Página web complementaria: https://www.w3schools.com/css/default.asp

1. Introducción Universidad Desarrollo Web

Página web complementaria: https://www.w3schools.com/css/default.asp

1. Introducción a CSS

CSS nos va a permitir mejorar visualmente nuestra paginas HTML. Por un lado, HTML tiene como objetivo agregar la estructura de nuestra página web, por otro lado, CSS va a mejorar visualmente estas etiquetas HTML.

2. Formas de aplicar estilos CSS

Como vimos en el curso de HTML podemos mejorar visualmente las etiquetas HTML directamente dentro de nuestro archivo HTML con la etiqueta style (Internal CSS), también podemos aplicarlo dentro de nuestros tags o, como una buena práctica, utilizar un archivo dedicado a los estilos css y vincularlos con el atributo id (normalmente utilizado para realizar bookmarks o a un solo elemento de nuestra página) o el atributo class.

Los estilos que toman prioridad son los del último scope que tome el CSS, es decir, del scope más general (un archivo CSS -> External CSS) al scope local (CSS aplicado al tag -> Inline CSS). Lo mismo ocurre dentro del estilo CSS, si escribí una etiqueta, por ejemplo, h1 dos veces, va a tomar la última coincidencia. -> Esto ocurre con los ATRIBUTOS que sobrescriba, no con la etiqueta en si (si tengo font-size y color en general y un scope local solo con color aplica el font-size y sobrescribe el color).

Con el atributo !important después del estilo declarado podemos romper esta relación de scopes (ej: color:aqua !important).

3. Selector ID en CSS

El selector ID nos va a permitir modificar con CSS un elemento HTML (en realidad puede funcionar como una clase, pero no es lo recomendable) que es único en nuestra página. (por ejemplo, un único h1).

El nombre del id inicia con un #.

4. Clases en CSS

Si queremos aplicar CSS a uno o más elementos HTML utilizaremos el concepto de clases en CSS. Podemos aplicar varias clases a elementos HTML simplemente separando con espacios los nombres dentro del atributo class.

El nombre de una clase inicia con un . y no puede contener caracteres numéricos al comienzo del nombre.

5. Selector universal en CSS

Si queremos aplicar algún estilo CSS a toda nuestra página HTML tenemos un selector universal.

Este selector se identifica con el carácter *.

6. Agrupar selectores en CSS

Vamos a ver como agrupar elementos HTML si vamos a aplicar los mismos estilos a varios elementos HTML o también clases o selectores. Podemos combinarlos y agruparlos.

Ejemplo: h2, h3{//contenido}

7. Subclases en CSS

p.centrar{} -> Primero se seleccionó el elemento y posteriormente le aplico el estilo CSS que va a ser UNICO para ese elemento que seleccione. La clase será solamente reutilizable para esa clase que especifique al comienzo.

2. Colores, Bordes y más en CSS

1. Colores por nombre en CSS

Tenemos 140 colores definidos por nombres en HTML (https://htmlcolorcodes.com/es/nombres-de-los-colores/).

También podemos agregar colores con código Hexadecimal o RGB.

Puedo agrupar clases para que se aplique un mismo estilo poniendo el nombre de la clase, separado con una coma. Ej:

```
.rojo(clase), .rosa(clase), .violeta(clase){
    color: white;
}
```

2. Colores y Bordes en CSS

Haciendo click en el color HTML podemos modificar/acceder al color en RGB y Hexadecimal.

Algunas propiedades de los bordes en CSS: https://www.w3schools.com/css/css_border.asp

Para indicar una unidad de medición NO se debe dejar espacios en su declaración (1px - 1 px <-NO).

3. Manejo de Bordes en CSS

Página de referencia para ver el tipo de bordes disponibles en CSS: https://www.w3schools.com/css/css_border.asp

border es una propiedad que simplifica otras propiedades, por ejemplo, border contiene la propiedad border-style, border-color, border-width, etc.

Podemos mixear bordes según nuestra conveniencia, seguirá el orden: arriba-derecha-abajo-izquierda.

4. Ancho de Bordes con CSS

Podemos aplicar distintos grosores a los bordes de CSS.

Si queremos aplicar 4 bordes de distinto tamaño utilizaremos el siguiente formato: superior-derecha-inferior-izquierda.

Si queremos aplicar 3 bordes de distinto tamaño utilizaremos el siguiente formato: superior-DerechaIzquierda-inferior.

Si queremos aplicar 2 bordes de distinto tamaño utilizaremos el siguiente formato: SuperiorInferior-DerechaIzquierda.

Tenemos las propiedades thin, medium y thick como propiedades de tamaños definidas en CSS.

5. Códigos de Colores en Bordes CSS

Si bien con la propiedad de border (que engloba otras propiedades) podemos aplicar color, también tenemos la propiedad border-color. Al igual que con el width, podemos jugar con el color superior-inferior-izquierda-derecha de nuestro ancho.

Al igual que con bordes, se mantiene el mismo criterio para los lados.

Si queremos aplicar 4 bordes de distinto color utilizaremos el siguiente formato: superior-derecha-inferior-izquierda.

Si queremos aplicar 3 bordes de distinto color utilizaremos el siguiente formato: superior-DerechaIzquierda-inferior.

Si queremos aplicar 2 bordes de distinto color utilizaremos el siguiente formato: SuperiorInferior-DerechaIzquierda.

6. Redondeo de Bordes en CSS

Comparte los mismos criterios que los 2 anteriormente vistos.

Si queremos aplicar 4 redondeo bordes de distintos utilizaremos el siguiente formato: SuperiorIzquierdo-SuperiorDerecha-InferiorDerecha-InferiorIzquierda.

Si queremos aplicar 3 redondeo bordes de distintos utilizaremos el siguiente formato: SuperiorIzquierdo-SuperiorDerechaInferiorIzquierda-InferiorDerecha.

Si queremos aplicar 2 redondeo bordes de distintos utilizaremos el siguiente formato: SuperiorIzquierdoInferiorDerecha-SuperiorDerechaInferiorIzquierda.

3. Box Model en CSS

1. Box Model en CSS

Cualquier elemento HTML está compuesto por Box Model. Es una "caja" que se compone, de adentro hacia afuera, de los siguientes elementos: contenido-padding-border-margin. Cada uno de estos elementos que rodean al contenido tiene un width de alto y ancho.

El padding y margin son transparentes, y el border puede tener color.

Para calcular el alto y ancho de nuestro contenido HTML debemos tener en cuenta estos 4 elementos.

2. Ancho Elementos Box Model en CSS

Al único elemento que le podemos agregar color es al borde. El margin y padding son transparentes.

A nuestro elemento HTML le podemos asignar un ancho y un alto para predefinir cuanto lugar va a ocupar en pantalla.

3. Outline en CSS

Entre el margen y el borde tenemos un concepto que se llama outline. Así que también podemos establecer un outline entre el borde y el margen, también podemos manipular el color. Con este concepto no debemos sumar los pixeles que establezcamos para el ancho y alto del elemento.

Funciona y es parecido en términos de propiedades al border (No puede combinar ciertas propiedades).

No debemos sumarlo al ancho y/o alto de nuestro Box Model porque el outline va a ocupar parte de los pixeles del margen (Salvo que tenga un margen de 20px y un outline de 30px).

Si me paso de outline con respecto al margo, el color que haya elegido va a solapear/superponer a los otros contenidos que coincidan en la pagina. Tampoco va a ocupar espacio como elemento de linea en HTML, el oultine estaría fuera del rango del elemento.

4. Outline Offset en CSS

Podemos establecer un espacio entre nuestro concepto de outline y el border.

Para ello utilizaremos la propiedad de outline-offset.

Al igual que la parte 3, existen restricciones con respecto a las propiedades en comparación a border (todas las propiedades son seteadas de manera global, no se puede personalizar superior-derecha-inferior-izquierda).

No podemos asignar un color, de por sí es transparente.

5. Padding en CSS

El padding es posible personalizarlo de manera superior-derecha-inferior-izquierda.

Si queremos aplicar 4 paddings distintos utilizaremos el siguiente formato: superior-derecha-inferior-izquierda.

Si queremos aplicar 3 paddings distintos utilizaremos el siguiente formato: superior-DerechaIzquierda-inferior.

Si queremos aplicar 2 paddings utilizaremos el siguiente formato: SuperiorInferior-DerechaIzquierda.

6. Box Sizing en CSS

Para no pasarnos del ancho en el outline y no generar problemas de superposición podemos agregar la propiedad de box-sizing.

Con la propiedad border-box ya no hace falta calcular el ancho y alto, la misma propiedad va a calcularla y va a ajustar nuestro elemento a los pixeles que tenga dentro de los 4 elementos de Box Model.

7. Propiedad max-width-height en CSS

Si un elemento tiene X cantidad de width (ancho) y achicamos la pantalla hasta pasar a una resolucion menor al width indicado tendremos una barra de scroll horizontal en el navegador web.

The max-width property in CSS is used to define the maximum width of an element. The value of the width cannot be larger than the value by max-width. If the content is larger than the max-width then it will go to the next line and if the content is smaller than max-width then it has no effect. (https://www.geeksforgeeks.org/css-max-width-property/).

8. Propiedad margin: auto para centrar elementos en CSS

Esta propiedad es muy recurrente para centrar elementos independientemente de la resolución de pantalla del equipo que este visualizando la página web.

También podemos aplicar las reglas de superior-derecha-inferior-izquierda, superior-DerechaIzquierda-inferior y SuperiorInferior-DerechaIzquierda a la propiedad de margin.

9. Propiedad inherit en CSS

Esta propiedad nos permite heredar de otras propiedades declaradas en una clase CSS. La herencia funciona como una mamushka. En el ejemplo de nuestro div podemos heredar del margen ya que es el elemento externo más cercano que tiene declarado este elemento.

10. Concepto de margin collapse en CSS

Los márgenes superiores e inferiores a veces pueden colapsar entre elementos, es decir, se juntan ambos márgenes y provocar una "colisión".

Es posible que el margen del elemento inferior que colisiona con el elemento superior fusione estos márgenes dependiendo del tamaño del cual sea mayor.

Inspeccionar ambos elementos en el navegador.

Mas información: https://www.w3schools.com/css/css_margin_collapse.asp

4. Manejo de Colores en CSS

1. Manejo de Colores en CSS

Existen 140 tipos de colores por nombre en HTML.

Fuente: https://htmlcolorcodes.com/es/

Con la propiedad de float vamos a poder flotar los elementos uno al lado del otro de izq/der (siempre que haya espacio) en lugar de ponerse de manera vertical cada elemento. Se vera más adelante bien en profundidad.

2. Código de colores RGB en CSS

Los colores RGB se componen por Red Green y Blue, nos permite una gama de ~16M de colores distintos.

En RGBA la A será el alpha, es decir, la transparencia que tendrá nuestro color RGB. El Alpha va de 1 a 0, admite obviamente números flotantes.

3. Código de colores Hexadecimal en CSS

El código hexadecimal se compone de 16 dígitos (0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F).

El código debe comenzar por el símbolo #, continuado por los 6 dígitos que componen dicho código.

Los 6 dígitos se componen de manera parecida al RGB (#RRGGBB).

Si nuestros dígitos RRGGBB se componen del mismo simbilo hexadecimal podemos obviar el 2do digito de la composición anteriormente nombrada, si por ejemplo tuviésemos RRGGBB (ffffff) podríamos poner RGB (fff). Simplificamos pares.

También podemos aplicar transparencia a nuestro código hexadecimal. Esto funciona agregando un par más de dígitos al código hexadecimal.

4. Código de colores HSL en CSS

HSL es un acrónimo para Hue-Saturation-Lightness. Se compone de colores del 0-360, Saturation del 0-100% y Lightness del 0-100%

Fuente de consulta: https://purple11.com/static/fed42130c194b0c240a4ec10408adf97/8282f/hsl-cover-2.png

También podemos aplicar transparencia, funciona de igual manera al rgb.

5. Manejo de Fondos (background) en CSS

1. Manejo de Fondos en CSS

Con las propiedades background-image, con la función url podemos darle el path interno/externo a nuestra imagen.

Es importante saber que solo aplicando esta propiedad la imagen no se va ajustar a la resolución de pantalla del dispositivo que este visualizándola, se duplicará la imagen para llenar el fondo X cantidad de veces.

Para cambiar el comportamiento debemos utilizar la propiedad background-repeat: (opción que nos sea conveniente).

Se recomienda aplicar un color de fondo para que si la imagen no se repite se ponga un color acorde a la imagen del fondo.

2. Manejo de fondos en CSS - parte 2

También podemos aplicar imágenes de fondo a cualquier elemento HTML.

Podemos cambiar el posicionamiento inicial de nuestra imagen con background-position: (opción).

La propiedad background-attachament cambia el comportamiento de nuestro fondo al hacer zoom. background-attachament: fixed; Va a mantener la imagen fija al hacer scroll en la pantalla.

La propiedad background engloba muchas de estas propiedades que vimos anteriormente.

Ejemplo: background: dodgerblue url("../img/fondo.png") no-repeat right top;

6. Manejo de Texto en CSS

1. Formato de Texto en CSS

Para dar formato al texto podemos cambiar la fuente, agregar sombreado, subrayado, espacio entre líneas, espacio entre palabras, espacio entre las letras, etc.

Con la propiedad text-transform podemos indicar que tipo de formato (mayus, minus, mayus y minus) queremos para nuestro cuerpo de texto.

La propiedad text-shadow genera una sombre para nuestro texto con el formato EjeX-EjeY-Blur-Color.

La propiedad letter-spacing nos permite separar las letras entre sí, para darle un mayor espaciado. Permite tanto valores positivos como negativos.

La propiedad word-spacing nos permite separar las palabras entre sí, para darle un mayor espaciado. Permite tanto valores positivos como negativos.

La propiedad text-align nos permite poner mover el contenido hacia la izquierda, derecha, centro o justificarlo, entra otras cosas. Funciona como en WORD cuando queremos centrar un texto, o ponerlo hacia la izquierda, etc.

La propiedad direction marca como empezara el texto.

La propiedad text-decoration es mayormente utilizada en links, nos permite subrayar el texto, tacharlo, entre otras cosas.

La propiedad text-indent va a agregar sangría al comienzo de nuestro párrafo.

La propiedad line-height sirve para separar las líneas, por defecto es 1.2. No hace falta utilizar una medida ya que es un valor que se va a tomar de manera proporcional según el tipo de fuente que hayamos seleccionado.

La propiedad white-space tiene varios atributos accesibles, entre ellos el nowrap, que permite romper la caja contenedora del texto y salirse de sí misma.

7. Fuentes en CSS

1. Manejo de Fuentes en CSS

Existen varias familias de fuentes y clasificaciones.

La familia serif se clasifica por ser más sofisticadas y tener redondeos en sus letras. Dan más formalidad y elegancia.

La familia sans-serif se clasifica por ser más minimalista y simplificada. Da más modernidad.

La familia monospace se clasifica por ser utilizada para dar un aspecto más de "máquina de escribir".

La familia cursive se clasifica por ser de escritura cursiva.

La propiedad font-family nos despliega una lista de fuentes de la misma familia, muy parecidas en tipos. Si una falla o no está disponible en el navegador web continua con la siguiente en la lista (De la más específica a la más genérica). Si el nombre de la fuente es compuesto, debemos utilizar comillas para indicar el nombre, de lo contrario no hace falta.

Ejemplo: font-family: 'Times New Roman', Times, serif;

2. Estilos de Fuentes en CSS

Con la propiedad font-family podemos aplicar, como en word, que la letra sea del tipo normal, italic, bold, oblique, etc.

Con la propiedad font-weight podemos darle un ancho a nuestra fuente

La propiedad font-variant es para especificar si queremos que nuestra fuente se muestre en mayúsculas, pero con el concepto de small caps.

3. Tamaño Fuentes en CSS

La propiedad font-size nos permite incrementar o decrementar el tamaño de la fuente que estamos utilizando.

Podemos utilizar diferentes tipos de medidas o los parámetros por defecto que nos provee el CSS (large, small, etc).

Lo común en pixeles es de 16px en navegadores.

Una unidad de em (1em) es igual a 16px.

Otras unidades: https://devcode.la/tutoriales/unidades-vh-vw-css/

4. Google Fonts en CSS

Si no queremos utilizar ninguna de las fuentes que nos provee CSS, podemos utilizar fuentes de google -> https://fonts.google.com/.

Utilizar fuentes externas ubicadas en internet son más lentas para cargar y más permeable a errores. Podemos descargar las fuentes y vincularlas al CSS para no causar estos problemas.

Google fonts nos provee de uno o más tags <link> para vincular la biblioteca de la fuente a nuestro HTML/CSS o podemos utilizar el @import de google para vincularlo a nuestro CSS (al principio del archivo) directamente quitando el elemento de style. Se deja este elemento si vamos a utilizarlo dentro de una etiqueta.

El mismo google fonts nos dice como utilizarlo en nuestro CSS debajo del vínculo que nos provee -> CSS rules to specify families.

Al día del realizado el curso estamos utilizando la segunda versión de google fonts (css2 en el link) pero podemos importar datos de la primera versión de google fonts.

La primera versión permitía utilizar efectos, los cuales se pueden ver en la documentación.

Guía: v1 -> https://developers.google.com/fonts/docs/getting_started | v2 -> https://developers.google.com/fonts/docs/css2

5. Atributo font en CSS

Podemos poner todas las propiedades de fonts que vimos en un principio en una sola línea.

El orden es el siguiente: (font-style font-variant font-weight *font-size/line-height *font-family) | * -> Elementos requeridos sí o sí.

8. Iconos en CSS

1. Iconos de Bootstrap en CSS

¿Qué es Boostrap? Bootstrap es una biblioteca multiplataforma o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como extensiones de JavaScript adicionales. A diferencia de muchos frameworks web, solo se ocupa del desarrollo front-end.

Página de Iconos: https://icons.getbootstrap.com/ - Se puede utilizar a través de la instalación por npm, descargar por zip o importar a nuestro CSS o HTML con un link (CDN: Content Delivery Network) al igual que hacíamos con las Google Fonts.

Para utilizar un icono vamos a utilizar el tag y lo vamos a asociar con la clase que queremos de nuestro Bootstrap.

Podemos importar un svg para personalizar aún más el archivo (cambiarle el color, tamaño, etc) o hacerlo directamente en el CSS o etiqueta HTML.

2. Iconos de Google en CSS

Su vinculación funciona de la misma manera que Bootstrap.

Página de Iconos: https://fonts.google.com/icons - https://developers.google.com/fonts/docs/material_icons

A diferencia de Bootstrap, vamos a indicar el icono que queremos no como parte de una clase sino como parte de valor del tag icono. Ejemplo: search.

El tag va a depender del icono. Google nos provee con el tag ya pre fabricado para importar.

La propiedad vertical-align nos permite alinear los elementos a un mismo eje X.

3. Iconos de Ionic en CSS

¿Qué es Ionic? Ionic es un SDK front-end de código abierto para desarrollar aplicaciones móviles híbridas que utilizan tecnologías web como HTML, CSS y JavaScript. Proporciona componentes basados en tecnología web optimizada para dispositivos móviles, así como API nativas que utilizan Cordova e Ionic Native.

Página de iconos: https://ionic.io/ionicons - En la solapa Usage está todo lo que debemos saber para utilizar correctamente el framework.

---

4. Generalidades

Recordemos que mientras más vínculos de este tipo tengamos en nuestra página web, la página se vuelve más pesada para cargar en el usuario.

9. Display en CSS

1. Propiedad Display en CSS

La propiedad display en CSS nos va a permitir mostrar u ocultar elementos o también modificar la forma en que se muestran algunos elementos HTML.

Los elementos div, hX, p, listas entre otros tiene en común que su propiedad es display: block;

Los elementos span, anchor, links, imágenes entre otros son un elemento inline.

display: block; -> Cada uno de estos elementos va a ocupar una nueva línea por completo.

display: inline; -> El elemento no ocupa una nueva línea, simplemente continua a partir del otro elemento.

Si queremos ocultar algún elemento HTML y que no ocupe espacio podemos hacer un display: none;

Si queremos ocultar algún elemento HTML y que siga ocupando espacio podemos hacer: visibility: hidden;

10. Posicionamiento de Elementos con CSS

1. Centrar elementos de tipo Div con CSS

Una ventaja de utilizar porcentajes y no pixeles es que se adapta a la resolución de pantalla en la cual se despliega el navegador.

2. Posicionamiento en CSS

Existen 5 tipos de posicionamientos en CSS: static, relative, fixed, absolute y sticky.

Por defecto todos los elementos de tipo div son estáticos. Las propiedades top, bottom, left y right no se aplican a los elementos estáticos, si a los demás.

Al utilizar las propiedades top, left, bottom y right tenemos que tener en cuenta que no afectará al posicionamiento de los demás elementos que pongamos a continuación.

La posición static mantendrá el div en la posición por default, sin poder modificar su posición.

La posición relative mantendrá el div en la posición por default o la que le indiquemos en el CSS.

La posición fixed se va a colocar de manera relativa al viewport. Se va a mantener de manera fija en la posición de pantalla que le indicamos/que tiene por default. Si tenemos un scroll vertical y utilizamos el scroll, este elemento va a bajar conforme se utilice la barra de scroll.

La posición absolute es similar a la static, se utiliza normalmente cuando estamos trabajando dentro de otro elemento de tipo div. El absolute se va a posicionar de manera "absoluta", pero teniendo en cuenta el elemento que lo contiene. Si no lo contiene ningún elemento se va a tomar en cuenta el body (En este caso, si solo tiene el body es una buena práctica utilizar el default). El absolut no funciona si su div padre es static. Se va comportar como un elemento de línea.

La posicion sticky se caracteriza por tener que usar alguna de las propiedades de top, left, bottom, right. Se posiciona como un elemento de manera relativa y a la hora de mover la pantalla se utiliza como un elemento de manera fixed. Si trabajamos con safari debemos tener en cuenta una corrección, debemos agregar position: -webkit-sticky;. La posición sticky no funciona en Internet Explorer

3. z-index en CSS

Hasta ahora trabajamos con el eje de X (horizontal) y el eje Y (vertical). El eje Z representa las capas de profundidad de un elemento dado.

En el eje Z se van a ir superponiendo los elementos según coincidan y según el orden en que se van agregando se van a ir superponiendo. Sin embargo, la superposición va a depender de varios factores. Los elementos static son elementos no posicionados, en cambio cuando es diferente la propiedad decimos que son posicionados. Por default se agregan de manera automática pero podemos personalizarlo (tanto negativo como positivo), también podemos personalizar tamaño del z-index.

La propiedad CSS z-index indica el orden de un elemento posicionado y sus descendientes. Cuando varios elementos se superponen, los elementos con mayor valor z-index cubren aquellos con menor valor. - https://developer.mozilla.org/es/docs/Web/CSS/z-index

4. Propiedad overflow en CSS

El concepto de Overflow en CSS ocurre cuando tenemos un contenido que supera el ancho de nuestro contenedor. Solamente la podemos trabajar si especificamos un alto/height de nuestro contenedor (se necesita un limitante vertical).

La propiedad por default es overflow: visible; Que mostrara el contenido que supera el contenedor.

La propiedad overflow: hidden; Todo lo que se sale del contenedor se ocultará.

La propiedad overflow: scroll; Generara un scroll para poder moverse de manera vertical y/o horizontal y visualizar todo el contenido.

La propiedad overflow: auto; Si se necesita un scroll se genera automáticamente de manera vertical y/o horizontal.

Las mismas propiedades aplican para overflow-x u overflow-y, de necesitar específicamente un eje.

5. Float en CSS

En ocasiones es necesario acomodar nuestros elementos de manera flotada. Esto nos permite agregar muchos elementos en una sola. Cuando no tiene más espacio disponible para agrupar los elementos flotados genera una nueva línea. "La propiedad float se comporta globalmente y no en el scope" -> No sé si es así realmente, averiguar.

Si queremos que los demás elementos que siguen ya no estén flotando vamos a utilizar la propiedad clear indicando si queremos limpiar el lado izquierdo, derecho o ambos.

6. Propiedad Display Inline-Blocke en CSS

Ya trabajamos con propiedades display inline y block, esta propiedad es combinada de ambas. La utilizaremos para combinar ambas propiedades y quitar las restricciones que estas poseen. Se comportará como un elemento de linea pero podremos agregarle propiedades de un elemento de bloque.

No serviría, por ejemplo, para asignarle un margen, padding, etc. A un elemento "inline".

11. Selectores Combinados en CSS

1. Selectores Descendientes en CSS

El selector combinado descendiente hace que una vez seleccionado un elemento se va a aplicar el estilo que hayamos configurado para todos los elementos descendientes sin importar si son directos o indirectos, es decir, son elementos hijos y a su vez elementos dentro de otros elementos (todos los nested elements dentro del descendiente).

Ejemplo: div.contenedor.descendiente p

2. Selector Child en CSS

El selector descendiente aplica a todos los elementos, sin importar si son elementos hijos directos del elemento o si son hijos de otro elemento ("hijo indirecto", siempre y cuando este englobado por mi selector descendiente). En cambio, con el selector child solo va a afectar a los "hijos directos" y no a los indirectos.

Ejemplo: div.contenedor.hijo > p | Con el simbolo > indicamos que vamos a seleccionar los hijos directos del elemento que indiquemos.

3. Selector Sibiling en CSS

El selector Sibiling/Adyacente únicamente va a seleccionar al elemento que esté directamente después del elemento que hemos configurado. Tiene que estar exactamente después. Deben tener el mismo padre en común (x ejemplo el body).

También tenemos el Selector general, que permite seleccionar más de un tag adyacente siempre y cuando sea del mismo tipo y se encuentre en el mismo nivel.

Ejemplo: div.contenedor.adyacente + p | El símbolo + indica que solo vamos a seleccionar los elementos del tipo. Si no lo utilizamos va a tener un tipo genérico y va a agarrar el primer elemento que se encuentre después.

div.contenedor.hermano ~ p | Selector hermano permite seleccionar más de un párrafo del mismo tipo seleccionado, siempre y cuando se encuentren al mismo nivel.

4. Pseudo Clases en CSS

Las pseudo clases nos va a permitir manipular ciertos estados de nuestros elementos HTML. Por ejemplo, pasar por encima de un elemento de tipo div el cursor, hacer que cambie de color, o cómo se comporta cuando tiene click, etc.

Con el : se manejan los estados de los elementos HTML.
https://www.w3schools.com/css/css_pseudo_classes.asp

5. Pseudo Elementos en CSS

Los Pseudo Elementos nos permiten agregar estilo a ciertos elementos HTML. Sin embargo, estos estilos que vamos a agregar son de maneras más particulares de lo que hemos visto con Pseudo Clases. Por ejemplo, podemos agregar un cierto elemento antes o después de un elemento seleccionado de tipo HTML, o también, por ejemplo, podemos modificar la primera letra de un párrafo, la primera línea, etc.

Con el :: se manejan los estilos del elemento HTML.
https://www.w3schools.com/css/css_pseudo_elements.asp

6. Transparencia de Imágenes en CSS

Al igual que con los pseudo elementos, podemos modificar imágenes para cambiar su transparencia, posicionamiento, etc. Según nosotros queramos.

12. Gradientes en CSS

1. Gradientes en CSS

Podemos utilizar gradientes lineales o gradientes radiales. https://www.w3schools.com/css/css3_gradients.asp - Generador online: https://html-css-js.com/css/generator/ o https://cssgenerator.org/

Gradiente lineal: background: linear-gradient(El color inicial, color que finaliza);

background: linear-gradient(to X,#f72585, #3a9ca3); | X = derecha, izquierda, arriba, abajo.

background: linear-gradient(El color inicial, color intermedio, color que finaliza);

background: linear-gradient(to X Y, #f72585, #3a9ca3); | Podemos indicar que sea to bottom right para que sea cruzado.

background: linear-gradient(Xdeg, #f72585, #3a9ca3); | Parecido al anterior, le proporcionamos un angulo.

background: radial-gradient(circle, #0d47a1, #42a5f5); | Un degrade del centro hacia fuera.

2. Sombras en CSS

Como vimos lecciones anteriores, podemos agregar sombras a nuestro HTML con la propiedad text-shadow.

La propiedad se compone de text-shadow: EjeX EjeY Blur COLOR;. Generará una sombra y la podemos mover y blurrear como se muestra más arriba. Los ejes aceptan coordenadas negativas.

Ejemplo: text-shadow: 2px 2px 3px green;

3. Box Shadow en CSS

Con la propiedad box-shadow vamos a poder agregar una sombra a un el elemento HTML, comunmente se utiliza en la etiqueta div. Funciona de manera similar al text-shadow pero tenemos un atributo más que podemos aplicar para hacer más grande la sombra. Se compone de la siguiente manera: box-shadow: EjeX EjeY blur SOMBRA COLOR;

Al igual que el text shadow, los ejes aceptan coordenadas negativas.

13. Flexbox en CSS

1. Flexbox en CSS

El concepto de flexbox nos va a permitir agregar elementos dentro de otros elementos. Por ejemplo, un contenedor div y dentro del mismo párrafos, otros contenedores, etc.

Es una caja flexible que permite la introducción de otros elementos dentro de la misma. Es muy útil por ejemplo para flotar elementos, ya que anteriormente sin los flexbox debíamos volver a especificar que no queríamos que se flotara la siguiente y era permeable a errores.

Para que un contenedor aplique este concepto debemos utilizar la propiedad display: flex;

Por default se cargan de izquierda a derecha.

flex-direction: X; Cambia el sentido en el cual se van a agregar, si en forma de columna, fila, etc.

Con la propiedad flex-wrap: wrap; decidimos que si nuestra pantalla se hace mas pequeña los elementos pasan a otro renglón o no.

El flew-flow: (DIRECTION) (WRAP) engloba ambas propiedades.

La propiedad justify-content indica como se van a empezar a acomodar los elementos según vayamos cargando tags, de izquierda a derecha, hacia el centro, con espacios entre medio, con espacios entre ellos y los lados, etc.

Por default nuestros elementos se alargan y se adecuan al height de nuestro flexbox, si no queremos que esto ocurra la propiedad align-items: flex-start; También contiene otras propiedades para ajustar los elementos. Alinea los elementos de manera horizontal.

La propiedad aling-content es el espacio que tenemos entre nuestros elementos de manera vertical, funciona parecido al aling-items.

La propiedad order dentro de un elemento HTML va a modificar como se van a ordenar los elementos dentro del flexbox. Por default tiene el valor de 0 y según como los vayamos declarando se van a ir agregando.

Con la propiedad flex-group podemos indicar que algún elemento ocupe mayor espacio que los otros elementos. Funciona sobre el eje X. Ejemplo: flex-grow: 3;

La propiedad flex-basis funciona de igual manera que flex-group pero con unidades de mediciones como los pixeles. Ejemplo: flex-basis: 100px;

Con la propiedad de aling-self vamos a poder alinear un ítem independiente según queramos, funciona parecido al aling-items.