.

min-height - Tamaño mínimo en vertical.

max-height - Tamaño máximo en vertical.

orientation - Posición en la que se encuentre el dispositivo.

resolution - Tamaño ocupado por el browser en píxeles.

color - Detecta si la pantalla en que se visualiza el sitio es a color o en blanco y negro.

light-level - Densidad de la luz del dispositivo.

@media print{

}

@media screen{

}

@media max-width:420px{

}

@media (min-didth:421px) and (max-width:720px){

}

Dependiendo de los media types deben crearse estilos individuales que se ajusten a las condiciones detectadas. El dispositivo renderizará los estilos creados específicamente para cada condición.

**TRANSFORMACIONES**

Propiedad Transform: permite girar el objeto en los ejes x y z

Transform: rotatex(180deg)

Transform: rotatey(180deg)

Transform: rotatez(180deg)

Propieidad scale: permite cambiar el tamaño del objeto

Transform sclae(1,2) se multiplica el ancho x 1 y el alto x2

Transform sclae(0.5,0.5) reduce el tamaño a la mitad

Propiedad perspective: permite definir distancia entre el objeto y la persona que observa

Transform-style:perspective-3d; (se puede definir al contenedor o al elemento)

Aplicado al elemento:

Transform: perspective(800px)

Rotatex(45deg)

Rotatey(-10deg); (el punto y coma se coloca al final de todo)

Aplicado al contenedor padre:

.contenedor{Perspective: 250px;

Perspective-origin: botton;} (se debe definir rotaciones a el elemento )

#elemento{t ransform:rotatex(-45deg) rotatey(10deg);}

Propiedad translate: permite mover los elementos basados en eje x y teniendo encuenta el orden horizontal y vertical y que los positivo trasladan derecha abajo y y los negativos izquierda y arriba

Transform: translate (80px, 100px)

Transform: translate (-10px, -35px)

Transform: translatey (80px)

Transform: translatey (-10px)

Transform: translatex (80px)

Transform: translatex (-10px)

Transform: translatex (none) regresa a su posición original

Transform: translatez (80px) adelante (necesita predefinir perspectiva)

Transform: translatez (-10px) atrás (necesita predefinir perspectiva)

Propiedad skew: permite inclinar los elementos

Transform: skew(30deg, 40deg);

Transform: skew(0deg, 40deg);

Transform: skewx(30deg);

Transform: skew(30deg, 0deg);

Transform: skewy(40deg);

**TRANSICIONES**

Propiedad transition: permite colocar una duracion entre los cambios de propiedad como color, altura y anchura.

#elemento{ backgraound-color: blue;

Transition: backgraund 1s;}

#elemento:hover{ backgraound-color: blue;}

#elemento{ backgraound-color: blue;

Transition-property: background, colors;}

Transition-duration: 1s, 2s;

#elemento:hover{ backgraound-color: blue;}

**ANIMACIONES**

@-webkit-keyframes **nombre**{

Una vez definida la animación se debe ir al elemento y colocarle la animación de la siguiente manera:

Animation: **nombre** 10s;

Se puede colocar el parámetro infinity para que la animación se repita indefinidamente.

Animation: **nombre** 10s infinity;

From{

Transform: rotate(45deg);

}

To{

Heigth: 0;

}

}

También se puede definir animaciones en tiempos específicos definidos en porcentajes:

@-webkit-keyframes nombre{

0%{

Opacity: 0;

}

25%{

Opacity: 0.25;

: rigth;

}

50%{

Opacity: 0.5;

Float: left;

}

75%{

Opacity: 0.75;

Float: rigth;

}

100%{

Opacity: 1;

Float: left;

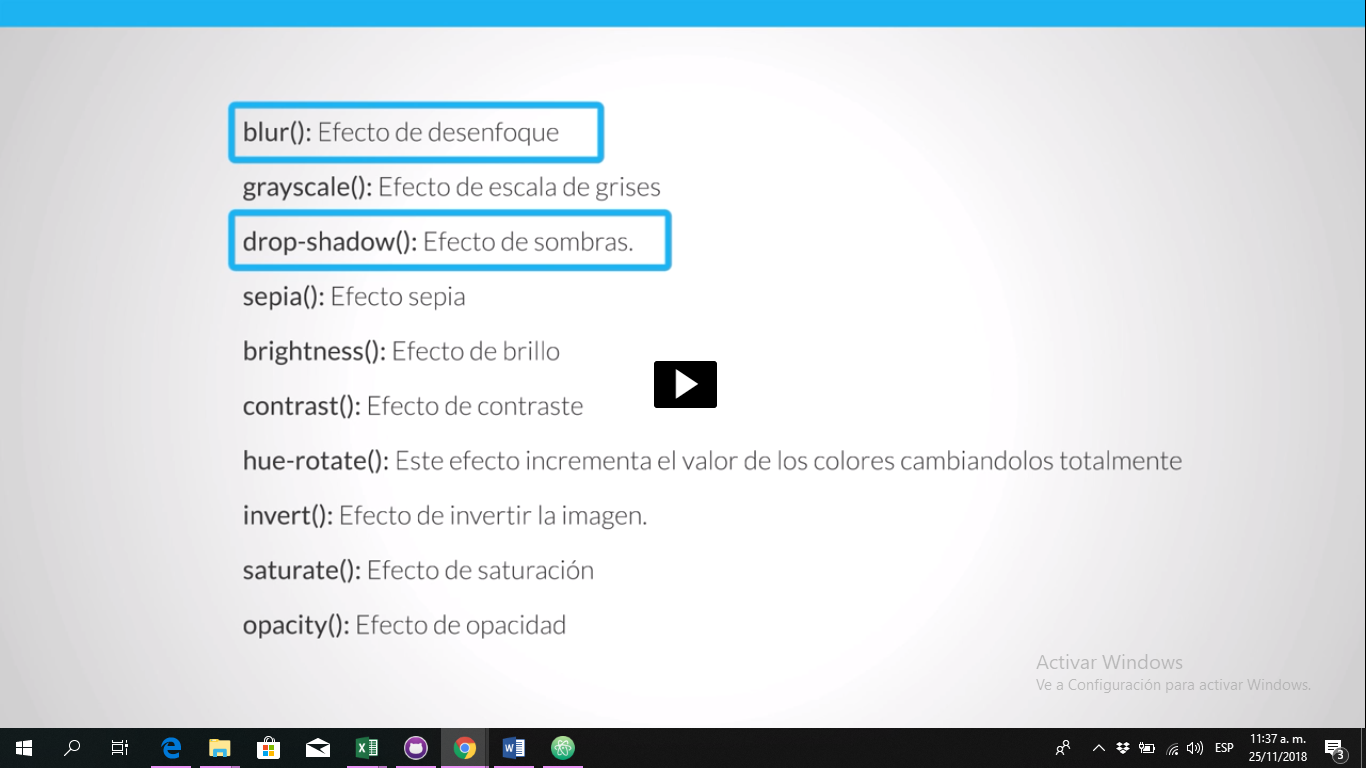
}

}

Animation-delay:1sg; (especifica el tiempo de retardo antes que comience la animación

FILTROS DE IMAGEN EN CSS3

Filter: grayscale(80%);



Todos reciben valores en % menos los seleccionados ya que reciben valores en px

Filter: Blur(2px);

Filter: drop-shadow(4px 3px 4px 5px blue;

BOOTSTRAP

<http://getbootstrap.com/>

Bootstrap es un framework es decir un conjunto de librerias css creado por twitter para facilitarnos y estandarizar los desarrollos de aplicaciones web, trae bastantes aplicaciones en su interior que nos permiten mejorar la presentación del cuerpo, formularios y botones de navegación. Su gran particularidad es que ofrece un diseño adaptativo para multiples plataformas también ofrece integración de Javascript.

Ventajas:

* maneja modelo multiplataformas responsive
* Permite optimizar tiempo de desarrollo
* Facilita modificar la estructura de nuestro sitio web caso de forma instantánea
* Incluye el código de compatibilidad para los deferentes navegadores que existen
* Maneja la filisofia mobile first
* Se puede usar online o ofline:

**Modo online:** se debe pegar este código en el head:

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO" crossorigin="anonymous">

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js" integrity="sha384-ChfqqxuZUCnJSK3+MXmPNIyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy" crossorigin="anonymous"></script>

**Modo ofline** de sebe descargar la librería y colocarlo en la carpeta CSS del proyecto vinculándolo a través de <link rel="stylesheet" href="css/bootstrap.min.css" media= “screen”>

**Clases a manejar con Bootstrap**

Class= “container” crea un contenedor sencillo que se amolde al tamaño de toda la pantalla

Class= “row” crea un contenedor con varias columnas en el que se debe usar el sistema de rejillas para ajustar los contenidos teniendo en cuenta que no se debe superar las 12 rejillas y que el total debe ser siempre de 12

<section>

<div class="row">

<div class="col-lg-4" style="background-color:blue">column 1</div>

<div class="col-lg-4" style="background-color:yellow">column 2</div>

<div class="col-lg-4" style="background-color:green">column 3</div>

</div>

</section>

Class= “col-lg-4” permite crear una columna y especificando su ancho teniendo en cuenta el sistema de rejillas su aplicacion se puede definir para diversos tipos de pantallas de la sig manera:

Class= “col-lg-4” aplica para pantallas grandes

Class= “col-md-4” aplica para pantallas medianas

Class= “col-sm-4” aplica para pantallas pequeñas

Class= “col-xs-4” aplica para pantallas muy pequeños

**column reset**: es establecer el comportamiento de las columnas cada vez que cambie el tamaño de la pantalla

<section>

<div class="row">

<div class="col-lg-10 col-md-8 col-sm-7 col-xs-6" style="background-color:red">column 3</div>

<div class="col-lg-2 col-md-4 col-sm-5 col-xs-6" style="background-color:black">column 1</div>

</div>

</section>

**Column offset**: Establece un espacio en blanco antes de colocar la columna

<section>

<div class="row">

<div class="col-lg-4 style="background-color:red">column 1</div>

<div class="col-lg-4 col-offseting-4 style="background-color:red">column 1</div> </div>

</section>

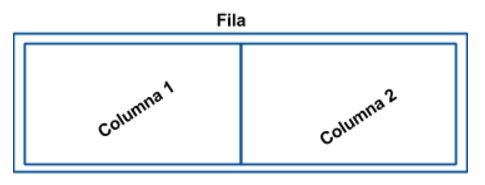
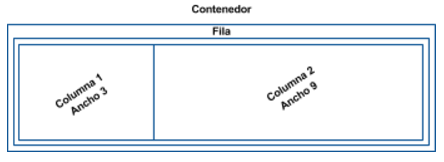
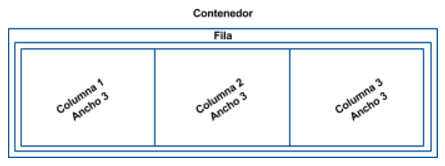
Column push: permite desplazar una columna a la derecha

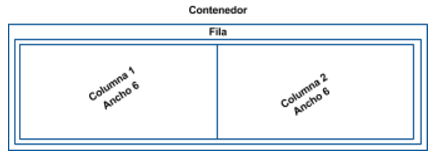
<div class="col-md-4 col-md-push-4 style="background-color:red">column 1</div>

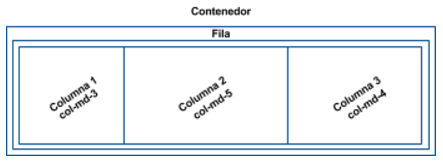
Column pull: permite desplazar una columna a la izquierda

<div class="col-md-4 col-md-pull-4 style="background-color:red">column 1</div>

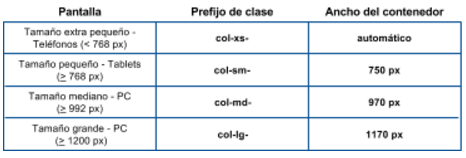
OBSERVACIONES PARA TRABAJAR CON BOOTSTRAP

* Las columnas siempre deben estar contenidas en las filas.  
  
* La etiqueta HTML que representa una fila debe tener asignada la clase “**row**”.
* Las filas deben estar contenidas dentro de una etiqueta y ésta debe tener asignada la clase “**container**”.
* Cada fila se puede dividir hasta en 12 columnas como máximo, pero hay que tener presente que somos nosotros quienes definimos el número de columnas en el que deseamos dividir cada una de las filas y su respectivo tamaño para el ancho de cada pantalla. Por ejemplo:



* El tamaño de las columnas se definirá con las clases que Bootstrap define para el tamaño de la pantalla de cada dispositivo, por ejemplo **col-md-xx**, teniendo a **xx** como el tamaño de la columna que puede tomar valores de 1 a 12.  
  

A continuación se indica el comportamiento del sistema de rejilla según el tamaño de la pantalla del dispositivo y que clase CSS podemos controlarlos:



Podemos asignar múltiples clases bootstrap a la columna para determinar su tamaño en los diferentes dispositivos.  
https://s3.amazonaws.com/nextu-content-production/Desarrollador_Web/03_Frameworks_y_Librerias_CSS/Ejercicios/WEB16S_C3_U2_L2_Ejercicio1_1/img7.png

**CLASES DE BOOTSTRAP**

Según vimos en la lección anterior, el manejo de los textos e imágenes que utiliza Bootstrap CSS nos permite ahorrarnos estilos y propiedades en una hoja de estilos. Con sólo adicionar las respectivas clases en nuestro HTML, podemos obtener la alineación de texto que necesitamos, ya sea a la derecha, a la izquierda, al centro o justificado. Adicionalmente, podemos cambiar un texto a mayúscula o minúscula sostenida, mayúscula inicial, entre otros. En cuanto a la imágenes, podemos, por ejemplo, hacerlas redondas o con las puntas redondeadas. Recordemos un poco los conceptos.

**Clases de BootsTrap CSS para cajas:**

Container: la caja se centra en la pantalla amoldándose al redimencionar

Container-fluid: la caja se amolda a todo el ancho de la pantalla

Row: organiza en columna todo su contenido debe usarse sistema de rejilla de 12 columnas

Navbar: Esta barra se adaptará al tamaño de pantalla, mostrando los elementos colapsados en un botón en pantallas pequeñas y de forma normal para pantallas más grandes.

Este elemento requiere que el plugin de JavaScript de Bootstrap esté incluido. Además, para cumplir con las reglas de accesibilidad se recomienda añadir role="navigation" a nuestras barras de navegación. <nav class="navbar navbar-default" role="navigation">

Navbar-inverse: invierte los colores de la barra de navegacion

Navbar-fixed-top: fija la barra en la parte superior junto a las clases navbar navbar-default.

Affix-top:fija barra de navegacion a la izquierda y arriba

**Clases de Bootstrap CSS para texto:**

text-left: Alinear el texto a la izquierda.

text-center: Alinear el texto al centro.

text-right: Alinear el texto a la derecha.

text-justify: Justificar el texto.

text-uppercase: Convertir el texto a mayúsculas.

text-lowercase: Convertir el texto a minúsculas.

text-capitalize: Convertir las iniciales del texto a mayúsculas.

**Etiquetas especiales de Bootstrap CSS para textos:**

<abbr>: Abreviar texto. Es importante agregar el atributo title, para que esta funcione.

<mark>: Marcar o resaltar texto.

<code>: Resaltar porciones o líneas de código.

<kbd>: Resaltar las teclas de acceso directo.

**Clases y etiquetas para el manejo de listas:**

<dl>: Abrir o iniciar una lista por definición.

<dt>: Primera etiqueta que se asigna después de abrir la lista por definición. En ésta se podría decir que va el término o palabra que se desea definir o describir.

<dd>: Etiqueta que va después de la etiqueta dt. En ésta se pone la definición o descripción del término o palabra que se puso en el dt.

list-unstyled: Clase utilizada para quitar la numeración o viñetas de una lista.

list-inline: Clase utilizada para mostrar las listas de manera horizontal.

dl-horizontal: Clase utilizada para mostrar las listas por definición de manera horizontal.

**Clases Bootstrap para imágenes:**

img-rounded: Redondea las esquinas a 16 píxeles.

img-circle: Redondea las esquinas con un radio de 50%.

img-thumbnail: Pone un marco alrededor de la imagen.

**Clases bootstrap para formularios**

Role=”form” atributo que ayuda a leer el formulario para personas discapacitadas

Form-group : permite agrupar los elementos del formulario (etiqueta y campo) en un contenedor independiente y facilitar que se expandan al 100% en horizontal dentro de su contenedor,

Form-control: hace que los elementos del formulario ocupen el 100% de su contenedor

Btn btn-default: coloca un aspecto mas agradable a los botones

Checkbox: agrupa y organiza de forma vertical los elementos check (crear otro contenedor)

Checkbox-inline: agrupa y organiza de forma horizontal los elementos check

Radio: agrupa y organiza de forma vertical los elementos radio

Radio-inline: agrupa y organiza de forma horizontal los elementos radio

<div class=”form-grup form-control”>

<div clas= “checkbox” >

<label for=”hobbie3”>videojuegos</label>

<input type=”checkbox” name=”hobbie3” id=”hobbie3”></input>

</div>

</div>

Listas desplegables: también se le agrega la clase form-control no se necesita encerrar en un div ya que lo encierra un select, permite agregar atributo **multiple** para seleccionar varias opciones al tiempo.

<select **multiple** class=”form-control” name=””>

<option value=””>seleccion 1</option>

<option value=””>seleccion 2</option>

<option value=””>seleccion 3</option>

<option value=””>seleccion 4</option>

</select>

**FRAMWORKS FUNDATION**

**ZURB**

Esta construido en HTML CSS y JAVAscript permite desarrollar código mas rápido y es compatible con cualquier tecnologis Back-end

* 50% reducción de código
* Componentes pensados para la accesibilidad
* Menos cantidad de estilos para sobreescribir
* Sistema de rejillas personalizables
* Herramientas de prototipado ZURB
* Movimiento en la interfaz del usuario y patrones de diseño mas flexibles
* Único framework soportado por una organización con 5 años de trayectoria

PASOS PARA DESCARGAR EL FRAMEWORK

1. Acceder a la página oficial [www.fundation.zurb.com](http://www.fundation.zurb.com)
2. Clic en download Se puede descargar esencial, completa o personalizada
3. Si se escoge personalizado desmarcar la opción ALL, definir opciones e ir al final y clic en descargar
4. Antes de descargar se debe definir el sistema de rejillas a utilizar grid o flexgrid

**Grid**: se deben definir los tamaños e las columnas que sumen 12 si son menos fundation dejara un espacio en blanco que se puede colocar al final con la opción **end**

**Flex gGrid**: no es necesario definir los tamaños de columna ya que el sistema es flexible, si no se definen fundation los divide por partes iguales si se definen algunos si y otros no fundación amplía el tamaño de los que no se les define tamaño para que ocupen el espacio restante.

USANDO FUNDATION:

Se puede empezar desde cero o importar el framework en una página existente

1. Descomprimir el archivo descargado
2. realizar una copia de la carpeta
3. Cambiar el nombre
4. Ubicar en el directorio de proyectos web

Si el proyecto esta creado y queremos importar

Se copian los archivos de las carpetas css y jss en el directorio de nuestro proyecto

**Enlazar las hojas de estilos**

<Link rel=“stilesheet” href=“css/fundation.min.css” >

<Link rel=“stilesheet” href=“css/app.css” >

Incluir el boque de codigo para js que podemos encontrar en el archive index.html del proyecto descargado

**Hay 3 archivos en la carpeta css**

App.css : archive para colocar nuestros estilos reemplaza a estilos.css

Fundation.min.css: estilos de foundation que ocupan minimo espacio

Fundation.css: hoja de estilos de foundation

**CLASES DE FUNDATION**

**Opciones para el sistema de rejillas Grid**

Grid-container: contenerdor para el sistema de rejillas con márgenes a los lados

Grid-fluid: contendor sin márgenes (ocupa 100% ancho pantalla o contenedor)

Row: clase contenedora para columnas

Callout: clase contenedora que contiene borde redondeado por defecto

Column o colums: clase para definir columnas del contenido

<div Class= “row”>

<div Class= “column small- 4 medium-3 large-7”> </div>

<div Class= “column small- 4 medium-6 large-3”> </div>

<div Class= “column small- 4 medium-3 large-2”> </div>

</div>

**Small**: cualquier pantalla generalmente orientado a Smartphone

**Médium**: 640 px o mas destinado a tablets

**Large**: 1024 px o mas computadores de escritorio

Row column: clase que permite crear una columna que ocupara toda la fila pero con márgenes

Expanded row: clase que permite crear una columna que ocupe toda la pantalla sin dejar márgenes

Large-offset-3: clase que mueve rejillas a la derecha

End: clase que se le coloca a la última columna cuando las rejillas suman menos de 12 para que la última columna no se coloque a la izquierda dejando un espacio, con la opción end el espacio quedaría después de la columna que se le coloco esta clase

Collapse: no hay espacios las columnas se mantienen horizontal

Uncollapse: se activan espacios las columnas se organizan en vertical

Centered: clase que centra una columna que no ocupe todo el espacio

Uncentered: clase que quita el centrado de una columna

Push: derecha small-push-4

Pull: mover hacia la izquierda small-pull-4

Order: permite definir el orden de las columnas, ideal para los cambios de tamaños de pantalla

Align-justify: justifica las columnas colocando espacio entre ellas

Align-center: centra las columnas

Align-rigth: columnas a la derecha

Align-left: columnas a la izquierda

Align-spaced: coloca espacios iguales entre columnas incluyendo los bordes

Align-top: alinea hacia la parte superior

Align-middle: centra las columnas de forma vertical

Align-botton: alinea hacia la parte superior

**Opciones cuando se descarga el sistema de rejillas Flex-Grid**

Shrink: define que el ancho de una columna se ajuste al contenido

Expand: permite colocar las columnas en vertical medium-expand

Stack: coloca las columnas en vertical

Unstack: columnas en horizontal

**Opciones cuando se descarga el sistema de rejillas Grid-xy**

Grid-container: contenedor con márgenes

Grid-fliud: contendor sin márgenes (ocupa 100% ancho pantalla o contenedor)

Grid-x: organiza las columnas horizontal

Grid-y: organiza las columnas en vertical

Grid-padding-x: coloca un padding a las columnas ya sea x o y

Grid-margin-x: espacio entre columnas

Cell: coloca los contenedores en horizontal cuando no se define la opción small

**Elementos de formulario en foundation**

Cada elemento debe estar dentro de la etiqueta label

<label for="">

<input type="text" placeholder="nombre" required>

</label>

De esta manera foundation identifica los diferentes tipos de elementos y agrega las ayudas y diseños correspondientes dependiendo el tipo de dato

los botones de selección debe colocarse dentro de etiquetas fielsetcon la siguiente estructura

<fieldset class="fieldset”>

<legend>Radio Buton</legend>

<input type="radio" name="color" value="rojo" id="rojo" required><label for="rojo">Rojo</label>

<input type="radio" name="color" value="azul" id="azul"> <label for="azul">Azul</label>

</fieldset>

Nota: la clase fieldset mejora la presentación de los elementos que van agrupados como radio, butom y chackbox