|  |  |
| --- | --- |
| Etiquetas (TAGs): ejemplo <br>  <br> = salto de linea | mi primer programa HTML!!  <br>  <br>  será esto realmente un programama? |
| <h1> = destacar el titulo (destaca lo que este en esa etiqueta)  el <h1> final va con “/” para señalar que es el fin:  </h1> | <h1>mi primer programa HTML!!</h1>  <br>  <br>  será esto realmente un programama? |
| Para crear links usamos la etiqueta <a> (ancora). Esa etiqueta funciona de la siguiente manera:   La etiqueta posee como contenido la página de Alura Latam, y el contenido puede ser encontrado haciendo clic para que el navegador cargue la página informada en el atributo href de la etiqueta. | Visite <a href="http://www.aluracursos.com"> Alura Latam</a>  href="http://www.aluracursos.com" |
| Para usar lenguaje de JS en un archivo de HTML debes de usar la etiqueda de <script> </script> para que las instruciones de JS si se ejecuten | <script>  alert("Esto sí es lenguaje de programacíón");  </script> |
| Siempre este sera el orden normal para empesar un archivo HTML: | <meta charset="UTF-8">  <script>  </script> |
| <meta charset="UTF-8">  Sirve para definir la codificacion que usamos, asi las letras con tile por ejemplo se veran normal en Exporer navegador | Convenciones: (reglas html)   -las etiquetas siempre son minúsculas (igual en las instrucciones de JS: obligatorio) |
| Para usar HTML dentro de la etiqueta de JS se puede usar:   document.write | <script>      document.write("La edad de Christian es ...")  </script> |
| Para comentar algo usar: // en JS Para comentar en HTML: <!-- esto es un comentario -->  <hr> = separar visualmente distintas secciones de contenido, separar visualmente los dos párrafos de texto.  <big> = aumentar el tamaño de una fuente o texto  <p>Este es un texto normal.</p>  <p><big>Este es un texto con un tamaño de fuente mayor.</big></p>  texto que se mostrará con una fuente más grande |  |
| Input =  <input/>  Da un elemento donde el usruario puede ingresar cualquier valor | Button = boton donde el usuario verificara si es correcto lo que ingreso  <button>  Verificar si aceró con el secreto  </button> |
| querySelector = extrae un elemento, en este caso el boton, para usarlo: - querySelector, uno de los posibles valores que podemos pasarle son los nombres de las etiquetas      var button = document.querySelector("Button");      button.onclick = Verificar;  Juego\_secreto.html (es el archivo en donde esta)  button.onclick = activa el boton, para que cuando le hagas click haga su funcion  button.onclick = Verificar; |  |
| .value = borra lo que has escrito en el boton:  input.value = "";  .facus = hace que el maus de teclado se focalice en el input para agregar el dato  input.focus(); |  |
|  |  |
| **HTML :**  <canvas>: Es un elemento de HTML utilizado para crear gráficos y dibujos en una página web mediante programación en JavaScript.  height (altura): Atributo del elemento <canvas> que establece la altura del área de dibujo en píxeles.  width (ancho): Atributo del elemento <canvas> que establece el ancho del área de dibujo en píxeles.  <canvas width="600" height="400"> </canvas>  **JS :**  ***Métodos del objeto CanvasRenderingContext2D***  **pincel.moveTo(x, y)** y **pincel.lineTo(x, y)** son métodos del objeto CanvasRenderingContext2D en JavaScript que se utilizan para dibujar líneas en un lienzo HTML5.  moveTo() mueve el cursor del pincel a la posición especificada por las coordenadas (x, y), sin dibujar ninguna línea. Por lo tanto, se utiliza generalmente para iniciar un nuevo trazo en una ubicación específica del lienzo.  lineTo() dibuja una línea recta desde la última posición del cursor hasta la posición especificada por las coordenadas (x, y), utilizando el estilo de línea actual. Se utiliza en conjunto con moveTo() y otros métodos de dibujo para trazar una serie de líneas rectas para crear una forma o imagen más compleja en el lienzo.  pincel.beginPath() es un método del objeto CanvasRenderingContext2D en JavaScript que se utiliza para comenzar un nuevo trazo en el lienzo HTML5. Este método inicia una nueva ruta vacía al restablecer el camino actual del lienzo y los subcaminos.  Cuando se llama a beginPath(), se restablece el estilo de línea actual del lienzo (ancho de línea, estilo, color, etc.), y cualquier trazo anterior se elimina de la memoria del lienzo. Luego, se puede comenzar a dibujar un nuevo trazo utilizando una combinación de los métodos de dibujo, como moveTo(), lineTo(), arc(), rect(), etc.  **pincel.fillStyle** es un atributo del objeto CanvasRenderingContext2D en JavaScript que define el estilo de relleno que se utiliza para dibujar formas en el lienzo.  Este atributo puede ser un color sólido, un gradiente o un patrón, y se utiliza como argumento para los métodos fill() y fillRect()  **pincel.fill()** es un método del objeto CanvasRenderingContext2D en JavaScript que se utiliza para rellenar la forma cerrada actual con el color o patrón especificado en el contexto de dibujo del lienzo.  pincel.fill() es un método del objeto CanvasRenderingContext2D en JavaScript que se utiliza para rellenar la forma cerrada actual con el color o patrón especificado en el contexto de dibujo del lienzo.  Después de trazar una forma cerrada utilizando los métodos moveTo(), lineTo(), arc(), rect(), etc. y llamar a closePath() para cerrar la ruta, el método fill() se utiliza para colorear el interior de la forma cerrada con el color o patrón especificado.  El método fill() utiliza el valor del atributo fillStyle del contexto de dibujo del lienzo para determinar el color o patrón que se utiliza para rellenar la forma cerrada. Este atributo puede ser un color sólido, un gradiente o un patrón, y se puede establecer utilizando los métodos strokeStyle, createLinearGradient(), createRadialGradient(), createPattern(), entre otros.  Es importante tener en cuenta que el método fill() solo pinta el interior de la forma cerrada, y no pinta el borde de la forma. Si se desea pintar el borde de la forma, se puede utilizar el método stroke() después de llamar a fill().  **El método** strokeStyle define el color, patrón o gradiente utilizado para el trazo de la figura en el lienzo. Si se establece como un color, puede ser cualquier valor de color CSS válido. Por ejemplo, "black" para negro, "#FF0000" para rojo, "rgb(255, 0, 0)" para rojo en notación RGB, etc.  **El método** strokeRect() dibuja un rectángulo sin relleno en el lienzo con el color de trazo definido por el método strokeStyle(). El rectángulo está definido por los argumentos x, y, ancho y alto. En este caso, el rectángulo comienza en la posición (0,0) y tiene un ancho y altura de 50 unidades. | **JS :   pantalla**: Es el objeto del elemento <canvas> que se obtiene mediante la función document.getElementById("pantalla").  La **función** de **<canvas> </canvas>:** pantalla.getContext("2d") devuelve un objeto que representa el contexto de dibujo en 2D del objeto pantalla. A través de este objeto, se pueden realizar operaciones de dibujo en el área del canvas, como dibujar líneas, formas, imágenes, etc.  Los gráficos se crean utilizando una API de dibujo en 2D o 3D que proporciona el navegador.  **.fillStyle:** es una propiedad del objeto de contexto de canvas en HTML5 que se utiliza para establecer el color de relleno que se aplicará a las formas dibujadas en el canvas.  Para utilizar esta propiedad, primero se debe obtener el contexto de dibujo del canvas utilizando el método getContext() y luego se puede establecer el valor de la propiedad fillStyle con una cadena de texto que represente un color en formato hexadecimal, RGB o RGBA. Por ejemplo, el siguiente código establece el color de relleno en rojo:      var pantalla = document.querySelector("canvas");      var pincel = pantalla.getContext("2d");      pincel.fillStyle = "green" // propiedad      pincel.fillRect(0, 0, 200, 400) // funcion  "**.fillRect()**" es un método del objeto de contexto del canvas en HTML5 que se utiliza para dibujar un rectángulo relleno en el canvas.  Para utilizar este método, primero se debe obtener el contexto de dibujo del canvas utilizando el método getContext(), y luego se puede llamar al método fillRect() y pasarle cuatro argumentos que representen las coordenadas x e y del punto superior izquierdo del rectángulo, así como su ancho y altura. Por ejemplo, el siguiente código dibuja un rectángulo relleno de color rojo en el canvas:      var pantalla = document.querySelector("canvas");      var pincel = pantalla.getContext("2d");      pincel.fillStyle = "green" // propiedad      pincel.fillRect(0, 0, 200, 400) // funcion  "**.beginPath()**" es un método del objeto de contexto del canvas en HTML5 que se utiliza para comenzar una nueva ruta de dibujo.  Cuando se utiliza el canvas para dibujar formas o líneas, es común definir una serie de puntos o vectores para indicar el camino que debe seguir el dibujo. Para indicar estos puntos, se utiliza una ruta de dibujo, que se define mediante una serie de comandos como "moveTo", "lineTo", "arc", entre otros.  El método "beginPath()" es el primer paso para definir una nueva ruta de dibujo en el canvas. Una vez que se llama a este método, se pueden agregar comandos para especificar la ruta de dibujo utilizando los métodos correspondientes, como "moveTo()" o "lineTo()".      var canvas = document.getElementById("myCanvas");      var contexto = canvas.getContext("2d");      // Comienza una nueva ruta de dibujo      contexto.beginPath();      // Agrega un punto inicial en las coordenadas (10, 10)      contexto.moveTo(10, 10);      // Agrega un punto final en las coordenadas (50, 50)      contexto.lineTo(50, 50);      // Dibuja la línea utilizando el color de trazo actual      contexto.stroke();  **La forma de imprimir una funcion** en JS es poener el nombre de la funcion y al final usar: ();  ();  **Pi:**  3.14 (pi) = Match.PI  La función **pincel.clearRect(x, y, ancho, alto)** se utiliza en el contexto del objeto pincel en HTML5 Canvas para borrar una región rectangular dentro del lienzo (canvas). Permite eliminar el contenido existente en el área especificada y dejarla en blanco, lo que crea un efecto de borrado en el lienzo.  Los parámetros utilizados en pincel.clearRect() son los siguientes:  x: La coordenada horizontal (en píxeles) del punto de inicio del rectángulo a borrar.  y: La coordenada vertical (en píxeles) del punto de inicio del rectángulo a borrar.  ancho: El ancho del rectángulo a borrar en píxeles.  alto: La altura del rectángulo a borrar en píxeles.  Al llamar a pincel.clearRect(), se especifica el área rectangular a borrar dentro del lienzo. Todo contenido previo dentro de ese rectángulo se eliminará y el área se mostrará como un espacio en blanco transparente.  Es importante destacar que pincel.clearRect() solo afecta al lienzo y no modifica propiedades o estilos adicionales del objeto pincel. Por lo tanto, si se desea borrar y establecer un color de fondo específico en el lienzo, es necesario utilizar métodos adicionales, como pincel.fillStyle y pincel.fillRect().  **setInterval** es una función en JavaScript que permite ejecutar repetidamente un fragmento de código o una función con un intervalo de tiempo predefinido.  La sintaxis básica de setInterval es la siguiente:  setInterval(funcion, tiempo);  Por ejemplo, el siguiente código ejecutará la función miFuncion cada 1 segundo:      function miFuncion() {              console.log("Ejecutando miFuncion");          }          setInterval(miFuncion, 1000);  Es importante tener en cuenta que setInterval seguirá ejecutándose hasta que se detenga explícitamente con la función **clearInterval.** Para detener la ejecución periódica, se debe almacenar el identificador de intervalo devuelto por setInterval y usarlo con clearInterval de la siguiente manera:      var intervalo = setInterval(funcion, tiempo);      // Para detener la ejecución periódica      clearInterval(intervalo); |
|  |  |
|  |  |
|  |  |
|  |  |