| **MAPA MENTAL** |
| --- |
| * Case sensitive * Punto y coma tras las instrucciones * Comentarios * Variables: declaración y ámbitos |

| **CASE SENSITIVE** |
| --- |
| Distingue entre mayus y minus.  let mensaje = “hola”;  let Mensaje = “caracola”;  Mensaje y mensaje son distintas variables, apuntan a diferentes direcciones de memoria. |

| **PUNTO Y COMA TRAS LAS INSTRUCCIONES** |
| --- |
| Solo cuando las instrucciones haya más de una instrucción en una misma línea.  Sino no hace falta, aunque es una buena práctica. |

| **COMENTARIOS** |
| --- |
| // un comentario de una línea  /\* este es un comentario  \* más largo, de varias líneas  \*/  /\* Sin embargo, no puedes /\* anidar comentarios \*/ SyntaxError \*/ |

| **VARIABLES** |
| --- |
| ¿Dudas sobre qué es una variable? Una variable es un contenedor de información. Para poderla usar, le daremos un nombre que explique para qué la vamos a utilizar. Una variable puede contener cualquier tipo de dato, como un número o un texto, y se le asigna un valor de la siguiente forma:  Imagen que contiene interior, tabla, sostener, pájaro  Descripción generada automáticamente  Ahora vamos a ver algunos ejemplos de declaración de variables. Te animo a probar el valor de la variable llevando este código a JavaScript Tutor:   * let numero = 3; //le asignamos el valor numérico 3 a la variable numero. * let texto = "Adios"; //le asignamos el valor de texto "Adios" a la variable texto. * let encontrado = false; //le asignamos el valor false a la variable que se llama encontrado |

| **DECLARACIÓN VAR vs LET // ÁMBITOS VARIABLES LOCAL vs GLOBAL** |
| --- |
| En la web de Mozilla Developer podéis ver varios ejemplos.  Diferencia entre let y var.  La principal diferencia está en que cuando declaramos una variable con let dentro de un bloque, aunque la variable ya estuviese declarada antes, fuera del bloque, la nueva variable no reescribe el valor de la declarada fuera. Estos dos ejemplos deberían aclarar las diferencias entre let y var:  let i = 4; var j = 5; {   let i = 6;    j = 7; } console.log(i); console.log(j);  En este caso, al salir del bloque, i conserva su valor inicial (4) mientras que el valor de j se ha modificado dentro del bloque.  i = 4; var j = 5; {    i = 6;   j = 7; } console.log(i); console.log(j);  Nota: Podemos usar una variable sin haberla declarado antes. Sin embargo, esa práctica no es recomendable. Por defecto usaremos siempre let para declarar las variables |

| **CONSTANTES** |
| --- |
| const PI = 3.14;  Las variables definidas como constantes no pueden cambiar su valor.  Excepciones: arreglos y objetos. |

| **ESTRUCTURA DE FICHEROS A CREAR** |
| --- |
|  |

| **Variables.js (más abajo está resuelto)** |
| --- |
| //Inicializa una variable con un valor alfanumérico y muéstralo por consola  //Reasigna un valor a esa variable y muéstralo por consola  //Javascript es un lenguaje de tipo dinámico, esto es, no necesita especificar el tipo, y una variable  //que fue asignado con un valor alfanumérico, puede ser reasignada con un valor numérico, ya que javascript  //hace la reconversión de tipos en tiempo de ejecución. Así que reasigna un valor numérico a esa variable  //y muestrálo por consola  //Create una variable y asignale un valor booleano  //y muestrálo por consola  //Declara tres variables a la vez usando un solo var  //y muestrálo por consola  //Recuerda que a la hora de declarar variables, no pueden empezar por un número, pero si por ejemplo con un \_  //también es recomendable que las clases empiezen por mayúsculas y los nombres compuestos pues la segunda por mayúscula o con  //un \_  *var* ClaseProducto;  *var* nombre\_producto;  *var* nombreProducto;  **Resultado consola:** |

| **Variables.js (resuelto)** |
| --- |
| //Inicializa una variable con un valor alfanumérico y muéstralo por consola  *var* producto = "Altavoces";  console.log(producto);  //Reasigna un valor a esa variable y muéstralo por consola  producto = "televisión";  console.log(producto);  //Javascript es un lenguaje de tipo dinámico, esto es, no necesita especificar el tipo, y una variable  //que fue asignado con un valor alfanumérico, puede ser reasignada con un valor numérico, ya que javascript  //hace la reconversión de tipos en tiempo de ejecución. Así que reasigna un valor numérico a esa variable  //y muestrálo por consola  producto = 100;  console.log(producto);  //Create una variable y asignale un valor booleano  *var* esprimo;  esprimo = true;  console.log(esprimo);  //Declara tres variables a la vez usando un solo var  *var* poblacion\_1 = "málaga"      poblacion\_2 = "marbella"      poblacion\_3 = "nerja";  console.log(poblacion\_1);  console.log(poblacion\_2);  console.log(poblacion\_3);  //Recuerda que a la hora de declarar variables, no pueden empezar por un número, pero si por ejemplo con un \_  //también es recomendable que las clases empiezen por mayúsculas y los nombres compuestos pues la segunda por mayúscula o con  //un \_  *var* ClaseProducto;  *var* nombre\_producto;  *var* nombreProducto;  //recuerda que las constantes no se pueden reasignar const  //recuerda que las locales es preferible usar let |