**CLIPS PARA BROCHURE**

**HTML**

|  |  |
| --- | --- |
| **INTRODUCCIÓN** | Quiero mostrarte la primera capa, la que está allí, sin que lo sepamos, pero es aquella de vital importancia para que todo funcione a las mil maravillas, te estarás preguntando ¿De qué se trata esto?, pues te cuento que en este curso hablaremos sobre,  **PRESENTACION**  el campo de los lenguajes de programación, su inicio, todo lo que es y el por qué es importante, para poder empezar un largo camino de aprendizaje en la programación, este es el campo que nos conduce en primera instancia a crear todo lo que constituye tecnología digital, es decir videos, imágenes, contenidos y demás que vemos siempre, día a día y en todo momento.  Estoy hablando de un lenguaje base para poder construir todo tipo de sitio web, este es el HTML…. Lenguaje de programación que nos ayuda a crear una comunicación más directa y entendible con todos a través de internet. Este lenguaje es el encargado de incorporar el texto.  Mi nombre es Tatiana Vega y quiero que te embarques conmigo por el camino de HTML, que va a ser de gran ayuda para tus programaciones digitales hoy por hoy y en el futuro |

|  |  |
| --- | --- |
| **NAVEGACIÓN** | **PRESENTACIÓN**  **TELEPRONTER**  Los menús son elementos que encontramos en todas las páginas en las que accedemos. Normalmente en la parte superior e inferior.  Tengo aquí un menú inferior de una página donde los elementos se marcan como si fueran un pie de página  Entonces para poderlo marcar como pie de página, lo que debo hacer es envolver en mi elemento footer.  **EDITOR**  Lo haremos en un ejemplo real, para verlo al final como el ejemplo finalizado que mostré anteriormente, **(Computador)**  primero el signo menor que, luego footer, y signo mayor que, **(Telepronter)** en la siguiente línea, **(Computador)**  signo menor que, p, mayor que, menor que, la a, espacio, Href, igual, abro comillas, el link, **(Telepronter)** en este caso tengo uno de ejemplo, **(Computador)** cierro comillas, mayor que, primero la palabra política, signo menor que, slash, a, mayor que, y cierro mi etiqueta p, con signo menor que, slash, p, y mayor que. **(Telepronter)** Repetiré esta misma acción dos veces más, **(Computador)** pero cambiaré aquí por privacidad…y aquí por términos y condiciones. **(Telepronter)** Ahora como resultado observamos que, **(Computador)** al pasar el cursor por encima de cada uno, cambia, **(Telepronter)** como si fuera una manito la cual nos indica que cada uno de ellos es un enlace.  Ahora debo convertirlo en elementos de lista, así que cambiaré un poco, la idea, del elemento de pie de página, **(Computador)** lo eliminaré y pondré un elemento llamado nav, **(Telepronter)** que quiere decir navegación, igualmente debe estar envuelto con **(Computador)** signo menor que y al final mayor que, luego, un enter y pondré, signo menor que, luego ul, **(Telepronter)** que me indica que es una lista desordenada, **(Computador)** espacio, class, igual, abro comillas, en este caso pondré classy, cierro comillas, signo mayor que, y ahora en cada uno de estos elementos **(Telepronter)** que están como párrafos los eliminaré **(Computador)** e indicaré que son elementos de lista con li, **(Telepronter)** tanto en el cierre como en el final. **(Computador)** Por último, cierro mi lista, desordenada con signo menor que, slash, ul, mayor que y **(Telepronter)** más abajo cierro la navegación, **(Computador)** signo menor que, slash, nav, y mayor que.  **(Telepronter)** Ahora veremos que nuestros tres textos cortos hacen parte de una lista desordenada.  Ya para darle un efecto visual como lo vimos en un comienzo, con el fondo de diferente color y demás debemos utilizar, CSS, tema que tomaremos en un curso diferente.  <footer>  <p><a href="esteesunejempllo.com">Política</a></p>  <p> <a href="esteesunejempllo.com">Privacidad</a></p>  <p> <a href="esteesunejempllo.com">Terminos y condiciones</a></p>  </footer> |

**CSS**

|  |  |
| --- | --- |
| **DISEÑAR DOCUMENTOS DE FORMA COHERENTE** | **PRESENTACIÓN**  CSS, Nos da la habilidad, de que nuestras páginas web se vean mucho más bonitas, podemos tener la capacidad de crear lo que deseamos, haciendo más amigable lo que ven nuestros ojos.  Quiero que me acompañes a descubrir este mundo de adornos, lo llamo yo, encontrarás los códigos perfectos para aplicar en tu página y verás como resalta a comparación de las demás  Además de crear lo que desees, también puedes actualizar las páginas web ya existentes, haciéndolas más novedosas.  Así que no te quedes sin saber y apuntar, lo que te voy a enseñar.  Acompáñame por este bello camino de descubrir nuestro lenguaje CSS  Yo soy Tatiana Vega y estaré contigo a lo largo de este camino de aprendizaje mutuo |

|  |  |
| --- | --- |
| **SELECTORES** | Creo que hay una pregunta rondando por nuestras cabezas en estos momentos ¿Por qué en la lección anterior al indicar un cambio en cada elemento se eliminaba el cambio anterior?  **PRESENTACIÓN**  Bueno, tenemos cuatro selectores ya vistos, universal, tipo, clase y por último id, en ese orden es que tienen valor. A que me refiero con valor, pues que en primer lugar el universal tiene un valor igual a cero, y de ahí en adelante en el mismo orden en que te los nombre van aumentando su valor, por ese motivo al aplicar un cambio en el tipo, se elimina el universal, al aplicar un cambio en la clase, se elimina el de tipo, y al aplicar el id, se eliminan todos los demás, ya que el id es que representa más valor en esta escala.  **CODEPEN**  [**https://codepen.io/Tatiana-Vega/pen/ZEpjqoN?editors=1100**](https://codepen.io/Tatiana-Vega/pen/ZEpjqoN?editors=1100)  La mejor forma de comprenderlo es aplicándolo, aquí vuelvo a mi ejercicio anterior, tenemos el HTML, exactamente igual y empezaremos con el selector universal para h1.  Primero asterisco, corchete, en el siguiente espacio color, dos puntos, Pink, punto y coma y en el último nuestro cierre de corchete. Y en nuestro resultado obtenemos todo el texto en color rosado.  Ahora nuestro selector de tipo, aplicado a h1, entonces h1, seguido de corchete, siguiente renglón color, dos puntos, red, punto y coma y corchete abajo. Nuestro resultado en el h1 elimina el color rosado y cambia a un color rojo.  Continuemos con el mismo ejercicio en h1, ahora con nuestro siguiente selector de clase, entonces indicamos un punto, luego nuestra clase que en este caso es post, h1, corchete, más abajo color, dos puntos, yellow, punto y coma, y abajo corchete para cerrar, ahora nuestro título h1 es de color amarillo.  Por último, nuestro selector ID, recuerda que empezamos con #, luego post, luego h1, corchete, más abajo color, dos puntos, blue, punto y coma y abajo corchete.  Pero ¿Qué pasa? ¿Por qué no se aplica?.... bueno ya lo sé. Debemos en este punto volver al HTML, y aquí encontramos el error, no hemos creado un selector ID, para hacerlo debemos hacerlo justo después de article, daremos un espacio, escribimos id, igual a, abro comillas post y cierro comillas, en este caso puse el mismo id que la clase, pero puedes indicar un id diferente si así lo deseas.  Ahora, como resultado final se ve nuestro último cambio, ahora el título h1 es de color azul.  Ves lo sencillo que es, está es la forma de eliminar según el valor de cada selector, así que |

**JAVASCRIPT**

|  |  |
| --- | --- |
| **BIENVENIDOS** | **TELEPRONTER**  JavaScript, cuando escuches este lenguaje de programación, debes haber avanzado con tus bases, es decir html y css, luego de tener un conocimiento en estos dos lenguajes. En ese momento nos adentraremos al tercer lenguaje de programación, Javascript, el que nos dará la interactividad que necesitamos agregar en la construcción de nuestra web. Aquí en este curso de formación básica de Javascript, pondremos en marcha, proyectos reales que nos ayudaran a comprenderlo.  Mi nombre es Tatiana Vega, y quiero que después de este curso quedes completamente listo para avanzar en tus proyectos aplicando los tres lenguajes de programación HTML, CSS y en el que nos enfocaremos aquí para cumplir esos propósitos, Javascript. |

|  |  |
| --- | --- |
| **RELOJ EN MOVIMIENTO** | **(telepronter)** Vamos a construir un reloj analógico, para hacerlo debemos apoyarnos de nuestros tres lenguajes de programación, cada uno de ellos cumple una función específica en su conformación. Así de esta manera lograremos el objetivo deseado que es verlo en movimiento.  **Computador**  Para empezar, tendremos un documento HTML, en este caso le di el nombre de Ejercicio 4.2.  En el cabeza enlazado a mi CSS a trabajar que nombre reloj y en el cuerpo enlazado al documento Javascript, también nombrado reloj.  Debo dar un contenido principal en donde voy a incorporar un gráfico vectorial, que será el cuerpo de nuestro reloj.  Más adelante, indicaré el elemento g, que el encargado de agrupar objetos, en este caso voy a crear 4 objetos. Uno para la figura del reloj como tal, otro para los minutos y otro para los segundos.  Comenzando con la figura, pondré un ID, y daré valores tanto para el círculo externo, es decir el más grande y se hace a través de coordenadas cx para el eje x del punto central, y cy para el eje y, r en este caso hace referencia al radio del círculo.  Luego la ruta, a la que debo asignarle una clase, que llamaremos marcas de la hora y luego un diámetro donde se debe primero indicar la letra m, que quiere decir movimiento, es decir moveremos la ruta a cierta distancia indicada aquí, y la v, nos da el movimiento vertical, también indicando en valores posteriores. Y por último indico valores para el círculo pequeño del medio, tanto cx, cy y radio. Debo cerrar el elemnto g.  Y realizaré la misma acción para la hora, solo indicando valores para el camino y el círculo, con una clase denominada caja de tallas, que vendrán a ser las manecillas del reloj y sus respectivos valores.  Misma acción para minutos, luego segundos, y finalizó el html con los respectivos cierres de etiquetas.  <DOCTYPE html>  <html lang="es">      <head>          <meta charset="UTF-8">          <meta name="viewport" content="width=device-width, initial-scale=1">              <title> reloj analógico </title>              <link href="C:\Users\Tatiana\Desktop\Curso Javascript\Css\reloj.css" rel="stylesheet" />        </head>      <body>            <script src="C:\Users\Tatiana\Desktop\Curso Javascript\Javascript\reloj.js"defer></script>      <main class="main">      <div class="clockbox">      <svg id="clock" xmlns="https://www.w3.org/2000/svg" width="600" height="600" viewBox="0 0 600 600">      <g id="face">        <circle class="circle" cx="300" cy="300" r="253.9"/>      <path class= "hour-marks" d="M300.5 298V142"/>      <circle class="mid-circle" cx="300" cy="300" r="16.2"/>      </g>      <g id="hour">            <path class="hour-arm" d="M300.5 298V142"/>          <circle class="sizing-box" cx="300" cy="300" r="253.9"/>      </g>      <g id="minute">          <path class="minute-arm" d="M300.5 298V67"/>          <circle class="sizing-box" cx="300" cy="300" r="253.9"/>      </g>      <g id="second">          <path class="second-arm" d="M300.5 350V55"/>          <circle class="sizing-box" cx="300" cy="300" r="253.9"/>      </g>      </svg>  </div>  </main>      </body>    </html>  Si revisamos la consola obtendremos este esquema de un círculo negro, tal cual como es la forma de un reloj, luego tendré que empezar a digitar valores en CSS, para que nuestro reloj empiece a tomar la forma  **CSS Computador**  El documento como lo ves, está en blanco.  Lo primero que haré es darle diseños, lo pondré como comentario y luego, al contenido principal le daré sus respectivos valores, para el contenido en este caso centrado, la alineación de elementos al medio, monitor, relleno y añlto.  Luego el diseño de la caja de reloj, a la cual solo le daremos un único valor de altura.  Para continuar daremos un espacio para los estilos del reloj, voy a crear el comentario solo con el fin de llevar un orden, y empezaré con los estilos del círculo, dándole una anchura al trazo, el límite del inglete, el relleno de la figura en ninguno y la el color del trazo en negro, color que también le daremos al círculo del medio.  Continuamos con las marcas de la hora que fue la primera clase creada en el html. Donde asignaremos valores para ancho de trazo, límite del inglete, relleno y color.  Acción que repetiremos aquí tanto para el brazo de la hora, los minutos y los segundos, recordando que el brazo de la hora debe ser más ancho y más corto.  Muy bien, definiendo estos valores continuamos dándole a la caja de talla un relleno de ninguno, es decir quedará en transparente.  /\*diseños\*/  .main {      justify-content: center;      align-items: middle;      display: flex;      padding: 3em;      height: 90vh;    }  .clockbox,  #clock {      height: 100%;  }  /\*estilos del reloj\*/  .circle {      stroke-width: 10;      stroke-miterlimit: 11;      fill: none;      stroke: black;    }  .mid-circle {      fill: black;  }  .hour-marks {      stroke-width: 10;      stroke-miterlimit: 11;      fill: none;      stroke: black;    }  .hour-arm {  stroke-width: 19;  stroke-miterlimit: 12;  fill: none;  stroke: black;  }  .minute-arm {      stroke-width: 11;      stroke-miterlimit: 10;      fill: none;      stroke: black;    }  .second-arm {      stroke-width: 5;      stroke-miterlimit: 10;      fill: none;      stroke: black;    }  /\*caja para los brazos que es transparente \*/  .sizing-box {      fill: none;  }  /\* realiza la acción de hacer girar los brazos alrededor del punto del centro. \*/  /\* que se vea animado y funcione \*/  #hour,  #minute,  #second {      transform-origin: 300px 300px;     transition: transform .5s ease-in-out;  }  **JAVASCRIPT Computador**  const HORARIO = document.querySelector("#hour");  const MINUTERO = document.querySelector("#minute");  const SEGUNDERO = document.querySelector("#second");  let hrPosition = 22;  let minPosition = 140;  let secPosition = 280;  HORARIO.style.transform ="rotate(" + hrPosition + "deg)";  MINUTERO.style.transform ="rotate(" + minPosition + "deg)";  SEGUNDERO.style.transform ="rotate(" + secPosition + "deg)";  **PRESENTACIÓN-TABLET**  La primera regla es la de comentarios, en CSS, manejamos los comentarios con slash, y asterisco, pues esta misma regla se debe aplicar en Javascript, pero para comentarios que son de dos líneas o más, empezaría con slash, asterisco, el texto, asterisco y slash. Y cuando el comentario es de una sola línea se hace con dos slash iniciando la línea, seguido del texto.  Para nuestra segunda regla, debemos indicar un punto y coma siempre cuando se finaliza cada declaración, ya que si no lo hacemos será muy difícil de leer para cualquier persona, así que ese punto y coma lo que hace es separar el texto, para que sea un texto mucho más claro para todos.  Para nuestra tercera regla, siempre debemos dejar espacios en blanco, al igual que con nuestra segunda regla, los espacios en blanco dentro del texto nos ayudan a comprender con más claridad lo que estamos leyendo, así que te invito a que tengas esos espacios en tu Javascript.  Una cuarta regla, sería, las minúsculas y las mayúsculas. Que quiero decir con esto cuando hayas escrito una palabra la primera en mayúscula y el resto en minúscula, asegúrate que si debes escribirla más veces durante el recorrido de la creación del archivo en el editor, todas esas veces debe estar escrita de la misma manera; por ejemplo: Hoy es, y luego más abajo escribo hoy es, inmediatamente me arrojará un error ya que las dos palabras no me coinciden, si notas acá la h está en mayúscula y aquí está en minúscula, así que debes escribirlo tanto correctamente, como de la misma forma todas las veces que lo escribas.  Para nuestra quinta y última regla, debemos escribir todo lo que son funciones, métodos y variables en minúscula. Los nombres que hacen referencia a la clase de objetos bene estar con una letra en mayúscula, y las constantes deben estar en mayúsculas.  Estas son las 5 reglas que te recomiendo apliques en tus proyectos Javascript con el fin de que todo salga perfecto y no encontremos errores a la hora de ejecutar nuestros códigos. |

|  |  |
| --- | --- |
| **CREAR UN RELOJ ANALÓGICO DESGLOSE DEL PROYECTO** | Practiquemos y tomemos nuevas demostraciones para que más adelante podamos realizar algoritmos complejos, como te has dado cuenta todo tiene un orden, una estructura y una forma que simplemente no se puede, romper, si simplemente no se hace de la manera en que está establecido no va a funcionar.  Hasta el momento hemos abordado datos simples en este proyecto, y debemos poner un poco mas de funcionamiento a nuestros propósitos, que quiero decir con esto es hora de agregar un poco de complejidad.  Para eso crearemos mediante un paso a paso un reloj analógico, si, así como lo oyes un reloj analógico, que me indique el minutero, el segundero, me marque las horas, que actúe por si solo, cambiando hora tras hora, minuto a minuto, segundo a segundo.  Así que acompáñame por esta ardua labor. |
| **USA CSS PARA MOVER LAS MANECILLAS DEL RELOJ**  **(como debe estar el html)** | Para que nuestro reloj funcione mis tres lenguajes de programación deben tener un contenido y ese contenido te lo explicaré a continuación en un paso a paso. Empecemos con el html  **EDITOR (ejercicio4.2.html)**  Como vemos aquí tiene la etiqueta html, la cabeza, nuestra meta etiqueta, el título llamado reloj analógico, este link representa, el enlace con el CSS, que es el documento con el cual trabajaremos este de acá llamado reloj.css, luego tenemos nuestra etiqueta guion con la ruta del javascript que utilizaremos que es este también llamado reloj.js, bien tenemos el archivo html, css y javascript, todos enlazaos en el html para que las cosas nos funcionen, luego el cierre de la cabeza. Aquí la apertura del cuerpo y en el cuerpo, encontramos lo siguiente, un elemento principal (main) con una clase principal (main). El cual posee un div, con una clase caja de reloj (clockbox), está clase la indique aquí por que es precisamente lo que vamos a crear en este momento la caja de nuestro reloj, es decir la estructura que conforma al mismo, por así decirlo, luego un svg, que representa los gráficos en vectores, luego encontramos el xmlns, que es el atributo que especifica, el espacio que hay en el documento para el nombre xml (que es el lenguaje de marcado que define un conjunto de reglas para codificar documentos).  Luego, le damos valores de ancho, alto, que son estos de acá establecidos en 300 y 300, respectivamente, y el valor al cuadro. Bien, estos son los valores que le di para la forma externa de mi reloj,  Luego, la etiqueta g, con un id de face, es decir la cara de nuestro reloj, Luego le decimos con la etiqueta cricle, que nos encierre esos valores, con un atributo cx, que define la coordenada del eje x del punto central, luego, el cy, define la coordenada del eje y del punto central, y el atributo r, nos distingue el radio del círculo que estamos creando. En la etiqueta de camino que la marcamos justo debajo del id, debemos definir la ruta que se utilizará, (path) así que le damos una clase marcas de hora (hour-marks) con esta m del principio le estamos diciendo que se mueva a esta distancia especificada aquí, que estamos indicando, luego la v que está aquí, que nos indica la línea vertical a estas coordenadas, es decir que con todos los valores que le dimos aquí a nuestra cara le dimos a un círculo un movimiento, con líneas horizontales, verticales, para la forma del reloj. Más abajo nuevamente una etiqueta de circulo, pero esta vez, con una clase mid-circle, que es para el circulo pequeño que encontramos justo en el medio del reloj, damos mismos valores en cx y cy, y para el radio lo estableceré en 16.2, cerramos la etiqueta, recuerda, que cada una de estas etiquetas tienen su respectivo cierre al final, identificado con un slash y con un signo mayor que, y al finalizar esta establecido el cierre de la etiqueta g.  Ahora, aquí lo que nos está mostrando más abajo son estructuras similares, una con id de hora, otra con minuto y la última con segundos, la ruta en la hora, le daremos un brazo para la hora, con una d, que es el diámetro, y le decimos que se mueva a 300.5 y que lo haga verticalmente, con esta v de aquí, y en este punto cierro con comillas y mayor que, más abajo al círculo le doy una clase caja de tallas (sizing-box), con respectivas coordenadas y radio tal cual como lo hicimos con la cara  Más abajo, con los minutos, sus respectivas coordenadas y finaliza con los segundos, cerramos la etiqueta de vectores, es decir svg, como está aquí, luego el cierre del div, cierre de main, que habíamos abierto en la parte de arriba, cierre de cuerpo nuestro respectivo enlace con la carpeta de javascript que utilizaremos, y el cierre final de todo el bloque html. Iré a mi consola, en este caso trabajaré con el documento llamado reloj analógico, actualizo. Y aquí tienes el molde del reloj, ya empieza a tomar sentido.  Muy bien está es la estructura que debe poseer html para realizar nuestro reloj analógico. En la próxima lección te explicaré, como está construido el CSS, para nuestros planes.   <path class="hour-marks" d="M300.5 94V61M506 300.5h32M300.5 506v33M94 300.5H60M411.3 107.817.9-13.8M493 190.2113-7.4M492.1 411.4116.5 9.5M411 492.318.9 15.3M189 492.31-9.2 15.9M107.7 411L93 419.5M107.5 189.31-17.1-9.9M188.1 108.21-9-15.6"/> |
| **USA CSS PARA MOVER LAS MANECILLAS DEL RELOJ**  **(como debe estar el css)** | **EDITOR (reloj.css)**  Entremos a ver la parte del CSS de nuestro reloj, y digitaremos el paso a paso, así que sígueme y trabajemos juntos.  Voy a indicar primero un comentario, llamado diseños, el mismo main que indiqué en el html, lo indicaré aquí antecediéndolo de un punto, abro corchete, y justo debajo daré valores para la propiedad principal, justificaré el contenido, al centro, luego alinear elementos, (align-items), y le diré que al medio (middle) el monitor más abajo establecido en flexible, luego el relleno, esta vez lo dejaré en 3em y el alto en 90vh, no olvides el punto coma finalizando punto y coma, y al finalizar los valores del elemento principal daré el cierre de corchete.  Más adelante tenemos que indicar un único valor a la caja de valor y ese es su alto, primero punto, clockbox, luego #clock, alto de 100% y corchete.  Bien, para separar el diseño de los elementos principales de lo que viene a continuación, crearé u comentario que diga estilo del reloj, y ahora si indicaré los valores para el circulo, y los brazos, primero punto, circle, paréntesis, la anchura del trazo (stroke-width), la dejaré en 10, el límite en 11 (stroke-miterlimit), le diré que para llenar lo deje en ninguno, (fill) y el trazo en color negro, bien estos son los valores que necesitamos para cada una de las partes, habiendo terminado el circule, iré con el circulo pequeño, el del centro del reloj y sólo daré el valor de llenar con un color negro también,.  Para las marcas de la hora, el ancho del trazo en 10, el límite del trazo en 11, llenar, igualmente establecido en ninguno, y el trazo en negro, bien cerramos, y repetimos la acción con as hora, minuto y segundo, para el brazo de la hora (.hour-arm), corchete, el ancho del trazo lo dejaré en 19, el límite del tazo en 12, llenar en ninguno, y el trazo en negro igualmente.  Déjame repetir la acción, para minutos y segundos, como el minutero es más delgado el ancho será 11 y el límite 10, todos los dejaré en negro, y el segundero es mucho más delgado lo dejaré en 5, y límite igualmente en 10.  Terminadas nuestras medidas para cada uno de los elementos del reloj, ahora separemos esta parte con un comentario, daré slash, asterisco, escribiré mi comentario, caja para los brazos que es transparente, cierro con asterisco slash el comentario, y le diré a la caja de tallas que me llene con ninguno.  Aquí en el css solo nos hace falta un paso para poder terminar, hacer que nuestro reloj empiece a tener movimiento, y que se empiece a ver como si fuera animado, así que pondré dos comentarios con esta información, uno de ellos dirá realiza la acción de hacer girar los brazos alrededor del punto del centro y al siguiente comentario, diré, que se vea animado y funciones.  Recuerda que estos comentaros son con el fin de darle organización a lo que estamos realizando.  Luego más abajo, pondré #hour, coma #minute, coma, #second, abro corchete, le diré que el origen de la transformación (transform-origin) será de 300px y 300px, y la transición, se transforma cada 5s de entrada a salida.  Si visualizamos por un segundo nuestra consola veremos la estructura del reloj, sus manecillas, su cuerpo ya estructura, pero estático, la verdad no creo que nos sirva de ha mucho un reloj que no se mueva, es decir que no marqué la hora, lo primero que haremos para eso es empezar a trabajar en javascript, creo que ya es hora de darle dinamismo, así que acompáñame a la próxima lección y veamos nuestro reloj en acción  /\*diseños\*/  .main {      justify-content: center;      align-items: middle;      display: flex;      padding: 3em;      height: 90vh;    }  .clockbox,  #clock {      height: 100%;  }  /\*estilos del reloj\*/  .circle {      stroke-width: 10;      stroke-miterlimit: 11;      fill: none;      stroke: black;    }  .mid-circle {      fill: black;  }  .hour-marks {      stroke-width: 10;      stroke-miterlimit: 11;      fill: none;      stroke: black;    }  .hour-arm {  stroke-width: 19;  stroke-miterlimit: 12;  fill: none;  stroke: black;  }  .minute-arm {      stroke-width: 11;      stroke-miterlimit: 10;      fill: none;      stroke: black;    }  .second-arm {      stroke-width: 5;      stroke-miterlimit: 10;      fill: none;      stroke: black;    }  /\*caja para los brazos que es transparente \*/  .sizing-box {      fill: none;  }  /\* realiza la acción de hacer girar los brazos alrededor del punto del centro. \*/  /\* que se vea animado y funcione \*/  #hour,  #minute,  #second {      transform-origin: 300px 300px;      transition: transform .5s ease-in-out;  } |
| **USA JAVASCRIPT PARA MOVER LAS MANECILLAS DEL RELOJ** | **EDITOR (reloj.js)**  Lo primero es crear tres constantes, la primera de ellas la llamaré horario, que vendría representando las horas, luego citamos al documento llamando a nuestro selector de consultas, abro paréntesis, y enseguida digito estos valores que indicamos en el CSS, uno por uno en cada una de mis constantes, primero indico la hora, (“#hour”), finalicemos aquí con punto y coma, la segunda constante es el minutero, conservamos la misma estructura del selector de consultas, y los minutos (“minute”), y la última constante que indicaré en mi javascript es la del segundero, y finalizo con (“second”). Estas tres son identificaciones de destino  Luego de la misma forma como creamos tres constantes, crearemos tres variables locales, es decir pondré let aquí, luego una más abajo y otra más aquí, en la primera estableceré una posición para la hora y lo hago con hrposition, e indicaré que se mueva a 22, finalizó con punto y coma, para la siguiente variable, le daré una posición a los minutos de 140 y para la última una posición a los segundos de 270 grados, estos valores son por ahora solo experimentales para comprobar si mi reloj se mueve, así que puedes indicar los valores que desees.  Luego a cada una de mis constantes les daré una transformación de estilo, para hacerlo debo indicar primero el nombre de la constante, comenzaré en orden tal cual como las establecimos aquí al principio, es decir horario, va en primer lugar, luego punto, style, punto, transform, igual a “rotate”, que me está indicando que la contante horario la transforme para que rote, luego abro paréntesis, abro comillas, signo más, indico la variable para el horario en este caso hrposition, más, abro comillas, deg, cierro paréntesis, cierro comillas y punto y coma. Ahora realicemos la misma acción con nuestras siguientes constantes, solo cambiaré así a min, y aquí, y para la última lo hago de la misma forma, la constante segundera, y variable en sec. Guardaré, y actualizo mi consola y enseguida vemos como se empiezan a mover las manecillas del reloj a la posición que le indicamos.  Muy bien, vamos progresando.  function giratoria () {      var date = new Date ();      console.log(date);      let hr = date.getHours();      let min = date.getMinutes();      let sec = date.getSeconds();      console.log("Hour: " + hr + "Minutes: " + min + "second:" + sec);      let hrPosition =(hr\*360/12)+(min\*(360/60)/12);      let minPosition =(min\*360/60)+(sec\*(360/60)/60);      let secPosition =sec\*360/60;    HORARIO.style.transform ="rotate("+hrPosition + "deg)";  MINUTERO.style.transform ="rotate("+minPosition + "deg)";  SEGUNDERO.style.transform ="rotate("+secPosition + "deg)";    var intervalo = setInterval(giratoria, 1000); |

**GIT**

|  |  |
| --- | --- |
| **CAPITULO / LECCIONES** | **PRESENTACIÓN** |
| **INTRODUCCIÓN** | **TELEPRONTER**  Hola, soy Tatiana Vega, me alegra saber que estas nuevamente acompañándome en un curso y si es la primera vez que visitas mi contenido, de la misma manera me alegra saber que estás aquí para aprender.  **PRESENTACIÓN**  ¿Sabes? Es realmente emocionante cuando nos hablan de facilitar las cosas, y para eso estoy aquí con este curso básico de GIT, GIT es un software de control de versiones, que nos hace el camino más amigable y también lo simplifica.  En este curso veremos los conceptos, las técnicas fundamentales, como poder escribir un código mucho mejor, y a comprender su funcionalidad.  **TELEPRONTER**  Como tal GIT es un sistema que tiene el propósito de tener un registro de los archivos y controlar las versiones con funcionalidad plena.  **PRESENTACIÓN**  GIT por su parte contribuye en los proyectos de código abierto y nos ayuda a desarrollar funciones.  En la actualidad GIT es el sistema de versiones más utilizado en el mundo, gran cantidad de proyectos de software en todo el mundo dependen de GIT.  Así que no nos extendamos más, te invito para que me acompañes a emprender en el gestionamiento de proyectos a través de este curso básico de GIT. |

|  |  |
| --- | --- |
| **CAPITULO / LECCIONES** | **PRESENTACIÓN** |
| **INTRODUCCIÓN FRONT – END** | **TELEPRONTER**  Accesibilidad y rendimiento, técnicas que alcanzarás con mi curso de formación básica de front-end.  Mi nombre es Tatiana vega, encargada de darle vida al material que verás a continuación, en Display For Play, el cual lo transmitiré a través de 11 lecciones, que te serán de utilidad en la programación.  El front-end se basa en transformar los datos en una interfaz gráfica, con el fin de que todos los usuarios podamos obtener información digital a través de los lenguajes de programación, HTML, CSS, Javascript, y SQL.  Adicionalmente, te daré las bases para que empieces con frameworks y bibliotecas de apoyo para la construcción de todo sitio web, como lo son React.js y Angular.  Existe una variedad de plataformas y herramientas que sirven de apoyo y giran en torno a la programación, y mi idea es que cuando termines con este curso sepas cuales utilizar y con qué propósito.  Así que en complemento veremos el entorno multiplataforma basado en Javascript, es decir Node.js, al igual que una interfaz de programación de aplicaciones llamada Learning rest Apis.  No obstante, apoyaré todo este material con un Software de control de versiones, comúnmente conocido como, GIT.  Y cerrando el combo perfecto para este curso está la plataforma Amazon Web Services (AWS), encargada del almacenamiento en la nube de una forma avanzada y globalizada.  Este es el curso perfecto y más completo que encontrarás, te aseguro que al finalizar estarás completamente listo para crear y mejorar cualquier proyecto.  “Así que”, qué esperas para acompañarme y conocer este grandioso mundo de la programación.  **(Ejemplo)**  <https://www.youtube.com/watch?v=1qSf4b49n38>  **Animaciones – cambios de cámara – musicalización suave – colorización, etc.** |