/\*

Formas de escribir selectores

============================================================================

Selector de elemento

Selecciona uno o múltiples elementos, dependiendo de cómo se conforme el selector

Seleccionará un elemento de acuerdo a su etiqueta.

p{

    color:blue;

}

============================================================================

Selector de clase

Son reutilizables e inician con un "punto"

.cliente{

    color:blue;

}

============================================================================

Selector de ID

Se pueden tener ID's ppor pagina pero no pueden repetirse

# cliente{

    color:blue;

}

============================================================================

Selectro de atributo

Selecciona elementos basados en algún elemento que tenga.

[src="logo.jpg"]{

    color:blue;

}

============================================================================

Combinaciones de Descendentes

Selecciona los elementos hijos cuyo padre sea una clase o ID en específico

.cliente .nombre{

    color:blue;

}

============================================================================

Todos los hijos

Aplica la siguiente regla a todos los  párrafos hijos

.cliente >p{

    color:blue;

}

NOTAS:

    SE RECOMIENDA SIEMPRE ESCRIBIR CÓDIGO CSS CON CLASES

    NO ES MUY RECOMENDABLE ESCRIBIR ID'S PORQUE AFECTAN ALGO LLAMADO ESPECIFICIDAD

    SE PUEDEN TENER MÚLTIPLES ID'S POR DOCUMENTO PERO NO CON EL MISMO NOMBRE!!!

    NO EMPEZAR UN SELECTOR CON UN NÚMERO

    ==============================================================================

    ESPECIFICIDAD EN CSS

    ==============================================================================

    Esto refiere a, ¿qué valores serán tomados en cuenta por el navegador?

    CÓMO EL NAVEGADOR TOMA EN CUENTA EL CSS EN FUNCIÓN DE QUÉ TAN ESPECÍFICO SEA EL SELECTOR QUE HAYAMOS

    CREADO.

    ////////////////////////////////////////////////////////////////////////////////

    Ejemplo 1: Misma especificidad.

    p {

        color:blue;

    }

    p{

        color:red;

    }

    Como tienenen la misma especificidad, se aplica el color rojo porque está al final

    del documento

    ////////////////////////////////////////////////////////////////////////////////

    Ejemplo 2: Misma especificidad.

    p.parrafo {

        color:blue;

    }

    p{

        color:red;

    }

    Como tiene más especificidad el primer selector, se aplicará el color azul y NO el

    color rojo.

    ////////////////////////////////////////////////////////////////////////////////

        Ejemplo 2: Misma especificidad.

    p#parrafo {

        color:blue;

    }

    p{

        color:red!important;

    }

    Como el segundo selector tiene la bandera !important, le pasa por encima a todos.

    --------------------------------------------------------------------------------

    Nota final: si un elemento tiene un selector más específico, no importa mucho

    donde haya sido declarado, CSS decidirá por su especificidad.

    --------------------------------------------------------------------------------

    ////////////////////////////////////////////////////////////////////////////////

    SIEMPRE Y CUANDO SE TRATE DE UN MISMO SELECTOR, COMENZARÁ A APLICAR LOS ESTILOS DE ABAJO HACIA ARRIBA

    Y NO DE ARRIBA HACIA ABAJO

    ENTRE MÁS ESPECÍFICO SEA UN SELECTOR, ES MÁS PROBABLE QUE SE APLIQUE, NO IMPORTANDO EN QUÉ PARTE DEL

    DOCUMENTO SE APLIQUE

    \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

     =========     =======                ========     ======     ======    =======

    ||           ||       ||  ||        ||       ||  ||      ||  ||        ||

    ||           ||       ||  ||        ||       ||  ||\_\_\_\_\_\_||  ||====    ||======

    ||           ||       ||  ||        ||       ||  || ||\_\_     ||               ||

    ||\_\_\_\_\_\_\_\_   ||\_\_\_\_\_\_\_||  ||\_\_\_\_\_\_  ||\_\_\_\_\_\_\_||  ||    ||    ||======   \_\_\_\_\_\_||

     ========   ========    ========

    ||         ||          ||

    ||         ||\_\_\_\_\_\_    ||\_\_\_\_\_\_\_

    ||                 ||          ||

    ||\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_||   \_\_\_\_\_\_\_||

    \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

    Se les puede decir a sus colores

    Funciona en cualquier navegador moderno.

    -por nombre

        color:blue;

        color:black;

        color:cornflowerblue

    -código hexadecimal (Tiene soporte a transparencia)

        HEX #000;

        #000000;

    -sistema rgb()

    -sistema hsl()

    \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

    D I S P L A Y

    \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

    Hay 2 tipos: block e inline

    display:block;

        El elemento se colocará uno debajo del otro sin importar

        su tamaño o contenido

    display:inline;

        Significa que el elemento se posicionará a la derecha una

        vez que haya tomado el espacio que requiere

    Ningún elemento tiene

        display:flex; -->Activa otras características

        display:grid;

    Por default

    No es necesario especificar un display por cada elemento, YA QUE POR DEFAULT PONE BLOCK.

     \_ \_ \_ \_ \_ \_ \_ \_\_ \_ \_ \_\_ \_ \_ \_\_ \_ \_ \_\_ \_ \_ \_\_ \_ \_ \_\_ \_ \_ \_\_ \_ \_ \_\_ \_ \_ \_\_ \_ \_ \_\_ \_ \_ \_\_ \_ \_ \_

    |>FLEXBOX

    |Fue diseñado como un modelo unidimensional para crear layouts.

    |Fue especialmente diseñado para alinear los elementos en tus diseños.

    |No añade efectos de animación, ni textos, es una tecnología utilizada únicamente para los

    |layouts y sustituye a los floats o table-cell.

    |\_ \_ \_ \_ \_ \_ \_ \_\_ \_ \_ \_\_ \_ \_ \_\_ \_ \_ \_\_ \_ \_ \_\_ \_ \_ \_\_ \_ \_ \_\_ \_ \_ \_\_ \_ \_ \_\_ \_ \_ \_\_ \_ \_ \_\_ \_ \_ \_

    Tiene dos ejes para distribuir elementos en una sola dirección:

        -fila (izq a derecha)

        -columna (arriba hacia abajo)

    \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

    F O R M A S   D E   E S C R I B I R   C S S

    \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

    Existen diferentes formas de escribir código CSS: BEM, Utility First o Módulo.

    Si el proyecto tiende a ser grande, es conveniente usar cualquiera de los 3.

    BEM - Bloques, elementos y modificadores

        .card{}

        .card\_\_titulo{}

        .card\_\_imagen{}

        .card\_\_boton{}

        .card\_\_boton--activo{}

    Utility First - Se crean clases con una sola propiedad, que describen qué es lo que se haría

        Básicamente, lo que se tiene es una propiedad y un valor por clase.

        .text-center{}

        .color-red-100{}

        .bg-blue-200{}

        -.p-2{}

        -.m-2{}

        Módulos

            Se define el contenido principal y después va seleccionando cada uno de los

            elementos en html.

            Se define una clase padre y después se va seleccionando todas las etiquetas del código html.

    \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

    R E S P O N S I V E    W E B    D E S I G N

    \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

    La solución para evitar crear múltiples sitios web y que sean adaptables a una gran cantidad

    de dispositivos es: Responsive Web Design con Media Queries.

    ¿Qué es?

    Este enfoque nos dice que nuestros diseños se deberán adaptar a las interacciones del usuario

    y a la resolución utilizada.

    Los sitios web deberán adaptarse a celulares, tabletas, laptops, computadoras de escritorio,

    televisores.

    ¿Cómo se logra esto?

    Con Media Queries

    A continuación, tamaños recomendados:

    --768px se activa con pantalla de computadora o televisión

    @media (min-width:768px){

    }

    --468px, se activa con teléfono

    @media (min-width:468px){

    }

    --Computadoras de escritorio o laptops o televisión

    @media (min-width:1444px){

    }

    @media (min-width:1440px){

    }

    NOTA IMPORTANTE!!!!

    Para probar diferentes tamaños, se recomienda descargar: ResponsivelyApp,

    permite simiular distintos tamaños y es de uso gratuito.

    \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

    C S S   B O X   M O D E L

    \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

    El tamaño de lo que se muestra en pantalla, está delimitado por 4 cosas:

        -tamaño de contenido

        -tamaño de relleno (padding)

        -tamaño del borde

        -tamaño del margen

    Todo en CSS es una caja, pero la forma o características de la caja, dependerá de

    4 cosas:

    \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

    C S S    G R I D

    \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

    CSS Grid te permite definir la ubicación y tamaño de los elementos de tu sitio

    WEB.

    \*/