**INTRODUCCION**

**HTML**



Su nombre proviene del inglés *Hyper Text Markup Language*, que significa "lenguaje de marcado de hipertexto".

¿Qué quieres decir con "lenguaje de marcado"? Bueno, HTML no se considera un lenguaje de programación como los demás y ya sabrás por qué.

Imagínate una página web, como la que estás leyendo ahora mismo. Puedes ver que hay varios elementos separados, como encabezado, título, párrafos, imágenes y tantos otros. Toda la organización de estos elementos se realiza mediante HTML. Se utiliza para crear toda la estructura de la página y, para ello, utiliza las famosas tags (etiquetas) para indicar dónde se implementará cada tipo de elemento.

Por ejemplo, si desea insertar un párrafo en la pantalla, puedes usar la tag <p> para eso, colocando el texto deseado dentro del elemento, como se muestra a continuación:

<p>Este es un párrafo</p>

Por lo tanto, existen varias tags de diferentes estructuras para ensamblar el esqueleto de la página. Pero, ¿cómo es posible que este párrafo sea azul? ¿O cambiar la fuente de la letra? Aquí viene nuestra próxima tecnología: CSS.

**CSS**



*Cascading Style Sheet*, más conocido como CSS, es un lenguaje de estilos que, como HTML, tampoco se considera un lenguaje de programación. Es la encargada de separar la parte estructural de la aplicación (que estará en manos de HTML) de la parte estética. Para usar CSS, usamos la siguiente sintaxis:

selector {

propiedad: valor;

}

El selector será el elemento que queramos estilizar (podría ser una tag, una clase, un identificador ...), la propiedad será lo que vamos a cambiar (como color, tamaño de fuente, ancho ...) y el valor será realmente el cambio (como: rojo para color, 18px para tamaño de fuente, y así por delante).

Entonces, cuando usamos una tag HTML como la tag <p> que usamos como ejemplo anterior, podemos diseñarla de la siguiente manera:

p {

color: blue;

}

Con eso, todos los párrafos se volverán azules.

Un dato más interesante: si traducimos el nombre de esta tecnología, tendremos “hoja de estilo en cascada”. ¿Qué significa eso?

CSS se puede escribir dentro del archivo HTML, usando el estilo como elemento <style> o como atributo de algún otro elemento <p style="">. También se puede escribir en un archivo separado, solo CSS, importándolo al documento HTML de la siguiente manera:

<link rel="stylesheet" href="nombre\_del\_archivo\_css.css">

Sin embargo, algo muy importante a destacar es la posibilidad de utilizar más de un archivo CSS al mismo tiempo, para estilizar la aplicación. De ahí la palabra "cascada". Esto permite varias interacciones diferentes, pero debes tener cuidado de no perderte y confundir el código, siempre siguiendo las reglas para esto.

**Javascript**



Para completar la gran tríada del desarrollo front-end, contamos con Javascript, el único lenguaje de programación de las 3 tecnologías mencionadas.

Puedes ver que ya hemos creado la estructura de la página con HTML y hemos aplicado estilo a los elementos con CSS. Pero, ¿qué pasa con las funcionalidades dinámicas que vemos cuando abrimos un sitio web?, ¿cómo suceden?

Esto lo hace Javascript. Agrega movimiento a las páginas web, además de permitir el procesamiento y transformación de los datos enviados y recibidos. Te permite crear contenidos que se actualizan de forma dinámica y animada, dando vida a aplicaciones que antes solo estaban estructuradas estáticamente con HTML.

Siempre que vea un cuadro de alerta sonando en un sitio web, o un mecanismo de autocompletar en algún campo, sepa que Javascript está funcionando. Un ejemplo de código usando el cuadro de alerta:

alert(‘Hello World’);

Al igual que CSS, Javascript se puede escribir dentro del código HTML:

<script> alert(‘Hello World’); </script>

Cómo en un archivo separado, importándose dentro del atributo de ese mismo elemento:

<script src="nombre\_del\_archivo\_js.js"></script>

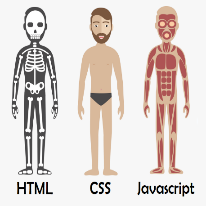
Algo importante a tener en cuenta es que este lenguaje se puede utilizar tanto en el lado del cliente como en el lado del servidor, utilizando tecnologías necesarias para ello, como Node.js.

**El papel de cada uno en la aplicación.**

Por tanto, después de conocer un poco sobre cada una de estas tecnologías, es posible ver las diferentes funciones que desempeñan dentro de una página web:

* [**HTML**](https://www.aluracursos.com/cursos-online-front-end/html-css): lenguaje de marcado utilizado para estructurar elementos de la página, como párrafos, enlaces, títulos, tablas, imágenes e incluso videos.
* [**CSS**](https://www.aluracursos.com/cursos-online-front-end/html-css)**:** lenguaje de estilo utilizado para definir colores, fuentes, tamaños, posicionamiento y cualquier otro valor estético para los elementos de la página.
* [**Javascript**](https://www.aluracursos.com/cursos-online-front-end/javascript)**:** lenguaje de programación utilizado para hacer la página con más movimiento, pudiendo actualizar elementos de forma dinámica y manejar mejor los datos enviados y recibidos en la página.

Comparando cada uno de ellos ahora, podemos usar el cuerpo humano como una página web como ejemplo, de la siguiente manera:



**HTML es el esqueleto, formado por los huesos que lo sostienen. CSS es la piel, el cabello y las ropas, creando la apariencia que realmente vemos cuando miramos a alguien. Y por fin, Javascript es el músculo que le da movimiento al cuerpo.**

CURSO PAGINA

HTML5 y CSS3 parte 1: Mi primera página web 6

HTML5 y CSS3 parte 2: Posicionamiento, listas y navegación 10

HTML5 y CSS3 parte 3: Trabajando con formularios y tablas 18

HTML5 y CSS3 parte 4: Avanzando en CSS 32

**HTML5 y CSS3 parte 1: Mi primera página web**

## Realice este curso para HTML y CSS y:

* Aprenda lo que es HTML y CSS
* Entienda la estructura básica de un archivo HTML
* Utilice el navegador para inspeccionar elementos
* Aprenda a definir estilos para elementos usando CSS
* Desarrolle una página Web desde cero

Otro editor de texto es, además de VSC, Atom id

Existen seis diferentes tamaños de títulos: **<h1>,<h2>,** etc.

**<b></b>** (bold) también se puede utilizar **<strong></strong>**

Al igual que **<i></i>** (italic) se utiliza **<em></em>** de énfasis.

<!DOCTYPE> lo colocamos en mayúscula por convención.

En el último vídeo, conocimos la etiqueta DOCTYPE. ¿Cuál es su función?

La etiqueta DOCTYPE sirve para informar al navegador cuál versión de HTML estamos usando.

La etiqueta informativa <meta charset=”UTF-8”> informa que estamos trabajando con el idioma español, portugués donde se tiene caracteres especiales.

¿Por qué es importante usar la etiqueta <meta charset=”UTF-8”>

Es necesario informar al navegador que estamos usando un diccionario específico que contiene caracteres especiales propios de varios idiomas populares en el mundo, siendo nuestra lengua española uno de ellos, ya que contiene acentos y la letra ñ por ejemplo.

¿Cuál es la diferencia de colocar informaciones en el head y en el body?

Se acostumbra colocar las etiquetas de contenido en el body y las etiquetas de información en el head

Como buena práctica de programación se acostumbra estructurar el HTML colocando las informaciones que queremos pasar para el navegador en el head y las etiquetas de contenido en el body

¿Cuál es la sintaxis correcta para separar la propiedad y su valor en el CSS?

Principio del formulario

El carácter ":" (text-align: center). El carácter : sirve para atribuir un valor a una propiedad.

La metodología in line es definir cada estilo en las líneas del HTML.

Otra manera de añadir estilos es utilizando la etiqueta ***<style>*** dentro de ***<head>***.

¿Cómo funciona el estilo en cascada en el CSS?

Todo lo que adicionamos en el elemento padre es reflejado en los elementos hijos.

¡Alternativa correcta! Su nombre lo dice Cascade Sheet Style (Hojas de estilo en cascada), por lo tanto, a medida que la cascada va descendiendo, todo aquello que hacemos en el elemento padre, refleja en el elemento hijo.

**Extra: Colores hexadecimales**

el sistema hexadecimal, que no es más que una combinación de nuestro sistema numérico decimal de diez elementos con las primeras seis letras de nuestro alfabeto, que en este caso voy a copiar este texto, sería una representación de los diez elementos, del 0 al 9, más las primeras seis letras de nuestro alfPor eso se llama hexadecimal, porque está representado por el 0 al 9, A, B, C, D, E, F. Entonces, la combinación de estos 16 elementos, me permiten en menos caracteres que binario por ejemplo, poder representar varias cosas.

[02:25] Es por eso que es muy usado en el mundo de la informática el sistema hexadecimal y van a ver que muchos códigos vienen representados en un formato hexadecimal justamente para usar menos caracteres, menos bits en este caso, y bytes de recurso computacional, para hacer representaciones y comunicarse con la máquina.

[02:48] En el caso de los colores, en el caso de la representación de colores en HTML, lo que tenemos en hexadecimal es una representación de seis posiciones, donde las primeras dos posiciones representan al rojo, las otras dos posiciones representan al verde y las otras dos posiciones representan al azul. Como pueden ver, coloqué aquí sus primeras letras en inglés, de red, green, blue.

[03:30] Y esto, puede que lo hayan visto, que es la combinación RGB. ¿Cómo se representan en algunos programas y en HTML también los colores? Entonces esta representación de hexadecimal significa que 0 es lo menos fuerte hablando de colores, sería lo más tenue en colores, y F sería la máxima cantidad de intensidad en el color.

[04:05] Entonces por ejemplo si yo quiero representar el negro, que sería la ausencia de color, lo represento con seis ceros. Voy a ponerlo aquí abajo. Entonces la representación del negro son seis ceros. Si yo vuelvo a mi código aquí, por ejemplo si quiero ponerle negro, puedo definir 2, 3, 4, 5, 6. Y fíjense que aquí me muestra que es el color negro. Entonces, seis ceros representan al negro.

[04:44] De la misma forma, si quiero representar el blanco es el otro extremo, son seis F, estoy aquí en este extremo. Volviendo acá si yo coloco seis F, el color representado va a ser el blanco. De la misma forma, si quiero representar el rojo, entonces las dos primeras casillas son rojos, dejo el F, el más fuerte, y los otros los dejo en cero.

[05:10] Entonces esto es lo mismo que en vez de poner red aquí, puedo colocarle el código hexadecimal y fíjense que me coloca el rojo. Igualmente puedo usar el sistema de codificación de colores RGB, donde significa que 0 en este caso es el nivel de intensidad de color menor, más pequeño, y 255 es el máximo.

¿De qué manera podemos representar colores en el CSS?

Hexadecimal, RGB y con los nombres de los colores

Podemos representar el color rojo con el nombre red, el hexadecimal #FF0000 y el RGB rgb(255,0,0).

* A configurar la presentación de los textos: alineamiento (text-align), tamaño de la fuente (font-size), el color del fondo (background) y el color del texto (color).
* **CSS inline**: adicionar estilos CSS en la misma línea donde tenemos nuestra etiqueta HTML.
* La etiqueta **<style>**: dentro de esta etiqueta podemos colocar marcaciones de CSS referentes a los elementos que tenemos en nuestro HTML.
* La presentación del CSS como un archivo externo.
* Cómo funciona el estilo en cascada del CSS.
* Cómo importar un archivo externo de CSS dentro de nuestra página HTML.
* Cómo representar colores en el CSS: 1) a través del nombre del color en inglés, 2) a través de su representación hexadecimal y 3) a través del RGB.

¿Cómo hacemos para adicionar una imagen en el contenido de nuestra página?

<img src="direccion">

La etiqueta img tiene la propiedad src, donde adicionamos la dirección.

El equipo de Front End, como lo dice su nombre, normalmente hoy en día en las empresas está formado por más de una persona, porque es prácticamente imposible que una sola persona se encargue de todo el desarrollo de una página web, debido a que con el pasar del tiempo esa construcción de página web ha ido ganando muchas más responsabilidades y muchas más actividades.

[00:43] Entonces no es solo programar. Entonces, aparte del programador, como lo vemos aquí, que ya lo vamos a explicar un poco más en qué consiste su rol como tal, también necesitamos tener responsable o responsables, porque pueden ser personas separadas. En algunos casos en algunas empresas este papel está en uno solo, va a depender sobre todo del tamaño de la empresa.

[01:10] Pero alguien que se encargue por el user experience y el user interface, que básicamente es la experiencia del usuario en ese sistema, en esa página web, y la interfaz del usuario, que es todo el sistema en sí donde el usuario interactúa.

[01:30] Y también, debido a que la generación de contenido es una buena estrategia para conseguir clientes, muchas empresas están optando por contratar un copywriter que es un generador de contenido, generador de artículos, generador de informaciones que llamen la atención y que llamen clientes, y que también muchas veces esa u otra persona se encarga del SEO, que es el Search Engine Optimization, que igual ahorita lo vamos a explicar con un poco más de detalle.

* Cómo reestructurar nuestro código, removiendo los estilos CSS inline y colocándolos en el archivo CSS externo.
* Cómo crear un identificador para marcar específicamente un elemento: cómo realizar esa referencia a ese identificador en el CSS.
* Cómo adicionar una imagen en nuestra página.
* Cómo ajustar la altura del elemento, a través de la propiedad height.
* Cómo ajustar la anchura del elemento, a través de la propiedad width.
* Cómo ajustar el margen interno del elemento, a través de la propiedad padding.
* Cómo ajustar el margen extern del elemento, a través de la propiedad margin.
* Cómo se conforma normalmente un equipo de front-end hoy en día.

Dada la siguiente lista:

* Ítem amarillo
* Ítem verde
* Ítem rojo

¿Cuál sería la estructura HTML para crearla?

Principio del formulario

ul>

<li>Ítem amarillo</li>

<li>Ítem verde</li>

<li>Ítem rojo</li>

</ul>

¡Alternativa correcta! La etiqueta para usar listas no ordenadas es <ul>, y cada ítem de la lista necesita ser marcado por un <li>

¿Cómo crear varios bloques de contenido en el HTML?

Bloques de código dentro de varios <div>

¡Alternativa correcta! La etiqueta div es la etiqueta específica para dividir el contenido en bloques, ayuda en la organización y estructuración de nuestro HTML.

The **vertical-align** [CSS](https://developer.mozilla.org/en-US/docs/Web/CSS) property sets vertical alignment of an inline, inline-block or table-cell box.

¿Cuándo debemos usar una class en nuestro código?

Cuando queremos repetir el estilo CSS en varios elementos.

¡Alternativa correcta! Cualquier elemento o etiqueta puede tener una class.

* Conocimos las listas (ordenadas y no ordenadas), trabajamos con listas no ordenadas.
* Vimos que para cada ítem de la lista necesitamos usar la etiqueta <li>.
* El concepto de las clases en el CSS, estas sirven para marcar ítems, solo que podemos replicar los estilos en varios elementos.
* Cómo referenciar una clase en el CSS.
* Divisores de contenido, utilizando la etiqueta <div>.
* Los comportamientos inline y block

No es una buena practica utilizar h1, h2, h3, etc en css para darle estilo a los títulos para esto, se utillizan clases.

¿Qué etiqueta usamos cuando queremos crear un encabezado en la página?

header

¡Alternativa correcta! La etiqueta header es una etiqueta de contenido de HTML que nos permite crear el encabezado de nuestra página web. Recuerda no confundir la etiqueta head con header.

* El concepto de encabezado de página y como crearlo.
* Que no es recomendado crear estilos usando etiquetas, lo ideal es usar selectores de clases para todo.

**HTML5 y CSS3 parte 2: Posicionamiento, listas y navegación**

En este curso se realizó la página web: productos.

Entonces, dentro del header vamos a hacer referencia a nuestra imagen, a nuestro logo, y la vamos a colocar dentro de un H1. Y ahora preguntarán, ¿por qué dentro de un H1, si no es un título? Justamente para darle esa importancia que se merece al logo de nuestro encabezado porque va a ser la parte principal en este caso de nuestra página y también de nuestro encabezado.

[02:52] Entonces, vamos a crear una etiqueta H1, y dentro de nuestro H1 vamos a usar la etiqueta img que representa a una imagen, y pasamos la propiedad source

¿Cómo podemos destacar el logo de nuestro sitio web? Elige, de las opciones abajo, la alternativa correcta que hace que la imagen del logo destaque, usando las etiquetas de título h1.

<h1>

<img src="">

</h1>

¡Alternativa Correcta! La etiqueta <h1> envuelve la etiqueta de imagen, que es donde buscamos el contenido que será expuesto.

**<a>: El elemento ancla**

El *Elemento HTML Anchor* **<a>** crea un enlace a otras páginas de internet, archivos o ubicaciones dentro de la misma página, direcciones de correo, o cualquier otra URL.

A modo de ejemplificar, vamos a ir aquí, a nuestro Visual Studio Code y les voy a mostrar que aparte de poder referenciar vínculos internos, páginas internas de nuestra página, también podemos hacer links, podemos enlazar a páginas externas de nuestro proyecto, de nuestra página.

En este caso, vamos a hacer referencia a la página de Google, bastante simple. Solo que tengo que tener cuidado de escribirlo completamente con http, dos puntos, dos barras, y aquí coloco la dirección completa de google.com.

Solo que a partir de HTML 5, en este caso, la quinta versión es un poco más buena con nosotros y nos creó una etiqueta, nos dejó disponible una etiqueta que se llama nav, que justamente es para colocar nuestro panel de navegación.

En HTML 5 se dieron cuenta que prácticamente el 99.99% de las páginas necesitan o tienen un panel de navegación, entonces para ello fue que crearon esta etiqueta, que nos la dejaron disponible, y ahora sí podemos usar directamente esta etiqueta para referirnos a nuestro panel de navegación.

 Entonces, en este caso, para volver nuestra lista, nuestro panel en formato horizontal, vamos a usar la etiqueta ***nav***, vamos a marcar, referir a la etiqueta ***nav*** y al ***li***, porque el formato, la propiedad horizontal aquí la habíamos visto anteriormente con la propiedad ***display,*** en el cual habíamos visto que las listas, por default, vienen con el ***display-block***, configurado como block, que significa que cada uno de los ítems ocupan el 100% del espacio de la página.

Entonces, vamos a colocar ese espaciamiento interno con la propiedad ***margin***. En realidad esta propiedad tiene cuatro parámetros. El primero me define un margen, un espaciamiento interno hacia arriba, que lo voy a definir con 0. El segundo a la derecha, que también le voy a colocar 0. El tercero, hacia abajo, que también va a ser 0.

[10:03] Y por último, a la izquierda que, en este caso, le voy a colocar 15 pixeles.

¿Cuál es la propiedad que usamos para poner el contenido del texto en mayúscula?

**text-transform: uppercase**

¡Alternativa Correcta! La propiedad text-transform es la responsable de la transformación del texto a mayúscula, cuando usamos el valor uppercase.

¿Por qué necesitamos quitar los estilos con el reset.css que el navegador crea automáticamente?

Para que nuestro estilo creado no sea influenciado por el estilo que el navegador agrega por defecto.

¡Alternativa Correcta! El navegador crea el estilo para todas las etiquetas que considera adecuadas, entonces, necesitamos moverlo para que el estilo que creemos no sea influenciado por el estilo que el navegador agregue.

¿Para qué sirve el posicionamiento absolute?

Para cambiar la posición del punto inicial donde se encuentra el elemento.

¡Alternativa Correcta! Con la position: absolute, yo puedo posicionar mi elemento en cualquier lugar de la página.

¿Cuál es la mejor técnica para alinear una div al centro?

Usar el cálculo de margin automático

¡Alternativa Correcta! Delegamos para que el navegador haga la cuenta al calcular la largura restante, dividir entre dos y agregar la mitad en cada uno de los lados.

¿Por qué debemos usar las etiquetas semánticas, como <main>?

Para dejar el código más legible, tanto para nosotros, como para el navegador.

¡Alternativa Correcta! Las etiquetas semánticas hacen que el navegador entienda mejor nuestro contenido, y con eso cargue mejor nuestra página.

¿Cómo se comporta un elemento que tiene su display con el valor inline-block?

Pudiendo ajustar el tamaño de su contenido, dejándolo regulable.

¡Alternativa Correcta! El tamaño puede ser ajustado, tanto en la largura, como en la altura.

¿Cuál es el comportamiento del elemento cuando agrego espaciamiento interno, el padding?

El elemento aumenta su tamaño.

¡Alternativa Correcta! El elemento aumenta su tamaño. Cuando declaro que existe un espaciamiento interno, estoy aumentando el contenido

Cuando agrego la siguiente declaración de borde:

border: 3px solid #000000;COPIA EL CÓDIGO

¿Cuál es el comportamiento esperado?

Un borde sólido, con 3 pixeles de ancho, entre el padding y el margin.

¡Alternativa Correcta! En la estructura de “cajas” del CSS, un borde queda posicionado entre el padding y el margin.

Bordes redondeados.

Es como si en nuestra figura paralelogramo, se insertaran en los vértices circunferencias. Ejem.

    border-top-left-radius:10px;

    border-top-right-radius: 25px;

    border-bottom-right-radius: 40px;

    border-bottom-left-radius: 50px;

o bien, la propiedad ***border-radius.***

border-radius: 10px 25px 40px 50px;

¿Cómo usar el CSS para hacer los diseños de las esquinas redondeadas de los bordes?

Usando la propriedad border-radius.

¡Alternativa Correcta! Con la propiedad border-radius, podemos diseñar la esquina redondeada de los elementos.

Lo que aprendimos en esta aula:

* A través del CSS, aplicar bordes en los elementos.
* Los diferentes tipos de bordes.
* A dejar el borde redondeado

CAPTURANDO EL MOVIMIENTO DEL MOUSE

Cómo construimos estos efectos? Lo primero que tenemos que tener conciencia es que yo necesito capturar eventos del usuario. O sea, cuando el usuario hace algún comportamiento, cuando él tiene algún comportamiento, yo lo tengo que capturar con CSS y hacer alguna cosa, o sea ejecutar alguna cosa.

[01:25] Por ejemplo, ¿qué es lo que está pasando? Estoy capturando el movimiento del mouse de mi usuario cuando se coloca en nuestro panel de navegación. Entonces, cuando se coloca en algún link de nuestro panel de navegación, pasa que se subraya y se coloca un color diferente y se destaca. Lo mismo aquí, cuando nos posicionamos en alguno de los productos, pasa el mismo efecto, se colorea el borde y da ese efecto de destaque.

Hay diferentes formas de hacer exactamente eso, y en realidad la forma más correcta o la forma más común trivial de usarla es con JavaScript, solo que el foco de este curso no es el JavaScript.

esa propiedad hover la que está capturando el movimiento del mouse, y le estamos diciendo que cuando capture dentro o encima, mejor dicho, de nuestro panel de navegación de los links de nuestro panel de navegación, que se cambien de color. (productos.html).

¿A qué elementos puedo cambiar su aspecto a través de su comportamiento con CSS?

A todos los elementos.

¡Alternativa Correcta! En todos los elementos de HTML puedo mapear el comportamiento con CSS y cambiar el aspecto del elemento.

 Así como tenemos el ***hover*** para capturar el movimiento del mouse, tengo otra propiedad que es el ***active*** para capturar el clic, o sea como diciéndole el movimiento activo, y a ese elemento activo, que es al que fue hecho clic, al elemento que está clicado, por así decir, aplícale alguna cosa, ejecutemos alguna instrucción.

Es importante destacar que, esta captura de eventos solo nos permite modificar la parte visual, los efectos visuales de nuestra página. Por ejemplo, a través de esa captura yo no puedo volver a cargar la página, o no puedo llenar un formulario con algunos datos. Eso se hace a través de JavaScript. Bueno, el alcance de este curso no involucra, no envuelve JavaScript.

¿Cuándo podemos usar el comportamiento del hover en un elemento?

Cuando queremos resaltar alguna característica en el momento en que el mouse del usuario esté por encima del elemento.

¡Alternativa Correcta! En el hover, mapeamos el comportamiento del mouse por encima del elemento, y podemos usar esto para descartar visualmente el elemento.

Por último, cuando el usuario pase el cursor sobre algún ítem de la lista productos, se modificará el tamaño del texto del h2:

.productos li:hover h2 {

font-size: 33px;

Lo que aprendimos en esta aula:

* Algunas pseudo-clases CSS
  + hover, cuando el usuario pasa el cursor sobre el elemento.
  + active, cuando un elemento está siendo activado por el usuario.
* A cambiar el color del texto y/o del borde de un elemento, cuando el usuario pase el cursor sobre el mismo.
* A cambiar el color del borde de un elemento, cuando el mismo está siendo activado por el usuario.

PIE DE PAGINA

y fíjense que ya tenemos nuestro background. Ahora, si vemos la imagen, esta imagen es cuadrada. No es una imagen rectangular. La pregunta es: "¿por qué me ocupa todo el espacio?" Justamente, porque al colocar la propiedad background, lo que hace el background es repetir como si hiciera un copypaste, como si copiara y pegara la misma imagen N veces hasta ocupar el 100% de la horizontal de nuestra página.

¿Cómo agregamos una imagen de fondo en un elemento?

Con el valor url en el background del elemento.

¡Alternativa Correcta! En el CSS, cuando hacemos referencia a la dirección de una imagen, usamos el valor url.

CARACTERES ESPECIALES

Dentro de nuestro footer, aparte de nuestra imagen, teníamos nuestro texto que es nuestro copyright, que actualmente está configurado con el valor padrón que le da nuestro navegador, que es el color negro, por eso es que no está legible con este background, y 16 pixeles de tamaño. Lo que tenemos que hacer es configurarlo para dejarlo como está aquí, en nuestra página modelo.

[00:56] Entonces, tenemos que alterar el color, ver cómo colocar este texto o carácter especial, y está un poco más pequeño y mejor localizado. Entonces, es justamente eso lo que veremos en este video. Yendo a nuestro HTML, la estructura de nuestra página, vamos a ver que tenemos una etiqueta p que es donde está nuestro texto dentro de nuestro footer.

[01:26] Entonces, yo podría ir a nuestro CSS y colocar aquí nuestro selector footer p, y me estoy refiriendo a él. Funcionaría correctamente, solo que no es una buena práctica como lo hemos visto anteriormente, sobre todo abusar las p directo con las etiquetas, porque el día de mañana nuestra página puede crecer y puede que tengamos la necesidad de acrecentar algún otro texto.

[01:56] Entonces, siguiendo esa buena práctica de programación de usar las clases, vamos a construir una clase que la vamos a llamar copyright. Entonces, con nuestra clase creada, puedo venir aquí y ya no utilizo el selector de etiqueta sino utilizo el selector de clase copyright, recordando que necesito colocar el punto.

 solo nos está faltando ese símbolo o carácter especial, que representa al copyright. Parece una arroba de los correos electrónicos, pero es una C con circunferencia alrededor. Esa C con su circunferencia alrededor pertenece al diccionario unicode.

[03:45] El diccionario unicode, es el diccionario que contiene todos los idiomas, todos los caracteres especiales y caracteres normales de los idiomas del occidente de nuestro planeta, todas aquellas lenguas, aquellos idiomas conocidos también como indoeuropeos, que dentro de ellos está el inglés, el español, el portugués, el francés, en fin, todos los idiomas, vuelvo a repetir, que hablamos en la parte occidental de nuestro planeta.

[04:18] Y para saber, voy a venir aquí, para no confundir nuestra página en construcción, para saber qué código representa ese carácter, tenemos una página que se llama Unicode, ya me aparece aquí porque ya la había visitado, unicode-table.com, entro en esta página, ya me aparece en español porque mi navegador está en español, y aquí busco copyright.

[04:51] Al buscar copyright me da dos resultados de búsqueda. El primero, grabador de sonidos no, signos de derecho de copia. Hago clic aquí y fíjense que es el símbolo que estamos buscando, y aquí me muestra varias formas de representar este símbolo en nuestro código. Entonces, tenemos un número unicode, un código HTML, un código CSS y una entidad.

[05:22] Podemos usar el que queramos, pero normalmente, mayormente se usan, van a ver en bastantes páginas web, el símbolo de entidad. Entonces, vamos a usar esa entidad, que sería ese símbolo ampersand y la palabra copy. Y ese texto, ese código lo colocamos en nuestro texto dentro del HTML, básicamente como si fuera un texto más.

Extra: La expresión del hemisferio occidental puede hacer referencia a la mitad de la esfera terrestre que se encuentra al oeste del meridiano de Greenwich La principal masa terrestre que este contiene es **América y pequeñas porciones de África, Europa y Oceanía**.

¿Qué tipo de caracteres especiales podemos usar en nuestro texto?

Podemos usar cualquier carácter del Unicode.

¡Alternativa Correcta! Los caracteres Unicode están disponibles para ser usados como un complemento a nuestro texto.

Lo que aprendimos en esta aula:

* La etiqueta <footer>, para el pie de página de nuestra página.
* Que con CSS, podemos poner una imagen de fondo en un elemento. Cuando ponemos una imagen de fondo en un elemento, el CSS, por defecto copia y pega la imagen varias veces hasta ocupar el espacio del elemento.
* La tabla **Unicode**.

**HTML5 y CSS3 parte 3: Trabajando con formularios y tablas**

## Realice este curso para HTML y CSS y:

* Cree formularios complejos
* Trabaje con campos para dispositivos móviles
* Presente informacioes en tablas
* Use estilos para formularios, campos y tablas
* Entienda la jerarquía en el CSS
* Trabaje con transformaciones y transiciones

COMENZANDO LA PAGINA DE CONTACTO

Un formulario es todo aquello donde el usuario puede colocar algunos datos o hacer clic en algún botón o hacer clic en algún checkbox o seleccionar un ítem de una lista y eso va a generar alguna acción. En este caso, la acción es enviar el formulario para que nosotros efectuemos una reserva o recibamos algún mensaje que este cliente, que este usuario, nos quiere enviar.

[08:28] La página inicial de Google, conceptualmente hablando, no es más ni menos que un formulario, porque yo coloco mi búsqueda aquí, por ejemplo, voy a buscar café colombiano y si se fijan en la URL de esa búsqueda, en algún lugar Google pasa para su servidor que lo que estamos buscando es café colombiano.

[08:54] Y justamente aquí lo tenemos, fíjense, él coloca café + colombiano, que esa URL pasa para alguno de los servidores de Google y va y mapea todas las páginas posibles donde pueda tener esta palabra o que está haciendo referencia a la palabra café colombiano y así realiza la búsqueda. Entonces, esta página inicial en su esencia es un formulario.

[09:22] Lógicamente que hay un montón, hay varios otros recursos que usa Google, sin embargo, lo que quiero que nos quede claro o pasarles el mensaje, que esta caja de texto es como si fuera un formulario, una caja de texto de un formulario que cuando nosotros ejecutamos nuestra búsqueda, Google realiza alguna acción en función de esos datos que estamos pasando.

[09:47] Entonces, la idea es que nuestro formulario haga o tenga ese comportamiento de Google, que el usuario, el cliente coloque sus datos y nosotros podamos efectuar algunas acciones a partir de ahí.

Para decidir cuándo debemos utilizar un formulario en nuestras páginas, se pueden analizar algunos motivos:

1) Cuando queremos enviar datos a un servidor

2) Cuando queremos capturar información que el usuario escribe

3) Cuando queremos enviar los datos a otra página

¿Cuáles realmente se pueden tener en cuenta?

Todos los motivos pueden ser tomados en cuenta

¡Alternativa correcta! Un formulario puede servir, entre otras cosas, para enviar datos a otra página, capturar información que el usuario escribe y enviar datos a un servidor.

CAMPOS BASICOS

 hay una etiqueta que se llama ***form,*** justamente que hace referencia, que representa, que marca un formulario

siempre cuando colocamos un input de tipo texto, tiene que ir con un label asociado para indicar a qué se refiere ese input; es como si siempre van casados. Y para casarlos, para relacionarlos aún más, podemos aquí colocarle un id a este input.

[02:37] Le colocamos que ese id es el nombre y apellido, recordando que como buena práctica, en los nombres de variables de id usamos únicamente minúsculas, y dentro del label colocamos un for, que le estamos diciendo, fíjense que incluso el Visual Studio me ayuda, le estamos diciendo: "referite al id, al input con id nombre y apellido".

 Y para finalizar, todo formulario necesita de un botón enviar o de dar el enter en el formulario para que sea enviado ya sea a un servidor, ya sea a un correo electrónico o a otra página en la que estemos trabajando donde le mandamos estos datos.

[06:07] Y para ello, tenemos una etiqueta que justamente se llama input type submit, entonces, en este caso, ya nuestro HTML nos ayuda porque nos va a crear ese botoncito de submit y aquí está en enviar. Fíjense que me coloca enviar. ¿Por qué? Porque mi navegador está configurado en español, entonces, ya me reconoce el idioma y me coloca que ese submit es enviar.

[06:45] Si mi navegador estuviera configurado en otro idioma, bueno, va a colocar en el idioma respectivo. Pero para garantizar que independientemente del navegador, yo quiero mostrarle en español, yo coloco un value, aquí la propiedad value, y le colocamos en español "Enviar formulario". Vamos a guardar, volvemos aquí, actualizamos y ya tenemos aquí el botoncito de enviar formulario. (contancto.html)

¿Cuál es la etiqueta que usamos para emparejar un input, para tener un título para la entrada de datos?

<label>

¡Alternativa correcta! La etiqueta <label> es una etiqueta para la entrada de datos para el <input>.

 Recordando que tenemos la propiedad display y hay elementos de HTLM que por default vienen display-block, que block es cuando ocupan el 100% de la página, y otros elementos vienen con ese default inline. Inline es cuando ocupan solo su contenido, o sea, el espacio de su contenido dentro de la página.

[03:28] En el caso de los elementos de un formulario, todos ellos vienen con inline por default. Es justamente por eso que tenemos este comportamiento. Entonces, lo que haremos será vamos a nuestro archivo y vamos a usar form, en este caso, directamente label, para cada uno de nuestros elementos, form label y display, y al display le vamos a colocar block.

¿Qué propiedad debo cambiar para que mi <input> tenga un espacio entre el borde y el contenido?

padding

¡Alternativa correcta! La propiedad padding sirve para cambiar el espaciado interno, entre el contenido y el borde.

Lo que aprendimos en esta aula:

* A crear un formulario HTML.
* La etiqueta que lo representa es <form>.
* La etiqueta <input>, para la entrada de datos del usuario.
* A crear una etiqueta para el input con la etiqueta <label>.
* A conectar un input con su label.
* Colocamos un id al input y asociamos ese id al atributo for del label
* Algunos tipos de input, como text y submit.
* Qué label e input por padrón posee el display inline.
* A estilizar nuestro formulario.

TIPOS DE CAMPOS DIFERENTES

¿Cuál es la propiedad de un input del tipo radio que necesito crear para que todos los elementos formen parte del mismo grupo?

name

¡Alternativa correcta! La propiedad name solo se puede “completar” una vez, es por eso que, cuando seleccionamos uno de los ítems, se anula el otro, manteniendo solamente uno seleccionado.

 Vamos a nuestro Visual Studio entonces, y lo primero que haremos será corregir el tamaño de nuestro textarea, para ello, ya tenemos aquí el tamaño definido para los inputs, pero, en este caso, nuestro elemento se llama textarea. Hay una forma fácil en CSS de replicar estilos para varios selectores separándolos por la coma.

[01:01] Entonces, aquí donde tengo form input, puedo colocar form textarea, que va a hacer referencia a nuestro input textarea. (style.css)

 En el siguiente paso, lo que veremos será, fíjense que email, teléfono y WhatsApp que es nuestro input radio aparecen antes que el botón o que el espacio para seleccionar. Si voy a la página modelo, fíjense que eso aparece primero, o sea, el input radio aparece primero que el texto, y eso lo podemos solucionar directamente en nuestro HTML colocando nuestro input radio dentro de label, aquí.

[02:11] Entonces nuestro input radio va a aparecer antes que el label, la etiqueta, el no, y hacemos eso para los tres elementos, aquí, y para finalizar este, antes de WhatsApp.

está apareciendo con un salto de línea, aparece el ítem y después el label, da un salto, el otro item, un salto, el otro label. Y eso está así. ¿Por qué? Porque si recordamos todos nuestros inputs, aquí estamos aplicando el display block, entonces, ese comportamiento está siendo aplicado a todos los inputs, está tomando, está ocupando el 100% del espacio de la página, cada uno de los inputs.

[03:29] Para solucionar eso vamos a crear una clase para los input padrón que son estos de aquí, nuestros inputs que creamos iniciales, a los cuales queremos que el display block aplique.

[03:45] Entonces, en este input del nombre y apellido vamos a colocarle una clase que lo vamos a llamar input-padrón, diciendo que este input es del tipo padrón, para poder referenciarlos y así poder crear una característica, un visual específico, solo para esos inputs. Lo hago aquí con nombre, correo electrónico, teléfono, y también para el mensaje que quiero que tengan ese comportamiento.

¿Cuál es el mejor enfoque cuando tenemos una configuración de CSS aplicada a un ítem y queremos replicarla en un ítem similar?

Crear una clase

¡Alternativa correcta! Cuando tenemos un estilo que puede ser repetido, una buena estrategia es extraer eso a una clase y usar la clase para aplicar esos estilos en nuestros elementos.

Usar selectores de ítems separados por coma

¡Alternativa correcta! Esta es una forma sencilla de hacer que la misma configuración de CSS funcione para los dos ítems, ya sean selectores de elementos, de ids o de clases.

JERARQUIA EN EL CSS

Primero se encuentra los estilos inline, luego los id seguido por las clases, y por ultimo las etiquetas.

Pero, si tenemos una etiqueta en el css y otro selector con dos etiquetas, referenciando al mismo elemento, este ultimo tiene mas peso ya que es la suma de dos etiquetas. Por lo tanto se aplica el estilo dado en esta.

Ejem

Considerando el código de abajo:

<p class="parrafo">

p {

color: blue;

}

p.parrafo {

color: red;

}

.parrafo {

color: purple;

}

¿Cuál será el color final de mi párrafo?

Principio del formulario

red

¡Alternativa correcta! El selector p.parrafo es el más fuerte, y el color aplicado es el rojo (red).

Final del formulario

Seleccionando opciones

Si vemos nuestra página final, nos falta un ítem, un input de esos input complejos que estamos colocando, que es este dropdown, esta lista en la cual el usuario, el cliente puede colocar en cuál horario prefiere ser atendido, si por la mañana, por la tarde o por la noche.

[01:20] Y para hacer eso, vamos a crear nuestra etiqueta correcta en nuestra página de contacto en el HTML, antes de nuestro checkbox, vamos aquí a dar un espacio, y vamos a usar la etiqueta select, que justamente es la que hace eso.

y cada una de las opciones de nuestra lista, cada uno de los ítems de nuestra lista, se define con la etiqueta options, de opciones.

En el último video, se vio la etiqueta <option>. ¿Cuál es su utilidad?

Para crear los ítems seleccionables de un <select>

¡Alternativa correcta! La estructura de la etiqueta <select> está compuesta de uno o más <option>.

Lo que aprendimos en esta aula:

* El textarea, para entradas de texto de más de una fila.
* El input de tipo radio.
* Cómo agrupar varios input de tipo radio, impidiendo que más de un input sea seleccionado.
* El input del tipo checkbox.
* Que podemos crear un input dentro de un label, asociándolos.
* Más estilos para nuestra página.
* Cómo funciona la jerarquía en CSS .
* El select, que es selector, un campo de selección de un ítem, y el option, que representa cada opción del selector.

IMPUT PARA CELULARES

Lo que veremos ahora va a ser pensar un poco en la experiencia del usuario. Como desarrolladores web, personas que vamos a construir páginas web, tenemos que pensar que nuestro usuario es lo más importante.

[00:40] Tenemos que tener claro que nuestro driver, estamos construyendo una página web para que sea accesada y para facilitar la vida de alguien que, en este caso, tiene que ser nuestro cliente, nuestro usuario.

[00:56] Es por ello, que no debemos dejar pasar o no considerar que muchos de nuestros usuarios, muchos de nuestros clientes van a acceder a nuestra página a través de un dispositivo móvil, a través de un celular.

[01:10] Es más, hoy en día la mayoría de los accesos, gran parte de los accesos por internet son hechos a través de los dispositivos móviles. Es por ello, que HTML 5 ya tiene configuraciones padrón, que vienen por defecto, que ayudan a hacer que esa experiencia del usuario, cuando acceden a una página web a través del celular sea mucho mejor, solo que nosotros como programadores tenemos que hacer esas configuraciones.

[01:38] Para ello, les voy a mostrar una página web, una referencia que se llama, vamos a entrar directo al site, a la página mobileinputtypes.com. Ya había entrado aquí, doy un enter, y lo que tiene esta página es una página de contenido, de referencia de contenido, son los diferentes tipos de inputs que podemos definir.

[02:05] Entonces, texto es el que venimos trabajando que me va a permitir colocar cualquier texto. Si yo defino el tipo de input, el input type como email, ¿qué me va a permitir en el celular? Me va a aparecer el arroba y me va a ayudar a que mi usuario coloque el correo electrónico en una forma más fácil.

¿Por qué motivos es importante usar los tipos correctos de <input> para los usuarios que acceden a la página vía celular?

Para tener datos más correctos en el formulario

¡Alternativa correcta! Es mucho mejor tener datos correctos cuando tenemos nuestro formulario completo.

Para facilitar la vida del usuario

¡Alternativa correcta! Es extremadamente importante facilitar la vida de nuestros usuarios siempre.

Otra forma de ayudar a navegar a que nuestro usuario pueda usar de la forma correcta nuestro formulario es definiendo algunos campos como obligatorios, que es lo que veremos ahora.

[00:51] Eso, por un lado, ayuda a que el usuario sepa cuáles son los campos que tienen que ser llenados, y por otro lado, garantiza que nosotros como empresa, como barbería, colectemos datos de calidad, porque de nada sirve, de nada adelanta que nuestro usuario marque una agenda, una cita, y no nos envíe su nombre y apellido, no vamos a saber quién es o no nos coloque su correo electrónico, por ejemplo.

[01:18] Nuestra base de datos va a quedar incompleta si ese campo viene vacío. Entonces, es lo que vamos a ver, esa configuración y otras configuraciones para ayudar justamente a nuestro usuario a que esa navegación sea lo más friendly, lo más amigable posible. Vamos a ir a nuestro código.

[01:44] Aquí estamos en nuestro código, y lo que haremos será definir nuestro campo del nombre y apellido como ***required***. Usamos esa palabra reservada en inglés que significa requerido, y así de simple es para decirle que un campo es obligatorio, entonces fíjense, lo voy a enviar vacío y aquí me dice: completa este campo.

¿Cómo convertimos un campo común en un campo obligatorio?

Agregando la propiedad required

¡Alternativa correcta! Cuando agregamos la propiedad required, el navegador nos ayuda a validar si ese campo está rellenado.

MEJORANDO LA SEMANTCA DEL FORMULARIO

Ahora, lo que veremos será mejorar la estructura de nuestro código en HTML. Vengo aquí a nuestra página de contacto, contacto.html y aquí usamos la etiqueta div para separar los contenidos.

[00:56] Sin embargo, existe una etiqueta específica que se llama fieldset, que es una etiqueta que agrupa ítems que pertenecen al mismo grupo. Como aquí estamos, en este caso es ¿Cómo le gustaría que lo contactemos? Son los inputs radio, y todos los ítems de este grupo están hablando de la misma cosa, podemos usar el campo fieldset.

[01:30] No es que con div no funcione, sí va a funcionar. Solo que con fieldset le estamos diciendo: "Mira, navegador, todos los grupos que te estoy pasando, todos los ítems que te estoy pasando en este grupo pertenecen o hacen referencia a la misma cosa, a los mismos eventos o a la misma información, de alguna forma están relacionados". Eso es lo que queremos decirle, indicarle a nuestro navegador.

Y eso hacemos con los dos div. Entonces colocamos fieldset y para cerrar también fieldset. Y para citar un ejemplo cuando pedimos los datos de una tarjeta de crédito, por ejemplo, en una página web de Amazon o Spotify, cuando nos piden los datos de nuestra tarjeta de crédito para pagar, para suscribirnos o pagar una compra, esos campos, porque son varios, pueden y deben estar dentro de un fieldset, porque son datos que referencian a un tipo de conjunto de datos.

[02:39] Y el ***fieldset*** también, no usa la p para colocar texto. Como esto es un título, es el título de nuestro grupo, de aquí vamos a colocar una etiqueta que se llama legend, y es una etiqueta de contenido que abre y que cierra.

Para finalizar con estas buenas prácticas, ya que estamos hablando de ello, vamos a ir a nuestro código, y las imágenes suelen tener, es aconsejable colocarles un texto alternativo, entonces existe una propiedad alt. Aquí tenemos nuestro logo, en nuestro header, existe una propiedad alt donde le indicamos de qué se trata esa imagen. ¿Y eso para qué?

[04:34] Principalmente hay algunos navegadores o el usuario puede estar con problemas de internet o por algún motivo no identificado, la imagen puede no cargar en el navegador de nuestro usuario. Entonces, a través de ese texto alternativo le podemos decir de qué se trata esa imagen. Y esta imagen es el logo de la Barbería Alura.

[05:03] Y recordando que tenemos una imagen en el header y también tenemos una imagen en el footer. Para ello, lo copio y lo pego aquí, nuestra propiedad de ***alt***, en ambos casos, es el logo de la Barbería Alura. Es aconsejable colocar en todas las imágenes que tengamos en nuestra página siempre un texto alternativo, eso va a ayudar a la navegabilidad de nuestro usuario.

¿Qué etiqueta usamos para marcar un título de un grupo de campos en el formulario?

<legend>

¡Alternativa correcta! Es la etiqueta <legend> que usamos para un título de un grupo de campos en cualquier formulario.

Lo que aprendimos en esta aula:

* Algunos tipos de inputs para el celular: email, tel, number, password, date, datetime, month y search.
* Cómo configurar campos obligatorios, o sea, exigir al usuario para que algunos campos sean completados, a través del atributo required.
* Cómo mostrar una sugerencia para completar los campos, a través del atributo placeholder.
* Cómo dejar una opción marcada por padrón en nuestros input radio y checkbox a través del atributo checked.
* Cómo estructurar mejor nuestro código con fieldset y legend.
* Cómo agregar una alternativa a la imagen, describiéndola, con el atributo alt.

QUE SON LAS TRANSICIONES

 Volviendo a nuestra página, actualizo y fíjense que cuando me posiciono se cambia el color para un naranja más fuerte. Solo que esa transición es instantánea, no es en microsegundos. Voy a cambiarlo aquí a red para que noten que esa transición es superrápida. Fíjense cuando aquí está rojo.

[05:41] Y existe una propiedad en CSS que se llama ***transition***, es una propiedad que apareció en el CSS 3, que justamente lo que me permite es hacer esa transición en una forma más lenta. Entonces, en nuestro botón vamos a colocar transition, en el selector de clase de nuestro botón enviar, transition, dos puntos, y vamos a colocarle un segundo. Y aquí le vamos a decir que nos cambie el background.

Ahorita tenemos el cursor, que es esta flechita que viene por defecto, por padrón, y lo vamos a cambiar con el pointer, que es la manito con el dedo apuntando. Si yo actualizo aquí, ya tenemos el pointer apuntando a nuestro botón.

[07:50] Eso le ha dado un realce mayor a nuestro botón, a nuestro submit, y el usuario ya tiene más indicaciones. Es más intuitivo para él, que tiene que darle botón en submit, en enviar formulario, para que sus datos lleguen hasta nuestra barbería.

¿Cómo configurar la transición visual de un elemento que dura 3 segundos?

transition: 3s;

¡Alternativa correcta! Utilizamos la propiedad transition para proporcionar una transición visual de un elemento.

¿Qué propiedad del CSS usamos para aumentar un elemento proporcionalmente?

ransform: scale()

¡Alternativa correcta! Utilizamos la propiedad transform: scale() para aumentar un elemento proporcionalmente.

Lo que aprendimos en esta aula:

* Cómo colocar estilos al botón de envío de formulario.
* A realizar transiciones en nuestros elementos, con la propiedad CSS transition.
* A modificar el estilo del puntero del mouse, al pasar sobre un determinado elemento, a través de la propiedad CSS cursor.
* A realizar transformaciones en nuestros elementos, como aumentar proporcionalmente la escala de determinado elemento o rotarlo, a través de la propiedad CSS transform.

ESTRUCTURAS DE TABLAS

vamos a construir nuestra tabla en la página de contacto HTML, vamos a ir después del form. Una tabla está compuesta, cualquier tabla, si pensamos en una tabla de Excel por ejemplo, está compuesta de columnas y de filas. Para HTML existe una etiqueta específica que se llama table, del inglés tabla, y dentro de esta etiqueta nosotros definimos los table rows, las líneas de las tablas.

[02:18] Y dentro de cada table row definimos los td que vendrían a ser las celdas, cada una de las celdas de nuestra tabla, de cada una de las líneas. Entonces, en este caso, cada línea, nuestra tabla tiene cuatro líneas: el título, el encabezado y los tres días: lunes, miércoles y viernes, y cada una de las líneas tiene dos columnas, entonces, serían dos células que abren y que cierran, dos td.

. Perfecto, entonces solo repasando, la estructura de una tabla está compuesta de líneas y columnas, en cualquier lugar donde podamos ver, así podemos definir una tabla.

[04:14] Es bastante usada en Excel, como les comenté es tener ese racional de líneas y columnas, y la intersección de líneas y columnas es lo que conocemos como celdas en Excel. Esa estructura es bastante parecida aquí en nuestro HTM para construir tablas. Entonces, tenemos una etiqueta table, definimos cada uno de los tr que vendrían a ser las líneas, los table rows, row en inglés es línea o fila.

Y dentro de cada fila colocamos los td, que vendría a ser cuántas columnas queremos en esa tabla.

¿Qué etiqueta uso para crear filas en una tabla?

<tr>

¡Alternativa correcta! Utilizamos la etiqueta <tr> para marcar la fila de una tabla.

ETIQUETAS SEMANTICAS PARA TABLAS

así como en nuestro main, en toda la página, la podemos dividir en head, body y el footer, que sería el encabezado, el cuerpo de la página y el pié de página, la tabla también puede ser distribuida o estructurada de esa forma. Para ello, tenemos una etiqueta, refiriéndonos al encabezado, que se llama thead, t de table, tabla, y head de encabezado.

[01:00] Y dentro del thead, vamos a colocar nuestro título o nuestro encabezado de la tabla. Y también existe una etiqueta específica para las celdas que pertenecen a ese encabezado que son los th. Entonces, lo único que vamos a hacer aquí es sustituir, reemplazar los td por th. Perfecto.

[01:27] Así como tenemos un encabezado, un thead, tenemos un tbody, o sea, el cuerpo de la tabla. Y dentro del cuerpo de la tabla vamos a colocar el resto de las tr, de las etiquetas que representan a las filas de nuestra tabla y las otras columnas.

¿Cómo dividir una tabla de una forma más semántica?

Usando las etiquetas <thead>, <tbody> y <tfoot>

¡Alternativa correcta! Las etiquetas <thead>, <tbody> y <tfoot> ayudan a dejar el contenido de la tabla más dividido y más semántico.

Lo que aprendimos en esta aula:

* A crear una tabla HTML.
  + La *etiqueta* table, representa la tabla.
  + La *etiqueta* tr, representa la fila de la tabla.
  + La *etiqueta* td, representa la celda de la tabla.
  + La *etiqueta* thead, representa el encabezado de la tabla.
  + La *etiqueta* tbody, representa el cuerpo de la tabla.
  + La *etiqueta* th, representa la celda del encabezado de la tabla.
  + La *etiqueta* tfoot, representa el pie de página de la tabla.
* A colocar los estilos respectivos a nuestra tabla.

**HTML5 y CSS3 parte 4: Avanzando en CSS**

## Realice este curso para HTML y CSS y:

* Importe contenido externo a tu página HTML, como fuentes, videos y mapas
* Conozca las pseudoclases y los pseudoelementos
* Utilice selectores CSS avanzados
* Aprenda a lidiar con la opacidad y la sombra
* Comprenda Viewport y design responsivo

La etiqueta div es para dividir el contenido y que quede mucho mejor para el mantenimiento o para que otra persona pueda entender qué es lo que estamos colocando en nuestra página, para separar los bloques de contenido. Sin embargo existe una etiqueta section, que ya la hemos usado llamada section, que lo que hace es separar nuestro contenido por secciones, pero de una forma semántica.

O sea, si colocamos section, le estamos diciendo a la página web de que ese bloque de contenido está referenciando, se trata de un contenido que está relacionado entre sí, y como esta primera div que tenemos aquí en nuestro main es todo nuestro bloque, es un bloque de contenido sobre la barbería Alura donde presentamos un breve concepto, nuestra misión y qué productos estamos y servicios ofreciendo.

Entonces vamos a cambiar esta div por una etiqueta section

(index.html)

¿Cuál es la diferencia entre usar una <div> o una <section>?

Debemos usar la etiqueta <section> cuando el bloque es semántico

¡Alternativa correcta! Para un bloque donde el contenido tiene el mismo significado, el mismo sentido, usamos una <section>

 Una <div> se usa con el sentido de dividir el contenido HTML solo con fines visuales o para encapsular un bloque específico para aplicar algún estilo CSS.

La buena práctica que puede existir o que podemos tomar es que usamos ID cuando queremos alterar comportamientos y usamos clases cuando queremos alterar los estilos. Entonces como en el banner estamos alterando, actualizando el estilo, que es el tamaño, no es un comportamiento de la página, vamos a cambiar eso por una clase.

Tenemos dos clases: una clase, la section, de la sección, clase principal, y tenemos otra clase, nuestro título, que se llama título-centralizado. Si analizamos también este nombre de clase, es un pésimo nombre, una pésima nomenclatura para nombrar una clase, porque estamos mezclando el nombre que es el título, con un comportamiento.

[04:13] ¿Qué tal si no queremos que el título sea centralizado, que sea alineado a la izquierda? Nuestro nombre, nuestra nomenclatura dejó de tener sentido. Entonces la buena práctica es tener nombres que sean representativos pero que no sean tan específicos y principalmente no asociar comportamientos o estilos a esos nombres o esas nomenclaturas, porque esos comportamientos y estilos pueden cambiar a lo largo que vamos trabajando o pueden ser actualizados de aquí a un tiempo.

[04:43] Entonces vamos a colocarle aquí título principal, estamos alterando el nombre de nuestra clase, voy a copiar el nombre de nuestra clase, voy a nuestro style.css al final y aquí creamos un selector para nuestra clase título principal? ¿Qué vamos a hacer con ese título principal? Vamos a cambiarle el alineamiento, el text-align, queremos que esté en el centro nuestro título. Style.css

Y también vamos a cambiarle el tamaño de la fuente, el font-size. El tamaño de la fuente vamos a usar una referencia que es 2em. Al colocar 2em le estamos diciendo que queremos que sea dos veces el tamaño padrón, el tamaño por defecto de la fuente que tiene nuestro navegador.

[05:44] En este caso el navegador está usando 15 píxeles de tamaño de fuente. Entonces, al colocarle 2em le estamos diciendo que quiero dos veces ese tamaño. Para otros navegadores, para otros servicios operativos puede ser otro tamaño de fuente y lo que va a hacer es multiplicar por dos el tamaño de fuente padrón que tiene ese navegador. Style.css

Al crear cualquier estilo, ¿Cuál es el mejor padrón para marcar nuestro contenido al crear una clase?

Ser específico y genérico al mismo tiempo

¡Alternativa correcta! Un balance es la mejor alternativa. No tan específico, para poder repetir o reutilizar las clases y no tan genérico, para que no tengas que necesitar combinar clases.

Ser demasiado genérico hará que tus clases sean muy débiles y necesitarás combinar clases para "ensamblar" un elemento.

Ser demasiado específico hará que tus clases tengan nombres difíciles de reutilizar y acabes reutilizando muy poco las clases.

existe una propiedad que se llama ***float***, de flotar, de flotante, es como si nuestra imagen se despegara de la página, sin embargo deja una sombra, deja la sombra de esa imagen, y ese espacio, la sombra, es ocupado en la misma proporción, y en ese caso tenemos el texto redondeando a toda esa imagen. Y esa propiedad la usamos directo en nuestro CSS.

¿Para qué sirve la propiedad float?

Colocar el elemento en la lateral de nuestra página y dejar los otros elementos alrededor de él.

¡Alternativa correcta! Tanto el float:left como el float: right sirven para que el elemento se destaque en la pantalla, la superficie del elemento continúa siendo utilizada y los otros elementos de texto o elementos en línea se posicionan alrededor de él.

Al usar correctamente el float, hacemos resaltar solo para que el contenido se reordene alrededor. Para sobreponer elementos, usamos el position: absolute.

Lo que aprendimos en esta clase:

* A ajustar la página principal para utilizar los mismos padrones de la página de productos
* Medidas proporcionales con CSS
* Cómo funciona la propiedad float y cómo la podemos usar para que nuestros elementos floten sobre el lado que se indica.
* Como *limpiar* el float con la propiedad clear del CSS

FUENTES EXTERNAS

para ver la fuente, es importante resaltar y destacar que existen fuentes externas que podemos usar, sean privadas o sean públicas.

[01:46] Las privadas son las también llamadas propietarias, y hay empresas como Microsoft, Apple, en fin diferentes empresas que tienen sus propios tipos de fuente y los usan en sus sistemas operativos, en sus aplicaciones, en sus sistemas, en sus software y esas fuentes son propietarias esas empresas, para usarlas tendríamos que pagar, básicamente.

[02:10] Sin embargo existen otro tipo de fuentes que son públicas y que están abiertas para cualquier persona que quiera usarlas. Únicamente necesita importarlas y hacer esa referencia de manera correcta. Una de ellas, una empresa, una plataforma que las ofrece es el mismo Google.

Si entramos a Google fonts nos va a redireccionar a ***fonts.google.com,*** y aquí tenemos diferentes tipos de letras que son en este caso propiedad de Google, pero las disponibilizan, las dejan abiertas públicamente para todo aquel que quiera utilizarlas. Entonces aquí podemos ver diferentes tipos de letras.

Analice los siguientes beneficios de usar fuentes externas:

1. Estandarizar nuestro contenido en todos los navegadores
2. Dejar el sitio más bonito, fuera de lo común
3. Tener más alternativas de fuentes

¿Cuáles de las afirmaciones anteriores se benefician realmente del uso de fuentes externas?

Todas las opciones

¡Alternativa correcta! Usar fuentes externas es una alternativa muy funcional para que tengamos más opciones, que hacen que nuestro sitio web sea más bonito y exclusivo y además estandarizado en todos los navegadores.

TRABAJANDO CON MAPAS

amos a entender qué es esta etiqueta iframe y los datos con los que vino configurado. Iframe es una etiqueta para crear tipo ventanas, espacios, áreas en las cuales podemos colocar diferentes tipos de contenido, ya sean un mapa, ya sea videos, imágenes, GIF, cualquier elemento gráfico, visual o de video que queramos colocar en nuestra página es posible hacerlo a través de nuestra etiqueta iframe.

[05:02] Una de sus propiedades es el source. Aquí justamente es la ubicación en este caso de nuestra barbería, que es la ubicación de las oficinas de Alura en San Pablo. Vemos el tamaño, width 600, y una altura de 450, que fue el tamaño mediano con el que viene configurado por defecto, y vienen otras configuraciones también por default, que no las vamos a tocar ahora.

¿Para qué sirve la etiqueta iframe?

ara insertar diferentes elementos externos como ser imágenes, videos, mapas o hasta una página de facebook o twitter.

¡Alternativa correcta! Exactamente a través de la etiqueta iframe podemos insertar diferentes tipos de elementos externos en nuestra página. Sean estos mapas, vídeos, imágenes u otras páginas web o páginas de redes sociales.

¿Cuáles de las siguientes sentencias son verdaderas en relación a la incorporación de vídeos externos en nuestra página?

1. Solo pueden ser insertados vídeos de Youtube
2. Se puede configurar desde que tiempo específico del vídeo queremos que se empiece a reproducir
3. Se usa la etiqueta iframe para incrustar vídeos

Solo 2 y 3 son verdaderas

¡Alternativa correcta! Es posible configurar el tiempo donde queremos que inicien los videos que insertamos en nuestras páginas. Por otro lado, a través de iframe es posible insertar varios tipos de elementos externos, entre ellos los videos son una alternativa viable.

Lo que aprendimos en esta aula:

* A utilizar fuentes externas en nuestras páginas
* Cómo incorporar un mapa en nuestra página
* Cómo incorporar un video en nuestra página

AJUSTANDO LOS DIFERENCIALES

 Y aquí ya vamos a ajustar nuestra lista de diferenciales, y en este caso lo que haremos será ajustar su anchura para 40%, ya que la imagen tiene 60%. ¿Entonces qué es lo que estamos intentando hacer? Es dejar la lista que ocupe 40%, la imagen que ocupe 60%, de los 640 píxeles que estamos definiendo para todo ese bloque.

[04:58] Vamos a guardar aquí y vamos a ir a nuestra página. Actualizamos, vemos que se actualiza el tamaño en forma correcta, solo que no tenemos todavía el alineamiento sobre la horizontal. Hay un salto de línea entre la imagen y nuestra lista, y eso ocurre por una propiedad que ya la hemos visto anteriormente, que justamente vienen por display-block por defecto.

[05:32] Vamos a ajustar eso. Lo haremos en nuestra lista de diferenciales. En nuestra lista de diferenciales le vamos a decir display, ahora que en vez de block, que viene por padrón, nos cambie para inline-block. Volvemos, actualizamos. No hubo ningún ajuste. ¿Por qué?

[05:55] Porque en el caso del inline-block hace una lectura del HTML tal cual el HTML esté, tal cual se encuentre. Y si vemos aquí en nuestro HTML tenemos un espacio entre nuestra lista y nuestra imagen. Un espacio propio del HTML. Sin embargo el inline-block está considerando ese salto de línea que hemos colocado.

[06:25] Para ello tenemos que colocarlos pegados, tenemos que juntar estos dos elementos, nuestra ul que es nuestra lista con nuestra img que es nuestra imagen, guardamos y ahora sí tenemos el efecto que queremos. Nuestra lista por un lado, en el costado izquierdo de nuestra pantalla, y al lado la imagen. (</ul><img…>)

PSEUDO CLASES

Ahora, en nuestro bloque los diferenciales, en la lista, vamos a usar pseudo-clases. Y para entender ello, vamos a ir a nuestro código, vamos a ir a nuestro CSS y aquí vamos a hacer uso de esas pseudo-clases. Hacemos nuestro selector de clase padrón y común, los dos puntos, y una de las pseudo-clases es first-child, del inglés primer hijo.

Lo que le estamos diciendo es al primer hijo o al primer elemento de ese bloque de elementos que en esteo caso son nuestros items, recordemos que para items tenemos una clase que está identificando a todos esos items. Entonces, de todos esos items, al primero pintamelo de negrito. Venimos aquí y nos pinta nuestro primer item de negrito. ***items:first-child{}***

 ¿Qué pasa si queremos el último? Tenemos el ***last-child***, el último hijo. Venimos, actualizamos y ahora está nuestro último. Dentro de las pseudo-clases tenemos una ventaja, que podemos indicarle con ***nth-child()***, el número en el orden creciente, podemos indicarle cuál item queremos que se pinte de negrito en este caso o aplicar cualquier efecto que queramos.

Otra ventaja que podemos tener con esto es que podemos darle un 2n por ejemplo y se van a pintar todos nuestros items pares, números pares, el segundo, cuarto, sexto y si tuviéramos, así iría por la frente.

Usamos las pseudo-clases para hacer selecciones especiales en los ítems de la lista de los diferenciales y aplicar algunos efectos especiales. ¿Cuál de las alternativas está usando correctamente las pseudo-clases para obtener la **lista-ingredientes** con el resultado esperado:

<ul class="lista-ingredientes">

<li class="items">6 huevos</li>

<li class="items">200 grs de tocino</li>

<li class="items">1/2 pimentón</li>

<li class="items">1 tomate</li>

<li class="items">1 cuchara de aceite</li>

<li class="items">150 grs de queso muzzarella</li>

<li class="items">Sal a gusto</li>

</ul>

**Resultado esperado:**

* *6 huevos*
* 200 grs de tocino
* 1/2 pimentón
* **1 tomate**
* 1 cuchara de aceite
* 150 grs de queso muzzarella
* *Sal a gusto*
* items:first-child{font-style: italic}
* .items:nth-child(4){font-weight: bold}
* .items:last-child{font-style: italic}
* ¡Alternativa correcta! La sintaxis de las pseudo-clases están correctamente usadas y aplicando los estilos requeridos para colocar formato itálico en el primer y último ítem. Y pintar de negrito el cuarto ítem.

APLICANDO GRADIENTES

justamente ahora lo que veremos será la transición de colores. Antiguamente para hacer esa transición de colores era supercomplicado en nuestras páginas porque había que colocar imágenes con ese degradado para tener esos efectos. CSS 3 resuelve ese problema y nos deja una función con la cual podemos trabajar colocando ese tipo de efecto, el tipo de degradado.

[09:31] Se llama linear-gradient, es la función que vamos a usar, y lo único que necesitamos pasarle son los colores en los cuales queremos aplicar ese degradado o gradiente. Vamos a colocar orange y blue, naranja y azul. Estamos diciendo: ve del naranja al azul. Si volvemos a nuestra página, estamos yendo de un color naranja a un color azul. Y fíjense esa transición, está viajando de ese color inicial hasta ese color final.

[10:10] Podemos colocar más colores, los colores que quisiéramos, la verdad: naranja, azul, rojo, negro. Estamos diciéndole que viaje del naranja al azul, al rojo y al negro. Vemos ese efecto. Tenemos naranja, azul, rojo y negro. Por defecto, el color es aplicado en la horizontal como podemos ver.

[10:37] Y una opción que nos permite el linear-gradient es poder trabajar con ángulos, o sea, decirle que nos aplique los colores en un ángulo definido, por ejemplo de 45 grados. Y eso lo hago colocando como primer parámetro ese ángulo en grados. Si actualizamos, ahora vemos que tenemos la aplicación de colores, el degradé de colores, en una inclinación de 45 grados.

[11:06] Podemos colocarle 90 grados, y de esa forma tendremos la aplicación en formato vertical. Va de un color naranja en la izquierda hasta un color negro en la derecha. Todos por defecto ocupan un espacio definido dividido proporcionalmente igual entre todos los colores. Eso también se puede configurar, y después del color, por ejemplo el orange, el naranja, yo quiero que ocupe el 50% de todo ese espacio.

[11:42] Cuando actualizo, vemos que el naranja ocupa el 50% y CSS automáticamente los otros tres colores los divide en el espacio en el 50% restante. Existe también la opción de no usar linear-gradient, sería el background en formato circular, radial-gradient. Vamos a usar los mismos colores aquí, vamos a eliminar este anterior, y tenemos el naranja, le podemos continuar diciendo que queremos que ocupe el 50% del espacio, el azul, rojo y negro. Background.

[12:29] Guardamos aquí, y si actualizamos vemos que tenemos aquí un efecto parece hasta el sistema solar. Comienza con el naranja ocupando el 50%, de ahí una elipse con circunferencia mayor, después el rojo y por último el negro que lo podemos ver solo un poco en las puntas. Lógicamente que ese no es el efecto que queremos, no quedó nada agradable visualmente.

¿Cómo hacer un degradado de 5 colores en un elemento?

background: linear-gradient(gray, yellow, red, orange, blue)

¡Alternativa correcta! Usamos dentro del mismo paréntesis, separando los colores con comas.

Ejem linear-gradient(90deg,gray, yellow 50%, red, orange, blue)

PSEUDO ELEMENTOS

 CSS 3 nos proporciona los pseudo-elementos, que nos permiten hacer selecciones específicas de algunos elementos y poder trabajar y configurar los estilos de nuestras páginas, elementos de nuestras páginas de esa manera.

ecordando que lo que nos permiten los pseudo-elementos es alterar el visual de nuestra página y agregar elementos sin necesidad de alterar directamente nuestro HTML. Vamos a ir a nuestro título principal y vamos a cumplir o realizar ese desafío que les hice. Entonces título principal.

[02:44] Y así como les dije cómo alteramos la primera letra de nuestro título, tenemos una propiedad, un pseudo-elemento que es first-letter. Entonces es exactamente la primera letra. Aquí necesito colocar solo dos puntos y alteramos el font-weight para negrito. Volvemos a nuestra página, actualizamos aquí y ya tenemos la primera letra pintada.

.titulo-principal:first-letter{

    font-weight: bold;

}

tra aplicación de los pseudo-elementos es que en muchas páginas web, blogs, posts, artículos, a veces la primer línea de cada párrafo tiene un ejemplo diferente, por ejemplo itálico. Eso lo podemos hacer también haciendo uso de los pseudo-elementos, llamamos a todos los párrafos, no necesito colocar puntos porque estoy llamando a un elemento de CSS, colocamos los dos puntos y first-line.

[04:03] Al colocar first-line, abrimos y cerramos nuestras llaves, colocamos font-style: italic. Volvemos, actualizamos y podemos ver que tenemos el efecto itálico en las primeras líneas de nuestros párrafos. Otra aplicación de los pseudo-elementos y bastante usada es el after y el before, lo que nos permite colocar elementos antes o después de algún elemento.

Entonces por ejemplo imagínense que queremos colocar un corchete antes y después de nuestro título. Entonces, título principal, usamos la expresión before, de antes en inglés, y aquí en contenido, content, le decimos que nos abra corchetes. Vamos a hacer exactamente lo mismo, solo que ahora after, o sea después de los títulos, cerrame ese corchete.

p:first-line{

    font-style: italic;

}

.titulo-principal:before{

    content: "[";

}

.titulo-principal:after{

    content: "]";

}

¿Cómo creo un pseudo-elemento usando CSS que anteceda a un elemento de HTML?

Con la propiedad :before

¡Alternativa correcta! Cuando queremos crear un elemento en la pagina, via CSS, antes del elemento de HTML, usamos la propiedad :before.

Lo que aprendimos en esta aula:

* A mejorar la semántica de la página principal, con nuevas divisiones, clases, etc.
* Nuevas pseudo-clases
* Cómo aplicar un *background* degradado en la página
* Pseudo-elementos.

SELECTORES AVANZADOS

 Selectores avanzados, como su nombre lo dice, son recursos de CSS 3 que nos permiten seleccionar, capturar elementos en forma rebuscada, en forma específica y en forma granular.

O sea podemos llegar al mínimo elemento de nuestro HTML y aplicarle un estilo diferente, un visual diferente a ese elemento.

¿Cómo hacemos eso? Para eso vamos a crear en nuestro código, en nuestro HTML dentro de nuestro main, vamos a crear un párrafo de ejemplo y justamente lo vamos a llamar párrafo ejemplo.

[00:54] Guardamos aquí, volvemos a nuestra página, actualizamos y aquí tenemos nuestro párrafo ejemplo. ¿Qué pasa si quiero colocarle un color diferente a ese párrafo ejemplo, un background rojo por ejemplo a ese párrafo ejemplo? Puedo ir a nuestro CSS, al inicio del código CSS de nuestra hombre, que aquí inicia, antes del banner, porque lo estamos colocando en realidad después del banner, antes del bloque principal, aquí colocamos main p.

[01:32] O sea las etiquetas main, los párrafos que están dentro de main. Y aquí colocamos un background red como habíamos dicho, color rojo. Volvemos, actualizamos y aquí tenemos un problema. Estamos aplicando el color rojo, pintamos efectivamente nuestro párrafo ejemplo, pero fíjense que se pintan todos los otros párrafos, los párrafos de nuestro bloque principal, el párrafo que tenemos en nuestro mapa también.

[02:12] Y eso pasa porque al colocar esta sintaxis, main p, o sea, seleccioname el main y todos los párrafos que están dentro del main, justamente va a ir y aplicar a todos los párrafos independientemente de la jerarquía, recordando que CSS son estilos aplicados en cascada, entonces va a aplicar tanto a sus hijos, a los nietos, a los bisnietos y si esa jerarquía sigue creciendo, va a aplicar absolutamente a todos.

[02:48] Para ello es que CSS nos presenta algunos recursos, para hacer esa selección en forma específica. Y uno de esos símbolos o elementos que podemos usar es el símbolo mayor. Al usar el símbolo mayor, yo le estoy diciendo: "Seleccioname todos los párrafos que son hijos del main".

[03:09] O sea, la sintaxis sería así, padre o elemento, símbolo de mayor, hijo. Y solo va a aplicar a sus hijos de ese elemento.

<main>

            <P>Párrafo ejemplo</P>

HTML

main> p{

    background: red;

}

CSS

Usando esta estructura:

<h2>

<section>

<h2>

<p>

<p>

<h2>

</section>

¿Cómo hacemos para seleccionar el último <h2> ?

section > p + h2

¡Alternativa correcta! El último h2 es hijo directo de la section y hermano del p.

CALCULOS CON CSS

 Y en este video lo que veremos será un recurso adicional de CSS que son los cálculos con CSS.

Y estos nos permiten alterar en forma fácil y referenciada las dimensiones de las imágenes o gráficos o elementos visuales que podemos llegar a colocar en nuestras páginas

¿Cómo usar medidas proporcionales para dejar un elemento 100% de ancho con el equivalente a un tercio del elemento padre, menos 10px?

width: calc( (100% / 3) - 10px )

¡Alternativa correcta! Es necesario calcular el un tercio del elemento, esa es una de las formas de hacer eso, luego le restamos los 10px.

CSS calc()

It is an inbuilt CSS function which allows us to perform calculations. It can be used to calculate length, percentage, time, number, integer frequency, or angle. It uses the four simple arithmetic operators add (+), multiply (\*), subtract (-), and divide (/).

Lo que aprendimos en esta aula:

Selectores avanzados CSS

* Selector > , para acceder a los hijos de determinado elemento. Por ejemplo, para acceder todos los p dentro del main:

main > p { }

* Selector +, para acceder al primer hermano de determinado elemento. Por ejemplo, para acceder el primer p después de una img:
* img + p {

}

* Selector ~, para acceder a todos los hermanos de determinado elemento. Por ejemplo, para acceder todos los p después de una img:
* img ~ p {

}

Selector not, para acceder a los elementos, excepto algunos. Por ejemplo, para acceder a todos los p dentro de main excepto el p que tiene id missao:

* main p:not(#missao) {

}

* Cómo hacer cuentas con CSS, con la propiedad calc

OPACIDAD EN LOS ELEMENTOS

En esta sección, en esta aula, lo que veremos será aprender a trabajar con opacidad y con sombras. Primeramente vamos a entender lo que es opacidad. Opacidad la podemos definir como la capacidad que tienen los elementos de dejar pasar la luz. Podemos citar un ejemplo cuando decimos: "Ese vidrio es opaco", es porque en algún porcentaje no deja pasar la luz.

Otro ejemplo claro que vemos de opacidad es cuando le colocamos la película a los parabrisas de los vehículos, esta película oscura que no deja pasar la luz. Estas películas tienen un nivel de opacidad, entonces si ese porcentaje es bajo, deja pasar más luz. Hay unas películas que colocan a los parabrisas que no dejan pasar absolutamente nada, significa que tienen un nivel de opacidad bastante alto.

 aprendimos que podemos usar la simbología o la representación RGB, de red, green, black, que es una combinación de tres números, en este caso los tres ceros. Es como decir numeral 000, es exactamente lo mismo.

Si yo guardo y volvemos a nuestra página y actualizo no pasa nada, porque lo único que estoy es reconfirmando el color negro. Ya está por defecto y le estoy diciendo "quiero que continúe negro", por así decirlo. Pero existe una pequeña variación de RGB que es el RGBA, el cual me permite colocar un nuevo parámetro, un cuarto parámetro en esta función, que va a ser la opacidad.

¿Cuáles tipos de elementos pueden recibir una capa de opacidad?

Todos los elementos y todos los colores

¡Alternativa correcta! Todos los elementos y todos los colores pueden tener una capa de opacidad.

SOMBRAS EN LOS ELEMENTOS

Las sombras son otro recurso nuevo de CSS 3. Antiguamente para poder colocar sombras a los elementos era complicado porque había que colocar otras imágenes y comenzar a jugar con la posición de esas imágenes.

Actualmente CSS 3 nos presenta una propiedad que se llama ***box-shadow*** que nos permite realizar configuraciones supersimples y poder así obtener el efecto visual de sombras en los elementos

sa propiedad, inicialmente vamos a trabajar con tres propiedades, que son la ubicación de la sombra en el eje horizontal, ese es el primer parámetro, el segundo es la ubicación de la sombra en el eje vertical y por último el color de la sombra.

[01:21] Y vamos a colocar 10 píxeles a la derecha, 10 píxeles hacia arriba, la parte superior, y el color va a ser negro

Corrigiendo que yo dije los 10 píxeles en el eje vertical hacia arriba, en realidad cuando colocamos positivo es hacia abajo. En realidad es en el sentido horario.

[01:58] Si yo coloco negativo, para que quede claro esto, si yo coloco aquí un negativo, esa sombra va a ser a la derecha pero superior

El cuarto parámetro, existe un cuarto parámetro que podemos adicionar a la propiedad, en realidad un tercero si es de izquierda a derecha, que es la difumación de ese elemento, o sea, en qué proporción va de un sólido, de un color más fuerte, para transparente.

[02:41] Es más o menos como el degradado, como ya lo hemos visto anteriormente, que también se define en píxeles. Por ejemplo, vamos a definir 10 y tenemos una pequeña difuminación. Vamos a aumentarle para que se note un poco más, a 30, y aquí ya se nota más, va de sólido para transparente nuestra sombra.

[03:09] Un quinto parámetro y un cuarto en el orden es el spread. El spread en inglés es expansión y es hasta dónde queremos que esa sombra llegue. Igual se define en píxeles y por ejemplo podemos colocarle 25 píxeles. Actualizamos aquí y fíjense que ahora nuestra sombra tiene una expansión mucho más grande.

Incluso podemos colocar más sombras y separarlos por una coma, entonces por ejemplo en este caso vamos a colocar -10 píxeles negativo, que sería ubicación a la izquierda, -10 píxeles negativos en el eje superior que sería hacia arriba y un color amarillo.

box-shadow: 10px 10px 30px 15px rgba(0,0,0), -10px -10px yellow;

Puedo incluso crear sombras internas usando la misma propiedad de box-shadow, solo que definiendo un parámetro que se llama inset. Para ello vamos a usar nuestro bloque los diferenciales aquí, entonces colocamos box-shadow, la propiedad inset, le definimos cero, cero, queremos que nuestro punto de referencia sea desde el centro de nuestro bloque, y aquí definimos el spread y el color, por ejemplo red.

[05:31] Vamos a ver que para todo nuestro bloque los diferenciales nos colocó un marco que en realidad es una sombra interna, una sombra hacia adentro.

box-shadow: inset 0 0 30px red;

 también, así como aplicamos sombras a elementos, imágenes, bloques, podemos hacerlo en los textos. Vamos a mostrar cómo aplicar en nuestro título principal, en nuestro bloque del título principal, aquí tenemos nuestro selector de clase, y en este caso llamamos a la función text de texto, shadow de sombra, text-shadow.

Los parámetros son prácticamente los mismos, entonces puedo colocarle 2 píxeles en el eje horizontal, 2 píxeles en el eje vertical y un color, el primero que se nos ocurre, rojo. Vamos a actualizar y aquí tenemos nuestro título con ese sombreado rojo. Incluso podemos colocarle aquí el spread por ejemplo para 10 píxeles y ahí tenemos nuestro efecto.

Existen páginas donde podemos entrenar, conocer y hasta probar esos efectos. Si buscan Box Shadow en Google, CSS Generator, y les aconsejo usar el de Mozilla.

¿Cómo hacer una sombra interna en un elemento?

box-shadow: inset 0 0 5px black;

¡Alternativa correcta! Sombra interna, negra, con una extensión de 5px.

box-shadow: inset -3px -3px 5px black;

¡Alternativa incorrecta! Las sombras internas no se pueden mover. Los valores en los ejes X y Y deben ser 0.

Lo que aprendimos en esta aula:

* Cómo manipular la opacidad de los elementos, con la propiedad CSS opacity
* Cómo manipular la opacidad de los colores
* Cómo agregar un sombreado alrededor de los elementos, con la propiedad CSS box-shadow
* Cómo agregar un sombreado en textos, con la propiedad CSS text-shadow

DISEÑO RESPONSIVO

nos falta ver un punto importante que es que nuestra página sea responsiva. ¿Qué significa que nuestra página sea responsiva? Nuestra página que sea responsiva significa que al momento de ser abierta desde cualquier dispositivo, sea un monitor grande, una pantalla de una notebook, una PC, o un monitor de un dispositivo celular o una tablet, consiga adaptarse al tamaño en forma adecuada.

Del inglés responsive, van a escuchar bastante este término, significa justamente eso, que nuestra página pueda comportarse en forma correcta en un dispositivo móvil principalmente. Debemos recordar que actualmente la gran cantidad de accesos, la mayoría de los accesos a internet son hechas desde el celular.

 Hemos trabajado durante todo el desarrollo de nuestra página como si nuestra página tendría que ser abierta únicamente de los navegadores por PC. No nos preocupamos en su momento para hacer los ajustes para ser abierta en dispositivos móviles. Incluso en varios lugares ajustamos para 940 píxeles y es por eso que tiene este comportamiento actualmente.

[05:48] Pero eso lo vamos a ajustar justamente en esta sección. Para ello, tenemos un recurso en el CSS, que son las medias queries, que básicamente lo que hacen, vamos a ir al final de nuestro CSS, las medias queries lo que hacen es permitirnos trabajar con excepciones o, dada una condición, aplicar algo diferente.

[06:20] La forma de definirlas es con un arroba y colocamos media, aquí colocamos screen, porque vamos a trabajar con el monitor, con el print de nuestra pantalla, con nuestra pantalla screen. Aquí colocamos un and y aquí colocamos un max-width, de ancho máximo. Dos puntos y colocamos 480 píxeles. ¿Qué le estamos diciendo en esa instrucción?

[06:58] Le estamos diciendo: si encuentras una pantalla con hasta 480 píxeles, o sea que ese es el límite máximo, aplicame, vamos a citar aquí solo un ejemplo, imaginen que vamos a aplicar algo en el body de nuestra página y le vamos a cambiar el background para red. ¿Entonces qué va a hacer?

¿Cómo marcar un elemento para que se presente de otra manera a través de una condición, en una pantalla de hasta 480px?

@media (max-width: 480px) {}

¡Alternativa correcta! Dentro de esta media query podemos crear un estilo visual que incluye pantallas de hasta 480px.

¿Cuál es la principal táctica para no perder mucho tiempo creando layouts responsivos?

Usar medidas proporcionales para todo

¡Alternativa correcta! Hoy en día es la mejor forma de crear cualquier contenido.

Crear un layout fluido que se adapte siempre

¡Alternativa correcta! Es una manera perfecta de evitar la repetición de trabajos.

Lo que aprendimos en esta aula:

* **Design responsivo**: Cómo ajustar el estilo de nuestra página según el tamaño de pantalla del dispositivo que acceda:
* Meta *etiqueta* de **Viewport**
* *Media Queries*