**RESPONSIVE WEB DESIGN (Diseño Web Adaptativo)**

A – BUILDING CAT PHOTO APP

Los elementos de encabezado <h1> a <h6> se utilizan para indicar la importancia del contenido por debajo de ellos. Entre menor sea el número, el contenido será más importante.

El elemento <p> es usado para crear un párrafo de texto

Los comentarios te permiten dejar mensajes sin afectar lo que se muestra en el navegador. Te permite crear código que no se ejecutará. Un comentario en HTML comienza con <!--, contiene tantas líneas de texto como quieras, y termina con -->

HTML5 tiene diferentes elementos que ayudan a diferenciar diferentes tipos de contenido. Estos elementos hacen tu código HTML más fácil de leer y ayudan con el Posicionamiento en buscadores (Search Engine Optimization - SEO) y accesibilidad.

La etiqueta <main> identifica la sección principal de la página.

Los elementos anidados deben colocarse dos espacios más a la derecha del elemento en el que están anidados. Este espacio se llama sangría (indentación en programación) y se utiliza para facilitar la lectura de HTML.

Los elementos <img> tienen una etiqueta de apertura sin una etiqueta de cierre. Se conoce como etiqueta de autocierre (self-closing tag).

Los atributos HTML son palabras especiales usadas dentro de la etiqueta de apertura de un elemento para controlar el comportamiento del elemento. El atributo src en un elemento img especifica la URL donde se localiza la imagen.  
Ejemplo: <img src="https://cdn.freecodecamp.org/curriculum/cat-photo-app/relaxing-cat.jpg">

Todos los elementos img deben tener un atributo alt. El texto del atributo alt es utilizado por lectores de pantalla para mejorar la accesibilidad y es mostrado en caso de que la imagen falle en cargar. Por ejemplo, <img src="cat.jpg" alt="A cat">

Puedes hacer un enlace que te llevará a otra página con el elemento anchor <a>. Por ejemplo, <a href='https://freecodecamp.org'></a> te llevará a freecodecamp.org.

El atributo target con el valor \_blank en la etiqueta de apertura del elemento anchor (a) sirve para que el enlace abra en una nueva pestaña.  
Ej: <a href="https://freecatphotoapp.com" target="\_blank">

Se puede utilizar el elemento <section>, para separar el contenido.

Para hacer una lista desordenada, se usa la etiqueta unordered list <ul> y la etiqueta <li> - list item para agregar elementos a una lista.  
Ejemplo:  
<ul>

<li>milk</li>

<li>cheese</li>

</ul>

El código de una lista ordenada, ordered list <ol>, es similar al de una lista no ordenada, pero los elemento de una lista ordenada aparecen enumerados.

El elemento <figure> representa contenido independiente y te permitirá asociar una imagen a una descripción. El elemento <figcaption> se utiliza para añadir una descripción o leyenda para describir una imagen anidada en un elemento figure. Por ejemplo, <figcaption>A cute cat</figcaption> añade la leyenda A cute cat. Para enfatizar palabras en el elemento figcaption, se envuelve en un elemento énfasis <em>.(lo pone en itálica) El elemento <strong> se utiliza para indicar que una parte de un texto es importante o urgente. (lo pone en negrita)

Con <form> se pone un formulario web para recolectar información de los usuarios. El atributo <action> indica, dónde deben enviarse los datos del formulario. Por ejemplo, <form action="/submit-url"></form> le dice al navegador que los datos del formulario deben ser enviados a la ruta /submit-url.

El elemento <input> te permite recolectar datos desde un formulario web de diferentes formas (son de auto-cierre). Hay diferentes tipos de elementos input, los cuales los puedes crear con el atributo <type>. Puedes crear fácilmente un campo de contraseña “password”, un botón de reinicio “reset”, ingresar texto “text” o un control para permitir a los usuarios seleccionar un archivo desde su computadora.

Para que los datos de un formulario puedan acceder al destino especificado en el atributo action, debes darle al campo de texto un atributo <name> y darle un valor que represente a los datos que están siendo enviados. Por ejemplo, puede utilizar la siguiente sintaxis para un campo de texto de dirección de correo electrónico: <input type="text" name="email">.

Un texto provisional <placeholder> nos da una idea de que tipo de información debemos escribir en un elemento input. Por ejemplo, <input type="text" placeholder="Email address">.

Para evitar que un usuario entregue su formulario incompleto, necesitas añadir el atributo “required” al elemento input. El atributo required no necesita ningún valor.

Utiliza el elemento <button> para crear un botón cliqueable. Por ejemplo, <button>Click Here</button> crea un botón con el texto Click Here. (averiguar types)

los elementos input y button son elementos inline, los cuales no aparecen en una nueva línea.

Puedes usar botones de radio <input type=”radio”> para preguntas en las que quieres que el usuario solo te dé una respuesta a partir de múltiples opciones. Para hacer que al seleccionar un radio button, el otro sé deseleccione automáticamente, ambos botones deben tener un atributo <name> con el mismo valor. Para distinguir entre las distintas opciones se coloca un <value> identificatorio.

Los elementos <label> se utilizan para ayudar a asociar el texto a un elemento input con el propio elemento input (especialmente para las tecnologías de asistencia como los lectores de pantalla). Por ejemplo, <label><input type="radio"> cat</label> hace que al hacer click en la palabra cat se seleccione el radio button correspondiente.

El atributo <id> se utiliza para identificar elementos HTML específicos. Cada atributo id debe tener un valor único, diferente a los demás valores id de la página. Ej: <label><input type="radio" id=”cat”> cat</label>

El elemento <fieldset> se utiliza para agrupar elementos input y label relacionados dentro de un formulario web. Los elementos fieldset son elementos de nivel de bloque, lo que significa que aparecen en una nueva línea. (lo mete dentro de un cuadro)

El elemento <legend> actúa como una leyenda/descripción, para el contenido del elemento fieldset. Le da a los usuarios un contexto sobre lo que deben ingresar en esa parte del formulario. (Le manda título al recuadro)

Los formularios Usualmente usan casillas de verificación <checkbox> para preguntas que puedan tener más de una respuesta.

Hay otra manera de asociar un texto con el elemento input. Puedes anidar un texto dentro de un elemento <label> y añadirle un atributo “for=” con el mismo valor del atributo id del elemento input. Ejemplo:

<input id="loving" type="checkbox"> <label for="loving">Loving</label>

Para hacer que un checkbox o un radio button este seleccionado por defecto, necesitas añadir el atributo “checked”. No hay necesidad de agregarle un valor al atributo checked. Simplemente necesitas añadir la palabra checked al elemento input, asegurándote de que hay espacios entre ella y los otros atributos.

Ahora añadirás un pie de página con el elemento <footer>.

B – BUILDING A CAFÉ MENU

HTML

You can add style to an element by specifying it in the style element and setting a property for it like this:

<style>

      h1{

        text-align: center;

      }

    </style>

selector1, selector2 {

property: value;

}

Para que el estilo de la página se vea similar tanto en móvil como en ordenador de escritorio o portátil, necesitas añadir un elemento meta con un atributo especial content:

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

El elemento <div> se utiliza principalmente para própositos de diagramación a diferencia de los otros elementos de contenido que has usado hasta ahora. Para definir alto, ancho se estila en CSS.

IMAGEN DE FONDO

body {

  background-image: url(https://cdn.freecodecamp.org/curriculum/css-cafe/beans.jpg);

  }

<article> elements commonly contain multiple elements that have related information.

<article>

            <p>Frencha Vanilla</p><p>3.00</p>

          </article>

<p> elements are block-level elements, so they take up the entire width of their parent element.

To get them on the same line, you need to apply some styling to the p elements so they behave more like inline elements.

inline-block elements only take up the width of their content.

You can use an <hr> element to display a divider between sections of different content. (clava línea horizontal) hr elements are self closing. The default value of a property named border-width is 1px for all edges of hr elements.

CSS

Hay que vincular el archivo styles.css para que los estilos se apliquen:

  <head>

    <meta charset="utf-8" />

    <link rel="stylesheet" href="styles.css"/>

    <title>Cafe Menu</title>

  </head>

Comments in CSS look like this:

/\* comment here \*/

height: alto  
width: ancho  
font-family: tipo de letra  
font-style: italic, bold, underlined  
font-size: tamaño letra  
background-color:  
background-image:  
border-color:  
color: color de la letra

Change the width property's value to be 80%, to make it 80% the width of its parent element (body).

If you want to center the div horizontally. You can do this by setting its margin-left and margin-right properties to auto. Think of the margin as invisible space around an element. Using these two margin properties, center the div element within the body element.

BLOCK-LEVEL ELEMENT ACTING LIKE INLINE-ELEMENT (Pone en la misma línea)

.item p{

  display: inline-block

}

The current width of the menu will always take up 80% of the body element's width. On a very wide screen, the coffee and dessert appear far apart from their prices.

Add a max-width property to the menu class with a value of 500px to prevent it from growing too wide.

ou can add a fallback value for the font-family by adding another font name separated by a comma. Fallbacks are used in instances where the initial is not found/available

h1, h2 {

  font-family: Impact, serif;

}

Pseudo-selector: link visitado

a:visited {

   color: gray;

    }.

Pseudo-selector: pasa mouse por arriba

a:hover {

  color: brown;

  }

Pseudo-selector: mientras hacés click en el link

a:active {

  color: white;

  }

Las imágenes se manejan como si fueran elementos in-line. Para que se manejen como block-level:

img{

  display: block;

  margin-left: auto;

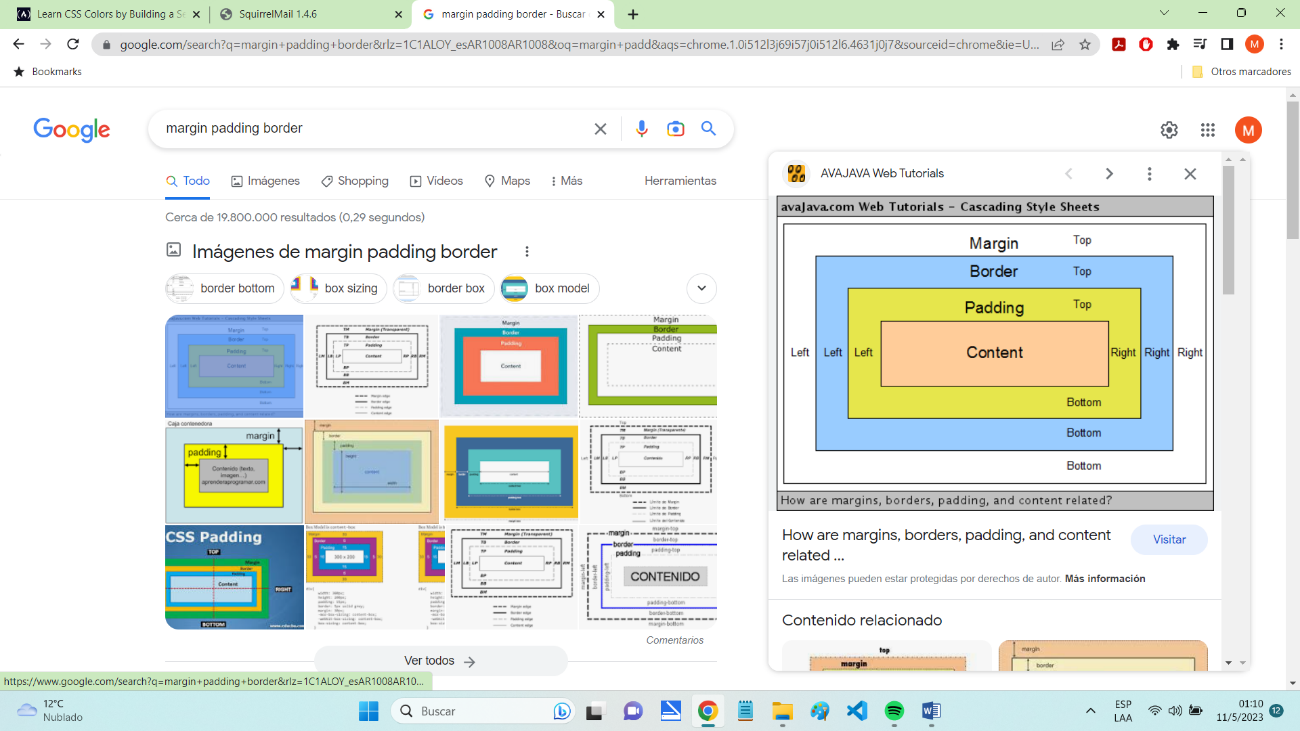
  margin-right: auto;

}

C – COLORED MARKERS

margin property to auto. This sets margin-top, margin-right, margin-bottom, and margin-left all to auto.

When the shorthand margin property has two values, it sets margin-top and margin-bottom to the first value, and margin-left and margin-right to the second value. The shorthand padding works similarly to the shorthand margin.



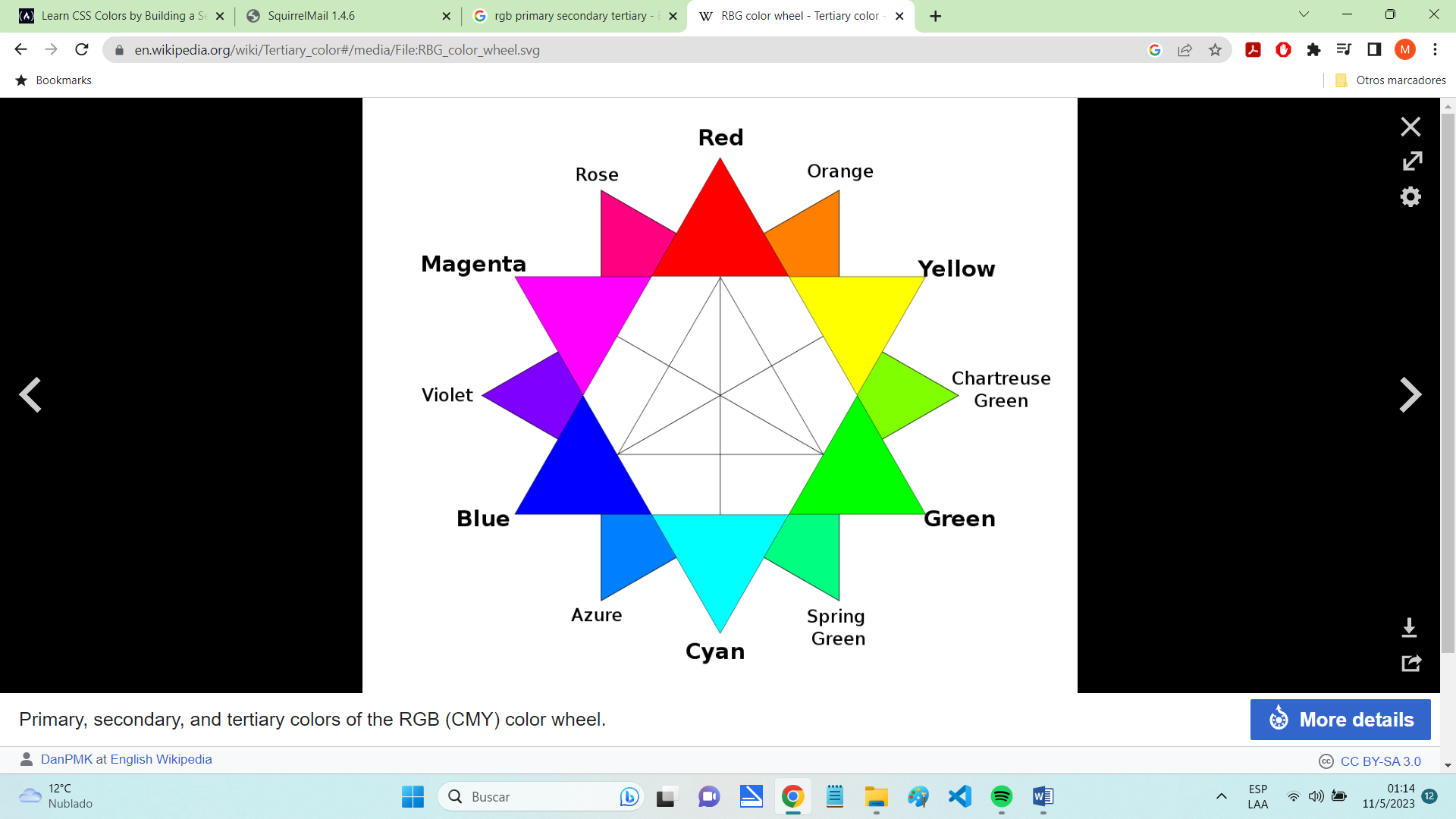
Multiple classes can be added to an element by listing them in the class attribute and separating them with a space. For example, the following adds both the animal and dog classes to a div element.

<div class="animal dog">

A function is a piece of code that can take an input and perform a specific action. The CSS rgb function accepts values, or arguments, for red, green, and blue, and produces a color:

rgb(red, green, blue);

Each red, green, and blue value is a number from 0 to 255. 0 means that there's 0% of that color, and is black. 255 means that there's 100% of that color.



hexadecimal or hex values. While hex values sound complicated, they're really just another form of RGB values. Hex color values start with a # character and take six characters from 0-9 and A-F. The first pair of characters represent red, the second pair represent green, and the third pair represent blue. For example, #4B5320.

The HSL color model, or hue, saturation, and lightness, is another way to represent colors. The CSS hsl function accepts 3 values: a number from 0 to 360 for hue, a percentage from 0 to 100 for saturation, and a percentage from 0 to 100 for lightness.

A gradient is when one color transitions into another. The CSS linear-gradient function lets you control the direction of the transition along a line, and which colors are used. One thing to remember is that the linear-gradient function actually creates an image element, and is usually paired with the background property which can accept an image as a value.

he linear-gradient function is very flexible -- here is the basic syntax you'll use in this tutorial:

linear-gradient(gradientDirection, color1, color2, ...);

Color-stops allow you to fine-tune where colors are placed along the gradient line. They are a length unit like px or percentages that follow a color in the linear-gradient function. For example, in this red-black gradient, the transition from red to black takes place at the 90% point along the gradient line, so red takes up most of the available space:

linear-gradient(90deg, red 90%, black);

If no gradientDirection argument is provided to the linear-gradient function, it arranges colors from top to bottom, or along a 180 degree line, by default. Color stops are automatically set at 0%, 50%, 100% etc.

With the CSS opacity property, you can control how opaque or transparent an element is. With the value 0, or 0%, the element will be completely transparent, and at 1.0, or 100%, the element will be completely opaque like it is by default.

Another way to set the opacity for an element is with the alpha channel. Similar to the opacity property, the alpha channel controls how transparent or opaque a color is. The rgba function works just like the rgb function, but takes one more number from 0 to 1.0 for the alpha channel:

the default display property for div elements is block. So when two block elements are next to each other, they stack like actual blocks. To position two div elements on the same line, set their display properties to inline-block.

The border-left shorthand property lets you to set the left border's width, style, and color at the same time.Here is the syntax:

border-left: width style color;

The box-shadow property lets you apply one or more shadows around an element. Here is basic syntax:

box-shadow: offsetX offsetY blurRadius spreadRadius color;

Here's how the offsetX and offsetY values work:

* both offsetX and offsetY accept number values in px and other CSS units
* a positive offsetX value moves the shadow right and a negative value moves it left
* a positive offsetY value moves the shadow down and a negative value moves it up
* if you want a value of zero (0) for any or both offsetX and offsetY, you don't need to add a unit. Every browser understands that zero means no change.

The height and width of the shadow is determined by the height and width of the element it's applied to. You can also use an optional spreadRadius value to spread out the reach of the shadow.

If a blurRadius value isn't included, it defaults to 0 and produces sharp edges. The higher the value of blurRadius, the greater the blurring effect is.

if you wanted to expand the shadow out further? You can do that with the optional spreadRadius value

ESTRUCTURA HTML

<!DOCTYPE html>. Esta cadena especial es conocida como una declaración y se asegura de que el navegador intente cumplir las especificaciones de la industria.

<html lang=”es”>

<head>

El elemento <title> determina lo que los navegadores muestran en la barra de título o en las pestañas del navegador.

Link a style.css

viewport

<meta charset=”UTF-8”>

</head>

<body>

<div>

<section>

</section>

</div>

</body>

<footer>

</footer>

</html>