

## 2.3. Variables

### Identificadores

A lo largo del programa va a ser necesario **nombrar elementos y asignar nombres a diferentes estructuras de datos** que permitan utilizarlas de acuerdo con las necesidades.

Por ejemplo, si se desea crear un espacio reservado en memoria que almacene la fecha del día en curso, se puede crear el identificador **fecha\_actual**.

Para definir un identificador hay que asegurarse de no violar ninguna de las siguientes reglas:

- Debe empezar obligatoriamente por una letra, guion bajo U o el símbolo del dólar (\$).
- Pueden seguirle más letras, números o guiones bajos.
- **Distingue entre mayúsculas y minúsculas.**
- Puede usar todas las letras que están definidas en UNICODE

# Comentarios

// → para iniciar el comentario. Se considerará que forma parte del comentario todo lo que aparezca a continuación y hasta final de la línea.



```
// Controla el primer bucle (número de iteraciones)
```

```
var contador = 0;
```

```
// Indica si el recurso está libre (true) u ocupado (false)
```

```
var recurso = true;
```

# Comentarios

`/* ... */` → para comentarios de varias líneas.

`/*`

Almacena la cantidad de elementos actuales de la estructura.  
Solo se incluyen los positivos.  
Debe coincidir con los dispositivos válidos.

`*/`

`let numElementos = 0;`

# Variables

Una **variable** es una zona de memoria a la que se asigna un nombre y en la que se guarda un valor.

Puede hacerse referencia a esa zona de memoria mediante un **identificador** que el programador ha definido, recuperar su valor o modificarlo .

JS es un **lenguaje dinámicamente tipado**: no se indica de qué tipo es una variable al declararla e incluso puede cambiar su tipo a lo largo de la ejecución del programa.

```
let miVariable;           // declaro miVariable y como no se asigno un valor valdrá undefined
miVariable='Hola';        // ahora su valor es 'Hola', por tanto contiene una cadena de texto
miVariable=34;            // pero ahora contiene un número
miVariable=[3, 45, 2];    // y ahora un array
miVariable=undefined;     // para volver a valer el valor especial undefined
```

JS es **débilmente tipado**: permite operaciones entre tipos de datos diferentes:

```
miVariable='23';
console.log(miVariable * 2); // mostrará 46 ya que convierte la cadena '23' a número
```

# Declaración de variables

Las variables se declaran con **let** o con **var**.

Con **let** el ámbito de la variable es sólo el bloque en que se declara mientras que con **var** su ámbito es global (o global a la función en que se declara).

Cualquier variable que no se declara dentro de una función (o si se usa sin declarar) es global.

```
{  
  var miVariable = "Soy visible fuera del bloque.";  
  let otraVariable = "No soy visible fuera del bloque.";  
}  
console.log(miVariable);  
console.log(otraVariable);
```





```
var miVariable = "He sido declarada fuera de todo bloque";  
function unaFuncion() {  
    var miVariable = "He sido declarada dentro de una función";  
    return 0;  
}  
console.log(miVariable);
```

