

RAC

Tabla de contenido

Bienvenidos	3
Primeros pasos	3
index.php	3
index2.html	3
styles.css	3
script.js	4
rac.sql	4
leer.php	4
nuevo.php	4
borrar.php	5
modificar.php	5

Bienvenidos

Hola buenas, esta es una introducción en la que voy a describir mi programa. Este programa es una página web que utiliza html, css, js y php en la cual hay un carrito para comprar cosas, direcciones de contacto y un comentario para que la gente se registre y nos valore la página web. Todo está bien organizado y ordenado entre los 4 lenguajes que hemos utilizado para que se optimice al máximo el resultado. Antes de entrar a la página web hay un login el cual si no pones bien el usuario y contraseña te vuelve a mandar a la misma página sino te envía a la página web.

Creado con el Personal Edition de HelpNDoc: [Generador de ayuda multiformato con todas las funciones](#)

Primeros pasos

Creado con el Personal Edition de HelpNDoc: [Actualice sin esfuerzo sus archivos de ayuda WinHelp HLP a CHM con HelpNDoc](#)

index.php

Primero llamamos a php y recogemos los datos que queremos por POST e iniciamos la conexión con la base de datos. Después cerramos la condición que comprueba si los datos introducidos son correctos y te envía al sitio web. Cerramos php y tenemos que hacer un html con los datos que vamos a recoger para el login por el método POST al final dándole al botón de guardar.

Creado con el Personal Edition de HelpNDoc: [Crear ayuda HTML, DOC, PDF e imprime manuales desde 1 solo lugar](#)

index2.html

Primero ponemos el título de nuestra página web y creamos nuestra cabecera en la que van a aparecer 3 conectores de nuestra página web para ir a la información que necesitamos rápido. Después hay una imagen de nuestra tienda con un texto encima que incita a la compra de los productos, y un breve resumen de quien somos abajo. Luego hay un div por cada uno de nuestros productos el cual tiene una imagen con un botón de +info que te envía directamente a enviarme un correo, un breve resumen y el botón de meter al carrito el propio objeto hay uno por cada uno. Al final del div de los productos está la sección en la que se cuentan los introducidos al carrito y se contabilizan y aparecen por pantalla. Luego hay una sección que pone ven a conocernos con un botón con el link directo al mail y después una foto en el mapa de donde estamos. Luego hay una breve descripción de donde estamos situados con un botón que lleva a nuestra página del comentario en la cual se tiene que loguear. Después hay una marca de agua y el link para que conecte al script.

Creado con el Personal Edition de HelpNDoc: [Convertir documentos de Word en libros electrónicos: una guía paso a paso con HelpNDoc](#)

styles.css

Primero agregamos los valores a nuestro gusto de toda nuestra página como el tipo de letra color de fondo y demás. Después configuramos todos los botones de la página con su hover para que tenga un efecto cada vez que pasamos por encima de él y tener imagen si tienen una imagen. A continuación le ajustamos los valores máximos a la clase container y le ajustamos todos los valores del header de color ajuste del texto y espaciado de él mismo. Luego le damos los valores a hero que es la página principal de la web que sale la imagen repetida de la tienda para que se vea bien con el texto encima. Vamos luego a asignar color y posición a somos-rac que es el mini resumen de quien somos. Y procedemos a decorar todos los productos disponibles de la tienda con su breve explicación al lado y todo en 3 columnas definido en .verting. En ven-a-conocernos ajustamos a nuestro gusto la letra el tipo, color y los efectos del botón; y servicio es lo que hay justo debajo con un mapita el cual centramos a la página y le establecemos los márgenes. Esta parte después de nuestros productos es el footer el cual cambiamos después de color y de especificaciones de texto. Y el último que es .producto que son los productos del carrito la generalización de como aparecen ellos en él.

Creado con el Personal Edition de HelpNDoc: [Haz el cambio a CHM con la sencilla herramienta de conversión WinHelp HLP a CHM de HelpNDoc](#)

script.js

Primero empezamos nombrando el carrito y haciendo una función para agregar productos al carrito y verificando si el producto ya esta dentro y luego actualiza el carrito. La siguiente función es para vaciar el carrito, en la cual primero verifica si el producto esta en el carrito y cuantos hay y va eliminando uno a uno hasta eliminarlos todos, y actualiza el carrito. Y después, una función para actualizar la representación visual del carrito con el elemento de HTML donde se va a mostrar el carrito. Luego object.keys recorre todos los productos del carrito y crea un div para mostrar la información del producto. Div.inner muestra el nombre del producto deseado, la cantidad y un botón para quitar uno.

Creado con el Personal Edition de HelpNDoc: [Escribir eBooks para Kindle](#)

rac.sql

Primero si la base de datos no esta creada la creamos a traves de visual studio code con algunos valores para que la tabla no este vacia, pero antes de ponerle datos creamos los valores que va a tener la tabla de nuestra base de datos.

Creado con el Personal Edition de HelpNDoc: [Escribir folletos de EPub para iPad](#)

leer.php

Leer: primero llamamos al php e inicializamos con las variables a las que acceder a la base de datos y después conectamos para que no haya errores y preparamos el select para hacer la consulta a nuestra base de datos rac. Y luego creamos la tabla con un poco de HTML y CSS en la que va a consistir mi trabajo con al final los nombres de la tabla pero antes un botón para crear nuevos registros en nuestra tabla. Después hay un foreach para que cada consulta haya una variable valor por cada campo de nuestra tabla.

Creado con el Personal Edition de HelpNDoc: [Convierta sin esfuerzo su documento de Word en un libro electrónico: una guía paso a paso](#)

nuevo.php

Llamamos al php y comprobamos si recibimos los datos por POST para después recoger las variables de nuestra tabla para comprobar luego la conectividad a nuestra base de datos. Si conecta bien luego prepara el INSERT con nuestros campos e introducir los nuevos datos a "rac" los cuales se guardan en un array y redireccionamos al leer.php para cerrar php. Y empezamos con el html para cuando le demos al botón de crear introduzcamos los valores de usuario, contraseña, comentario y a continuación el botón de guardar para que los datos se guarden por el método POST.

Creado con el Personal Edition de HelpNDoc: [Generar EBooks de EPub con facilidad](#)

borrar.php

Llamamos a php y le ponemos las variables de conectividad con la funcion para conectar con la base de datos. Despues obtenemos el codigo del registro q queremos borrar y nos preparamos la sentencia para borrarlo para luego ejecutarla y redireccionarlo a leer.php.

Creado con el Personal Edition de HelpNDoc: [Productor gratuito de EPub](#)

modificar.php

Llamamos al php, recogemos las variables para conectamos a la base de datos y las variables de la tabla de nuestra base de datos. Despues de la correcta conexion a nuestra base de datos comprobamos si recibimos los datos por POST y preparamos el update con nuestros campos los cuales vamos a actualizar y lo ejecutamos para redireccionarlo a leer.php. Preparamos el select y ejecutamos la consulta que obtiene un resultado que es nuestro coche actualizado y cerramos php. Como en el php de nuevo volvemos a hacer el html como en el de nuevo.php para introducir los valores de coche.

Creado con el Personal Edition de HelpNDoc: [Edite y exporte sin esfuerzo documentos Markdown](#)
