

Ejercicios HTML5, CSS3 y JavaScript

Formación Sopra Steria

Formación **HTML5, CSS3 y JavaScript**

Bloque HTML5, CSS3 y JavaScript

Versión 1.2 del jueves, 20 de diciembre de 2018

Destinatario(s)

Historial

Versión	Fecha	Origen de la actualización	Redactado por	Validado por
1.0	20/12/2018		José Pedro Checa Gutiérrez Gabriel Patricio Del Rey Apollonio Juan Ignacio García Nicolás	

Índice

1.	Capítulo 2: Construyendo la Interfaz de Usuario(UI) Usando HTML5: Textos, Gráficos y Multimedia	4
2.	Capítulo 3: Construyendo la Interfaz de Usuario mediante HTML5: Organización, Valores de Entrada y Validación	6
3.	Capítulo 4: Entendiendo CSS Básico: Contenido, Posicionamiento y Estilos	7
4.	Capítulo 5: Entendiendo CSS Básico: Layouts	8
5.	Capítulo 6: Manejando Texto Flotante Usando CSS	9
6.	Capítulo 6: Entendiendo JavaScript y Fundamentos de programación	10
7.	Capítulo 7: Trabajando con Gráficos y Accediendo a Datos	11

1. Capítulo 2: Construyendo la Interfaz de Usuario (UI) Usando HTML5: Textos, Gráficos y Multimedia

1. (**<title>**,**<a href>**,**<p>**,**<h1>**): Se debe generar un título que sea: Página de venta de coches del Proyecto Escuela, además, se debe hacer una cabecera principal del anuncio con este texto: "El Coche con la Mejor Calidad-Precio". Y más abajo, en un párrafo, un texto donde aparezca el siguiente texto: "No esperes más, éste es el coche de tus sueños, porque tiene la mejor relación de calidad-precio, con este coche vas a alcanzar la felicidad y si no, accediendo al siguiente link, seguro que lo encuentras... La página a la cual ha de dirigirse nuestro link deberá ser

https://www.bmw.es/es/ssl/citaprevia/encuentra-tu-concesionario.html?&scp=true#/dlo/ES/es/BMW_BMWI_BMWM."
2. (****,****,**<i>**,****,**<center>**): Se debe cambiar el formato de texto de la página del ejercicio 1. Se modificará, tal cual se especifica a continuación:
 1. "coche de tus sueños" deberá aparecer en formato Negrita e Itálica o Cursiva.
 2. "mejor relación de calidad-precio" deberá aparecer en formato Enfatizado y Negrita.
 3. "felicidad" deberá aparecer en formato Strong.
 4. Hacer que el encabezado aparezca centrado.
 5. Hacer que el párrafo de texto aparezca centrado.
3. (**<fig>**): Se debe añadir una imagen a lo que se ha hecho en el ejercicio 2: Entre el encabezado que se ha hecho y el texto, se deberá incluir una foto del vehículo que se quiera (la única condición que se pide es que sea de la marca de la cual se ha enlazado con el link que aparece en el texto), la foto deberá tener altura 250 y ancho 600 y deberá estar centrada.
4. (**<figcaption>**): En la foto que se ha puesto, se quiere que aparezca un texto debajo de la misma en el que se lea: "Ir en un buen coche es ir feliz."
5. (**<footer>**, **
**): Se debe añadir un pie de página, a la página web del ejercicio 4, en el cual se deberá poner el símbolo de copyright y a su lado 2018.

Y además, se pide que el enlace aparezca a continuación del texto, pero en una línea aparte.
6. (**<video>**, **<source>**): Partiendo del Ejercicio 5 y suponiendo que se ha grabado un vídeo promocional para esta página Web de anuncio de coches, se quiere que se vea el marco donde se vería el vídeo del coche, en el lugar de la foto que se ha puesto en los anteriores ejercicios. El vídeo podrá ser reproducido automáticamente y deberán poder verse los botones de control de pausa/reproducción, parar y los de controlar el volumen y se llamará miVideoPromocional.mp4 (ese es el nombre que debe aparecer en la parte de la source o fuente, que, aunque no va a funcionar porque no existe tal vídeo, debemos saber cómo poner el vídeo). El tamaño de la pantalla donde se reproduce el vídeo deberá ser 300 de alto y 500 de ancho. El vídeo deberá aparecer centrado.

7. (**<svg>, <circle>, <rect>, <ellipse>, <polygon>, id**): Se va a trabajar con SVG, en un nuevo archivo de tipo .html. Se debe hacer las siguientes figuras y todas en la misma página, una debajo de otra, alineadas a la izquierda de la pantalla y con tamaño de contenedor para cada uno 130x200:
- a. Un círculo de radio 50, centrado en la coordenada (50,50), que tenga como identificador circuloOliva y el color de relleno que sea verde-oliva.
 - b. Una elipse de radio mayor en el eje X 100, radio menor en el eje Y de 50 y que esté centrada en (100,50) de color azul.
 - c. Un rectángulo de base 100 y altura 150, que tenga como color de relleno rojo, redondeando las puntas redondeadas, para el eje X un 80 y para el Y un 50.
 - d. Un rectángulo de base 100 y altura 150 de color verde.
 - e. Y triángulo negro de vértices A= (50,1), B= (0,100), C= (100,100)
8. (**Repaso donde se junta SVG y encabezados**): Se debe poner título a la página que se ha creado en el ejercicio 8, que será "Figuras con SVG", y cabecera principal, que será "Trabajando con SVG". Le daremos un apartado con encabezado tipo h2, para el círculo, otro para la elipse, otro para el triángulo y otro para poner los rectángulos, donde cada rectángulo tendrá un encabezado tipo h3. Por último, las subsecciones de círculo, elipse, triángulo y de los rectángulos irán separadas por una línea horizontales. (Opcional)

2. Capítulo 3: Construyendo la Interfaz de Usuario mediante HTML5: Organización, Valores de Entrada y Validación

1. (**<section>**, **<nav>**): Se debe hacer una web simple que tenga un título, encabezados y con mínimo 3 secciones. Se debe hacer al final de la página un menú de navegación, que tengan por nombre el nombre de las diferentes secciones que se hayan hecho.
2. (**<summary>**, **<details>**): Se debe hacer una página donde se haga uso de estas etiquetas de la forma que se quiera.
3. (**<table>**): Se pide hacer una tabla de 4 filas y 4 columnas, rellenas con el contenido que se quiera.
4. (****):
 - a. Se pide hacer una lista ordenada, con el contenido que se quiera, mínimo 5 elementos en la lista y que esté enumerada con 1,2,3...
 - b. Se pide hacer otra lista ordenada, con el contenido que se quiera, mínimo 5 elementos en la lista y que esté enumerada con letras a,b,c...
 - c. Se pide hacer otra lista ordenada, con el contenido que se quiera, mínimo 5 elementos en la lista y que esté enumerada con números romanos i,ii,iii...
5. (****):
 - a. Se pide hacer una lista desordenada, con el contenido que se quiera, mínimo 5 elementos en la lista y que cada elemento de la lista empiece por un cuadrado negro.
 - b. Se pide hacer otra lista ordenada, con el contenido que se quiera, mínimo 5 elementos en la lista y que cada elemento de la lista empiece por un círculo negro.
 - c. Se pide hacer otra lista ordenada, con el contenido que se quiera, mínimo 5 elementos en la lista y que cada elemento de la lista empiece por un círculo en blanco.
6. (**Formulario**): Se pide hacer un formulario, con estos campos, uno detrás de otro, todos ellos deberán ser rellenados de forma obligatoria, sino saldrá un mensaje que diga que no se ha rellenado algún campo a la hora de enviarlo:
 - a. Nombre.
 - b. Apellidos.
 - c. DNI.
 - d. Correo Electrónico, en este campo se deberá verificar si es o no un correo electrónico lo que introduce el usuario.
 - e. Fecha de nacimiento, donde se despliegue un calendario que le permita al usuario elegir la fecha que quiera.
 - f. Mes y año de contratación, que aparezca un selector o contador que permita seleccionar el mes del año que se contrató a la persona.
 - g. Semana y año de contratación, que aparezca un selector o contador que permita seleccionar la semana del año y el año, que se contrató a la persona.
 - h. Contraseña, el cual aparezca en formato de contraseña, es decir, que no se vea lo que se ha escrito.
 - i. Enviar, que aparezca un botón para enviar el formulario.

3. Capítulo 4: Entendiendo CSS Básico: Contenido, Posicionamiento y Estilos

1. **<LINK>** - Dado el fichero HTML `capitulo4_ejercicio1.html`, se debe crear el fichero `capitulo4_ejercicio1.css` y relacionarlo con el fichero HTML.
2. En el fichero creado, `capitulo4_ejercicio1.css`, se debe generar el código necesario para que la fuente este en color azul, pero la fuente de la clase *opinion* este en color verde.
3. Se debe generar una nueva línea de texto ("*Esta línea es para el ejercicio 3*") en el archivo HTML, `capitulo4_ejercicio1.html`, que debe tener una fuente de la familia *Garamond*, con un tamaño de 20px, color marrón y en cursiva.
4. **INLINE FLOW Y BLOCK FLOW** - Se debe generar la lista de frutas, del fichero HTML `capitulo4_ejercicio4`, tanto en vertical como en horizontal, con un margen de 2px y un padding de 0.5em.
5. Dado el fichero HTML `capitulo4_ejercicio5.html`, se debe manejar el contenido desbordante de forma que se pueda hacer scrolling, sea visible y sea oculto.
6. Al igual que en el ejercicio anterior, se debe manejar el contenido desbordante de forma que esté oculto. (Opcional)

4. Capítulo 5: Entendiendo CSS Básico: Layouts

1. **FLEX-DIRECTION** – Se debe invertir el orden en el que aparecen las cajas del fichero HTML, capitulo5_ejercicio1.html. Se debe usar la función flex-direction.
2. **FLEX-WRAP** – En el fichero capitulo5_ejercicio1.html, generar la propiedad en las cajas para que no se modifique su tamaño al minimizar la ventana. Se debe usar la función flex-wrap. (Opcional)
3. **FLEX-FLOW** – Realizar los ejercicios 1 y 2 empleando la función flex-flow. (Opcional)
4. **FLEX-ORDER** – En el fichero capitulo5_ejercicio1.html, ordenar las cajas de forma que el orden sea el siguiente: Perro, Conejo, Loro, Gato. Se debe usar la función flex-order. (*Nota: esta función solo es aplicable en Internet Explorer*)
5. Generar una cuadrícula de los elementos: Perro, Gato, Loro, Conejo, dado el fichero HTML capitulo5_ejercicio5.

5. Capítulo 6: Manejando Texto Flotante Usando CSS

1. **COLUMN-COUNT** - Dado el fichero HTML, Capitulo6_Ejercicio1.html, se debe dejar el texto dividido en cuatro columnas.
2. **COLUMN-GAP** - A partir de la solución del ejercicio anterior, se debe crear un espaciado de columnas de 40px.
3. **COLUMN-RULE** - A partir de la solución del ejercicio anterior, se debe crear una línea verde sólida de 2px que separe las columnas.
4. **COLUMN-RULE-STYLE** - A partir de la solución del ejercicio 2, se debe crear una línea discontinua que separe las columnas.
5. **HYPHENS** - Dado el fichero HTML, Capitulo6_Ejercicio1.html, se debe dejar el texto justificado separándose por guiones a final de línea. (*Nota: solo se puede usar en el navegador Mozilla Firefox*)

6. Capítulo 6: Entendiendo JavaScript y Fundamentos de programación

1. **ALERT & CONSOLE.LOG("TEXT")**- Se crea un botón que, al ser clicado, llame a una función, muestre una ventana de alerta con un mensaje y además lo muestre por la consola del navegador (ver `console.log("texto")`).
2. **\$(#ID).HIDE() & \$(#ID).SHOW()** – Se crea un elemento con texto en su interior y un botón con la capacidad de llamar a una función, se debe mostrar el contenido (si el texto está oculto) y ocultar el contenido (si el texto se está mostrando) de este elemento.
3. **.VALUE & .INNERHTML** – Se crean dos elementos de entrada y un botón. El usuario introducirá un número en cada campo de entrada y, al hacer clic en el botón, una función de JavaScript sumará estos dos valores y mostrará el resultado por pantalla.
4. **CREATEELEMENT & APPENDCHILD** – Se crea un botón que, al ser pulsado, añada frases adicionales al contenido de la página, utilizando funciones.
5. **\$(#ID).MOUSEOVER() & \$(#ID).MOUSEOUT()** - Utilizando jQuery, se cambia de color y tamaño de un texto, dependiendo de si el ratón está encima de éste o no, utilizando funciones. (Opcional)

7. Capítulo 7: Trabajando con Gráficos y Accediendo a Datos

1. **CANVAS** - Se crea un canvas que contenga un rectángulo de color rojo.
2. **CANVAS** - Se crea un canvas que contenga una línea diagonal desde la esquina superior izquierda hasta la esquina inferior derecha. (Opcional)
3. **CANVAS** - Se crea un canvas que contenga un círculo centrado en su interior. (Opcional)
4. **DOCUMENT.TITLE** - Se crea una función de JavaScript que recoja el título de la página (el contenido en <title>) y lo muestre en el <body> como "El título de esta web es 'Ejercicios de JS'."
5. **DOCUMENT.TITLE** - Se crea un campo de entrada de texto y un botón; programar una función de JavaScript que recoja el texto del campo de entrada y cambie el título de la pestaña de la página.
6. **SPLIT** - Se crea un elemento <div> que contenga la siguiente cadena "uno;dos;tres;cuatro;cinco"; se crea también un botón que, al ser clicado, recoja la cadena de texto mencionada, y la convierta en un array de longitud 5 que contenga cada palabra por separado y las muestre individualmente una tras otra en pantallas de alerta.
7. **FILE** - Se crea la posibilidad de buscar archivos desde la web y mostrar el nombre del archivo seleccionado en una alerta. Solo se deben permitir archivos de tipo 'txt'; si se selecciona otro formato, el mensaje de alerta debe mostrar un texto que indique que la extensión es errónea.
8. **FORM VALIDATION** - Se crea un formulario al que se le deba introducir un número con la máscara XXX-XX-XX-XX. No se deben poder escribir letras ni símbolos y los guiones deben generarse automáticamente mientras se introducen los números.
9. **COOKIES** - Se utilizan cookies para recordar un texto que introduzca el usuario en un campo de entrada.
10. **.MATCH(REGEX)** - Se crean dos campos de entrada y un botón. Generar una función en JavaScript que tome los dos campos, compruebe si son números. Si son números, se debe mostrar si el primer número es mayor que el segundo; Si no lo son, se debe mostrar un mensaje aviso. (Opcional)