|  |  |
| --- | --- |
| 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 :   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(); |
|  |  |
|  |  |
|  |  |
|  |  |