

Memoria Documental

Índice

1. Breve resumen de la indexación del documento.
2. Recursos externos utilizados.
3. index.html.
4. index.css.
5. programacion.html.
6. programacion.css.
7. inscripcion.html.
8. inscripcion.css.

1. Breve resumen del documento.

Todos los documentos .html están creados con el nombre de cada documento en la etiqueta <Head> para que cuando nos encontremos en cada página muestre su nombre en lugar del puerto del local host que estamos utilizando. Además cada uno tiene su propio archivo .css para ayudar a hacer el código más legible y evitar la repetición de clases o id o elementos cuando damos estilos el documento, para ello he creado una carpeta llamada style.css y dentro hay tres archivos index.css, programacion.css y inscripcion.css que están enlazados con sus correspondientes documentos utilizando la etiqueta <link>

<Head>

Aquí he enlazado los archivos .css, y las fuentes y los iconos utilizando la etiqueta <link>, la cabecera es igual en todos los archivos.

<Header> y <Footer> tienen el mismo contenido en todas las páginas he utilizado <div>, , <H1> y <p>

<nav> formada por una lista desordenada de 3 enlaces utilizando la etiqueta <a> con la ruta a sus correspondientes urls, cada página está conectada a su enlace mediante el href atributo.

2. Recursos utilizados

Carpeta de recursos de imágenes y video procedente del archivo .zip proporcionado.

Colores proporcionados en el documento.

Header: #CCEAF4

Nav:: #114358

Barra media (en index dónde se realiza.....; y en programación: Programación y días...
#648da0

Footer: negro.

Caption : #0A3143

Primera fila tabla: #FF3B3F ,

Filas pares tabla: rgb(241, 121, 121),

Pie de la tabla: #FF3B3F ,

H1 header : #A9A9A9 ,

a: #F2AA1F,

Iconos (Font Awesome) enlazados mediante cdn.

```
<!-- font awesome -->
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.2.1/css/all.min.css"
integrity="sha512-MV7K8+y+gLIBoVD59lQIYicR65iaqukzvf/nwasF0nqhPay5w/9lJmVM2hMDc
nKlOnMGCdVK+iQrJ7lzPJQd1w=="
crossorigin="anonymous" referrerpolicy="no-referrer" />
```

#icono de localización.

```
<i class="fa-sharp fa-solid fa-location-dot">
```

#icono de reloj

```
<i class="fa-solid fa-clock">
```

Fuentes(Google Fonts)

```
<!-- fuentes -->
<link rel="preconnect" href="https://fonts.googleapis.com"><link
rel="preconnect" href="https://fonts.gstatic.com" crossorigin><link
href="https://fonts.googleapis.com/css2?family=Lato:wght@700&family=Libre+Baskerville:wght@400;700&family=Lora:ital,wght@0,400;0,700;1,500&family=Poppins:ital,wght@0,200;0,400;0,700;1,400&display=swap" rel="stylesheet">
```

3.Index.html

El body está dividido en tres partes principales que son <Header>, <Main>, <Footer>, y dentro del main he dividido el documento en tres <Section>.

<header>

<nav> formada por una lista desordenada de 3 enlaces utilizando la etiqueta <a> con la ruta a sus correspondientes urls, cada página está conectada a su enlace mediante el href atributo.

< section 1>

Contiene un <article> con un párrafo y la imagen, para los saltos de línea he utilizado
.

< section 2 >

Contiene 2 <article> con un h2 cada uno para los enlaces he utilizado la etiqueta <a>, en el segundo artículo he utilizado <div> para separar las diferentes imágenes con sus correspondientes pie de foto.

4.Index.css

Par hacer algunos estilos generales como los tipos de fuente he utilizado elementos, pseudo elementos y clases en su nomenclatura, en el centrado de los elementos he jugado con las propiedades padding, margin y width, para el alineado de las imágenes a la misma altura he utilizado el float, propiedades del texto.

5.Programacion.html

< section 1> contiene la tabla con los horarios y los iconos(font Icon), el título de la tabla está hecho con la etiqueta <caption> y las cabeceras de cada columna son <th> table head y dentro hay un <tbody> que contiene cada fila de la tabla, en la parte de abajo he puesto un < footer> para indicarle que es el pie de la tabla..

En la < section 2> he utilizado listas desordenadas anidadas y dentro con la etiqueta en los horarios he añadido la negrita.

6.Programacion.css

Aquí he utilizado algunas clases, pero también he utilizado id y pseudo elementos, para identificar cada elemento.

Para los estilos de las fuentes he utilizado propiedades como **font-size > tamaños, font-weight > interlineados, color > colores de las fuentes.**

Los fondos están hechos con **background- color> que colorea el fondo, alineados con padding, margin, align- content > para centrar los textos .**

Las **columnas** están alineadas **con la propiedad float.**

Para eliminar la decoración de las listas he utilizado la propiedad list- style > con valor none.

7.Inscripcion.html

Aquí he creado tres secciones.

< section1>

Contiene el título del formulario que lo he puesto con un H2.

Con la etiqueta <form> he creado todo el formulario para que el envío cuando se aprete el boton tenga todos los apartados en su interior, dentro de él he creado 2 <section> que a su vez contiene <fieldset > cada uno contiene una etiqueta <legend> utilizada para dar título al contenido, para crear las leyendas he utilizado los label con atributos para que el usuario sepa que tiene que introducir en el formulario, y para las cajas he utilizados la etiqueta <input> con atributos como require > es un campo requerido, maxlength y minlength para indicar el numero de caracteres minimo y maximo que se pueden utilizar, el pattern > para decirle que reglas de escritura se pueden o no introducir y así restringir cada input al contenido exigido (llamadas regex) reglas de expresión regular, y les he creado un #id como identificador único para la hoja de estilos.

< section 2>

Contiene un <fieldset> por cada bloque de preguntas que hay que responder, cada uno de ellos contiene las etiquetas <input> y <label>, en estos campos el atributo type juega el papel de decirnos que tipo de selección vamos a hacer yo he utilizado el type= "checkbox", una cajita que pone un check cuando se marca y en la ultima parte el type = radio, que nos deja seleccionar varias opciones también, además todos los campos son obligatorios gracias al required de los <fieldset>.

Botón de Enviar

No tenía muy claro que pudiera crear un botón así que he utilizado un input con el atributo type = submit, esto hace que haga la función de enviar y valide si los campos están correctos.

8.Inscripcion.css

Aquí he utilizado propiedades como background y color para colorear textos y fondos, los margin y padding para centrar divs y textos, propiedades para el texto como text-shadow para añadir sombras en la caja de los legend, para los colores de los input que cambien cuando los campos están rellenos correctamente he añadido las propiedades valid e invalid.

También he utilizado espaciado entre letras con letter-spacing, bordes redondeados en los input con border-radius

En el botón para enviar el formulario he creado una transición y se transforman en mayúsculas las letras al pasar el ratón con las propiedades text- transform con valor uppercase y transition para decirle cuanto tiempo tarda en hacer el cambio de color.