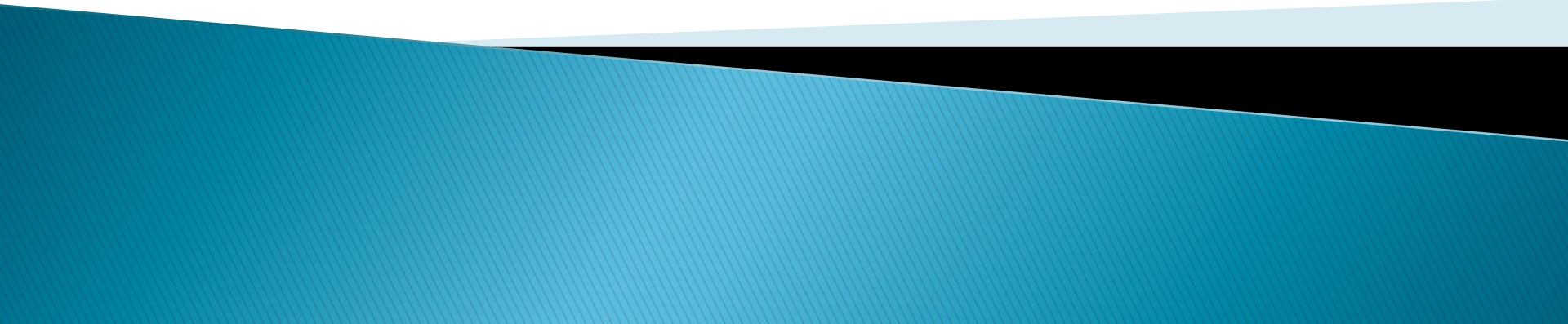
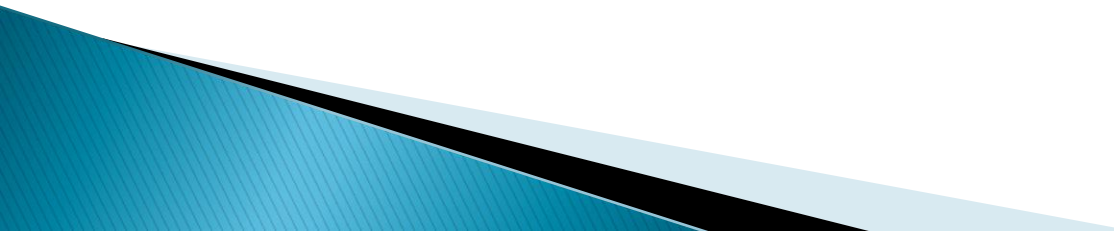


TEMA 3c

VARIABLES

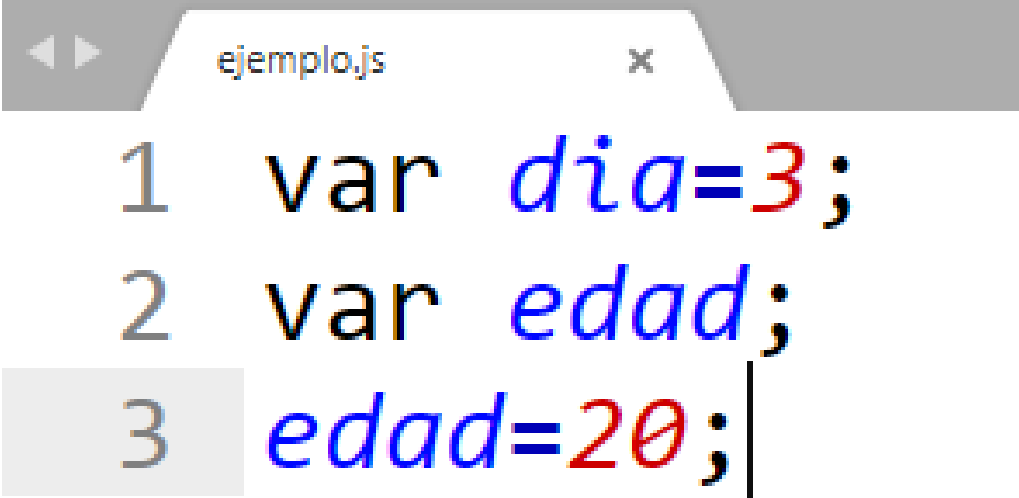


1. VARIABLES

- ▶ Las variables son nombres que ponemos a los lugares donde almacenamos la información. En JavaScript, deben comenzar por una letra o un subrayado (_), pudiendo haber además dígitos entre los demás caracteres.
 - ▶ Una variable no puede tener el mismo nombre de una palabra clave del lenguaje.
 - ▶ No debemos usar signos raros en la definición de las variables (ñ, @, #, acentos, etc.)
- 

1. VARIABLES

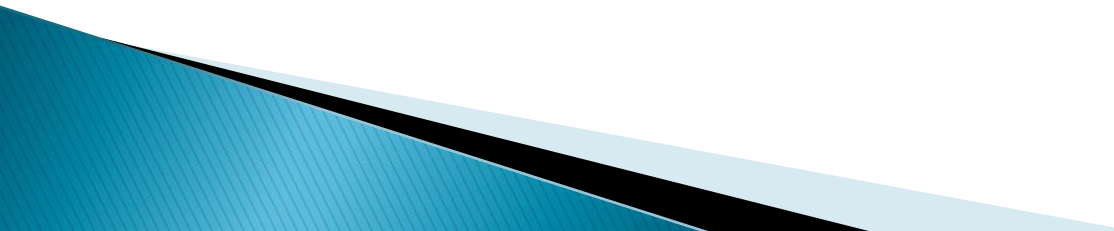
- ▶ Una variable se puede declarar en JavaScript, de dos formas:
 - ▶ Forma Explícita: `var nombre Variable;`
 - ▶ Forma Implícita: `var nombre Variable= valor;`



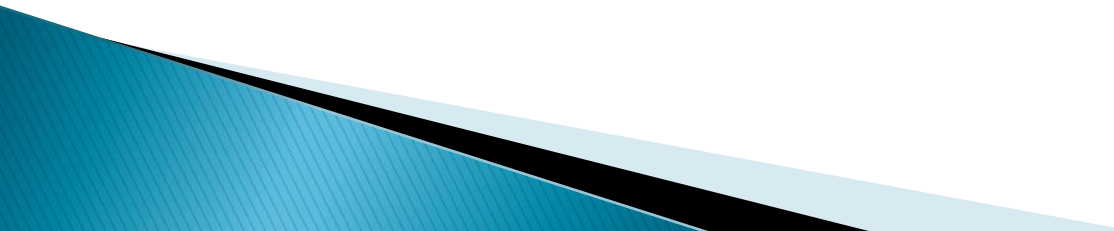
A screenshot of a code editor window titled 'ejemplo.js'. The editor contains three lines of JavaScript code: line 1: `var dia=3;`, line 2: `var edad;`, and line 3: `edad=20;`. The code is color-coded: 'var' is black, variable names are blue, and values are red. A vertical cursor is at the end of the third line.

```
1  var  dia=3;
2  var  edad;
3  edad=20;
```

1. VARIABLES

- ▶ JavaScript es un lenguaje de programación "Case Sensitive", esto es: no es lo mismo las mayúsculas que las minúsculas.
 - ▶ Es decir, para el JavaScript: pepe es distinto de Pepe y distinto de pEpe.
- 

1. VARIABLES

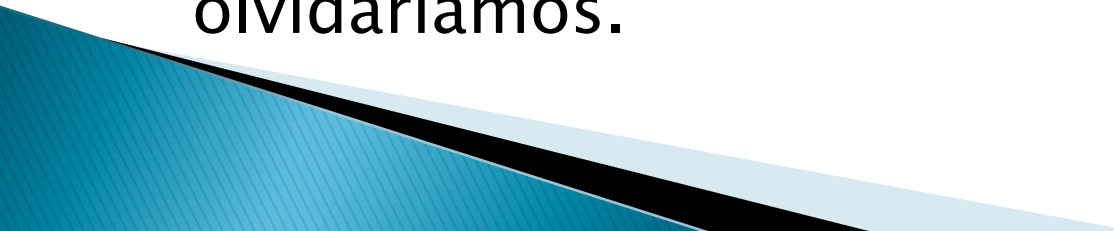
- ▶ LET te permite declarar variables limitando su alcance (scope) al bloque, declaración, o expresión donde se está usando.
 - ▶ Lo anterior diferencia let de la palabra reservada var , que define una variable global o local en una función sin importar el ámbito del bloque.
- 

1. VARIABLES

ejemplo13.js

```
1 |let a=3; //var a=3;
2 |{
3 |    let juan="hola";
4 |    var mesa="hola";
5 |}
6 |// alert(juan); --> no se podria usar
7 |alert(mesa);
```

1. VARIABLES

- ▶ Debemos elegir nombres de variables representativos. Los nombres dia, mes, anio son lo suficientemente claros para darnos una idea acabada sobre su contenido, una mala elección de nombres hubiera sido llamarlas a,b y c.
 - ▶ Podemos darle otros buenos nombres. Otros no son tan representativos, por ejemplo d, m, a.
 - ▶ Posiblemente cuando estemos resolviendo un problema dicho nombre nos recuerde que almacenamos el dia, pero pasado un tiempo lo olvidaríamos.
- 

1. VARIABLES

- ▶ Se pueden declarar varias variables en una misma línea:
 - ▶ `var dia, mes, anio;`
- ▶ A una variable se la puede definir e inmediatamente inicializarla con un valor:
 - ▶ `var edad=20;`
- ▶ O en su defecto en dos pasos:
 - ▶ `var edad;`
 - ▶ `edad=20;`

1 A. MODO ESTRICTO

```
ejemplo13.js
1  'use strict'
2  let a=3; //var a=3;
3  {
4      let juan="hola";
5      var mesa="hola";
6  }
7  // alert(juan); --> no se podria usar
8  alert(mesa);|
```


2. TIPOS DE DATOS

- ▶ JavaScript distingue entre tres tipos de datos y todas las informaciones que se puedan guardar en variables van a estar encajadas en uno de estos tipos de datos.
- ▶ Podemos conocer de que tipo es una variable mediante la función **typeof(variable)**

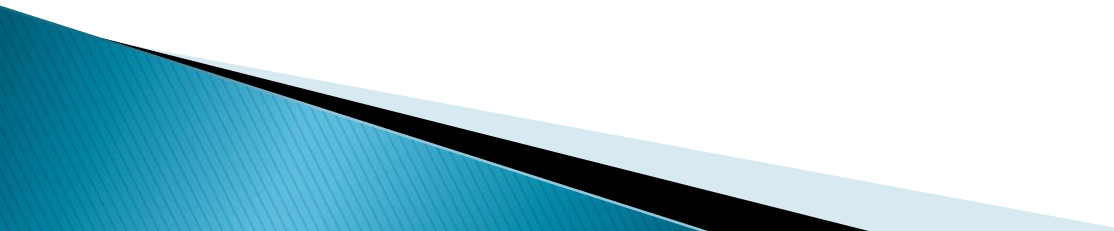
```
var edad=30;  
document.write(typeof(edad));
```

- ▶ Veamos detenidamente cuáles son estos tres tipos de datos.

2A . Tipo de datos numérico

- ▶ En este lenguaje sólo existe un tipo de datos numérico, al contrario que ocurre en la mayoría de los lenguajes más conocidos. Todos los números son por tanto del tipo numérico, independientemente de la precisión que tengan o si son números reales o enteros.
 - ▶ Los números enteros son números que no tienen coma, como 3 o 339. Los números reales son números fraccionarios, como 2.69 o 0.25, que también se pueden escribir en notación científica, por ejemplo 2.482e12.
- 

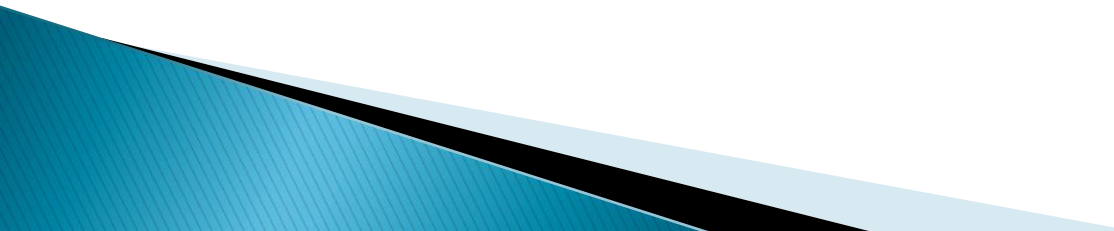
2B. Tipo booleano

- ▶ El tipo de datos booleano sirve para guardar los valores verdadero(true) o un falso(false).
 - ▶ Se utiliza para realizar operaciones lógicas, generalmente para realizar acciones si el contenido de una variable es verdadero o falso.
 - ▶ Los dos valores que pueden tener las variables booleanas son true o false.
- 

2B. Tipo booleano

```
var miBoleana = true;  
var miBoleana = false;
```

2c. Tipo cadena de caracteres (texto)

- ▶ El último tipo de datos es el que sirve para guardar un texto.
 - ▶ JavaScript sólo tiene un tipo de datos para guardar texto y en el se pueden introducir cualquier número de caracteres.
 - ▶ Un texto puede estar compuesto de números, letras y cualquier otro tipo de caracteres y signos. Los **textos** se escriben entre **comillas**, **dobles** o **simples**.
- 

2c. Tipo cadena de caracteres (texto)

```
var miTexto = "Pepe se va a pescar"  
var miTexto = '23%%$ Letras & *--*'
```

- Todo lo que se coloca entre comillas, como en los ejemplos anteriores es tratado como una cadena de caracteres independientemente de lo que coloquemos en el interior de las comillas.

2c. Tipo cadena de caracteres (texto)

► Secuencias de escape

Secuencia de escape	Resultado
\\	Barra invertida
\'	Comilla simple
\"	Comilla doble
\n	Salto de línea
\t	Tabulador
\f	Salto de pagina
\r	Retorno de carro
\b	retroceso