

## VARIABLES

Una variable es una zona de memoria de nuestro equipo que utilizará nuestro programa para almacenar valores. JavaScript, al igual que otros muchos lenguajes de programación, admite el uso de distintos tipos de variables: números, cadenas o strings, lógicas o booleanos, arrays y objetos.

Para poder utilizar una variable en nuestro script tendremos que declararla haciendo uso de la palabra reservada *var* y asignándole un nombre que la identifique. He aquí un sencillo script que realiza una prueba del uso de variables:

### *javascript8.html*

```
<!DOCTYPE html>
<html>
<head>
  <title>Javascript 8</title>
  <meta charset="UTF-8">
</head>
<body>

<h1>Uso de variables</h1>

<script>
  var a=3;
  var b=5.6;
  var suma;
  suma = a+b;
  alert(suma);
</script>
</body>
</html>
```

Observe que:

- Las variables se declaran con la palabra reservada *var*
- Podemos asignarle un valor predeterminado mediante el operador '='
- Al igual que otros lenguajes podemos asignar a una variable el valor resultado de una operación cualquiera utilizando de nuevo el operador '='.
- Invocando *alert(suma)* mostraremos en un pop-up el contenido de la variable suma.

## UN SENCILLO EJEMPLO DE USO CON FORMULARIOS

### *javascript9.html*

```
<!DOCTYPE html>
<html>
<head>
    <title>Javascript 9</title>
    <meta charset="UTF-8">
</head>
<body>
<h1>Escribe tu nombre</h1>
<input id="entradaTexto" type="textbox">
<button onclick="saludo()">Login</button>
<script>
    function saludo() {
        var cadena;
        cadena = document.getElementById("entradaTexto").value;
        alert('Bienvenido '+cadena);
    }
</script>
</body>
</html>
```

Observe que:

- Podemos asignar a la variable cadena el valor que el usuario ha introducido en el campo de texto.
- Las cadenas se unen(concatenan) usando el operador '+'

### **Ejercicio:**

Realice un script usando JavaScript que dado un usuario y una contraseña presente en pantalla el mensaje *"Bienvenido usuario X con la contraseña Y"* siendo X e Y los valores de las cadenas introducidas en sendos campos de formulario.

### **Ejercicio:**

Realice un conversor de euros(€) a dólares americanos(\$) y de dólares a euros haciendo uso de formularios, cajas de textos, botones y variables. Considere, por ejemplo, el cambio de 1 \$ = 0,94 €.

### **Ejercicio:**

Realice un sencillo conversor de grados Celsius a grados Fahrenheit usando JavaScript. Tenga en cuenta la siguiente regla de conversión:

$$T(^{\circ}\text{F}) = T(^{\circ}\text{C}) \times 9/5 + 32$$

## ALCANCE DE UNA VARIABLE

El ámbito de uso de una variable puede ser global, esto es, puede ser utilizada en todo el documento HTML o local, solamente la podremos usar dentro de la función donde la declaramos.

### *javascript10.html*

```
<!DOCTYPE html>
<html>
<head>
  <title>Javascript 10</title>
  <meta charset="UTF-8">
</head>

<body>

<p>Prueba de variable global</p>

<button onclick="suma()">Incrementa</button>
<button onclick="resta()">Decrementa</button>
<p id="demo"></p>
<script>
var numero=1; // Variable declarada de manera global

document.getElementById("demo").innerHTML = numero;
function resta() {
  numero -=1;
  document.getElementById("demo").innerHTML = numero;
}
function suma() {
  numero +=1;
  document.getElementById("demo").innerHTML = numero;
}
</script>

</body>
</html>
```

Observe que:

- Declarando la variable fuera del ámbito de una función hacemos que pueda ser modificada en cualquier parte de nuestro código.

### *javascript11.html*

```
<!DOCTYPE html>
<html>
<head>
  <title>Javascript 11</title>
  <meta charset="UTF-8">
</head>
<body>
<p>Prueba de variable local</p>
<button onclick="pruebaVariableLocal()">Prueba</button>
<p id="dentro"></p>

<script>
alert("Fuera de la función la variable a no existe y produce un error en
esta línea" + a); // Mira el inspector de código, consola
function pruebaVariableLocal() {
  var a;
  a = 44;
  document.getElementById("dentro").innerHTML = "Dentro de la función la
variable a vale " + a;
}
</script>

</body>
</html>
```

Observe que:

- El código no muestra convenientemente el valor de la variable *a* al ser invocado en el *alert* por estar fuera del ámbito de la función y aún no está definida(véase *inspector de código -> consola*)

## ARRAYS

Un array en JavaScript es un tipo de variable especial que permite almacenar varios valores al mismo tiempo.

Podemos declararla de diferentes formas:

```
// Declaración mediante corchetes
var tweets = ["Buenos días, que tal? #twitteron", "Trabajar y trabajar sin
parar...", "No por mucho madrugar amanece más temprano"];

// Declaración con palabra reservada Array
var colores = new Array("rojo", "verde", "amarillo");
```

Podemos acceder a los distintos elementos de un array a través de un índice entero que comienza por 0 y termina en la *longitud-1* del tamaño del array.

### **javascript12.html**

```
<!DOCTYPE html>
<html>
<head>
    <title>Javascript 12</title>
    <meta charset="UTF-8">
</head>
<body>
<h1>Arrays</h1>
<script>
    var palabras=["casa", "perro", "arbol", "coche"];
    document.write(palabras[0]);
    document.write(palabras[1]);
    document.write(palabras[2]);
    document.write(palabras[3]);
</script>
</body>
</html>
```

Algunos métodos útiles para arrays nos permiten conocer su longitud, añadir elementos, sustraerlos, etc.

```
var vector = [5, -6, 23, 60];
var longitudArray = vector.length;
vector.push(100); // Añade un elemento al array
vector.pop(); // Sustraer el último elemento al array
```

## javascript13.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Javascript 13</title>
  <meta charset="UTF-8">
</head>

<body>

<p>Pulsa el botón para sustraer elemento.</p>
<button onclick="sustraer()">Sustraer</button>
<p id="demo"></p>
<script>
var frutas = ["Platano", "Naranja", "Manzana", "Mango"];
document.getElementById("demo").innerHTML = frutas;

function sustraer() {
  frutas.pop();
  document.getElementById("demo").innerHTML = frutas;
}
</script>

</body>
</html>
```

Podemos recorrer un array utilizando la estructura repetitiva *for* y realizar lo que deseemos con todos o parte de sus elementos.

```
var frutas, texto, longitud, i;

frutas = ["Platano", "Naranja", "Manzana", "Mango "];
longitud = frutas.length;
texto = "<ul>";
for (i = 0; i < longitud; i++) {
  texto += "<li>" + frutas[i] + "</li>";
}
```

## MÉTODOS DE ARRAYS

Podemos realizar una gran cantidad de operaciones utilizando arrays. Algunas de las más comunes las podemos visualizar en esta tabla:

Método	Acción
<b>toString()</b>	Convierte el array en una cadena separando todos sus elementos por ','
<b>join(str)</b>	Une todos los elementos dentro de una cadena separándolos por la cadena str.
<b>pop()</b>	Elimina el último elemento del array
<b>push(x)</b>	Añade por el final el elemento x al array
<b>shift()</b>	Elimina el primer elemento del array
<b>unshift(x)</b>	Añade por el principio el elemento x al array
<b>delete array[i]</b>	Elimina el elemento i-ésimo del array
<b>splice(a,b,e1,e2)</b>	Añade en la posición <i>a</i> los elementos e1 y e2 eliminando <i>b</i> elementos
<b>splice(0,1)</b>	Elimina el primer elemento del array
<b>concat(array2)</b>	Concatena al array que lo invoca el array2
<b>slice(i)</b>	Crea un nuevo array partiendo el original a partir de la posición i-ésima
<b>indexOf(e)</b>	Devuelve el índice que ocupa el elemento e en el array
<b>sort()</b>	Ordena el array en orden alfabético
<b>reverse()</b>	Ordena el array en orden inverso


### Ejercicio:

Realice un script que, mediante botones y cajas de texto, permita añadir/eliminar elementos a un array, ordenarlo de manera creciente y decreciente así como la posibilidad de buscar un elemento en la estructura de datos indicando en pantalla su posición.

## OBJETOS

Un objeto en JavaScript es un tipo especial de variable que tiene propiedades y métodos.

Ejemplo:

	Propiedades	Métodos
	<code>coche.nombre = "BMW"</code> <code>coche.tipoCombustible = "gasolina"</code> <code>coche.cantidadCombustible = 0</code> <code>coche.modelo = "318"</code> <code>coche.potencia = 135</code> <code>coche.peso = 815</code> <code>coche.color = "azul"</code>	<code>coche.arrancar()</code> <code>coche.frenar()</code> <code>coche.repostar(n)</code> <code>coche.detener()</code>

Todos los objetos comparten las mismas propiedades y métodos, si bien las primeras pueden tener distintos valores en nuestro código entre los distintos objetos del mismo tipo.

La sintaxis a la hora de declarar un objeto es la siguiente:

```
var idObjeto = {  
  // Propiedades  
  propiedad1: valor1,  
  propiedad2: valor2,  
  ...  
  propiedadN: valorN,  
  // Métodos  
  metodo1: function() {  
    ...  
  },  
  metodo2: function() {  
    ...  
  },  
  ...  
  metodoM: function() {  
    ...  
  }  
};
```

Podemos acceder a las propiedades de un objeto:

```
idObjeto.propiedad1
```

o bien:

```
idObjeto["propiedad1"]
```



## javascript14.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Javascript 14</title>
    <meta charset="UTF-8">
</head>
<body>
<h1>Objetos</h1>
<p id="nombre"></p>
<p id="tipoCombustible"></p>
<p id="velocidad"></p>
<p id="estado"></p>

<script>
var coche = {
    // Propiedades
    nombre: "BMW",
    tipoCombustible : "Gasolina",
    cantidadCombustible: 0,
    velocidad: 0,
    modelo: "318",
    potencia: 135,
    peso: 815,
    color: "azul",
    estado : "Detenido",
    // Métodos
    repostar: function(n) {
        this.cantidadCombustible=n;
    },
    arrancar : function() {
        this.estado = "En marcha";
    },
    detener : function() {
        this.estado = "Detenido";
    },
    frenar: function() {
        this.estado = "Frenando";
    },
    acelerar: function(n) {
        this.velocidad = n;
        this.estado = "En marcha";
    }
};

document.getElementById("nombre").innerHTML = coche.nombre;
document.getElementById("tipoCombustible").innerHTML =
coche.tipoCombustible;
coche.acelerar(120);
document.getElementById("velocidad").innerHTML = coche.velocidad;
document.getElementById("estado").innerHTML = coche.estado;

</script>
</body>
</html>
```

## ESTRUCTURAS CONDICIONALES

Durante el flujo de ejecución de un script por diversos motivos puede interesarnos realizar un conjunto de instrucciones u otras en base a condiciones. JavaScript, al igual que otros lenguajes de programación convencional, utiliza la sentencia *if* para estos casos.

### javascript15.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Javascript 15</title>
    <meta charset="UTF-8">
</head>
<body>
<h1>Adivina un número entre 1 y 10:</h1>
<input id="entrada" type="text" placeholder="Escribe número:">
<button id="boton" onclick="comprueba()">Probar!</button>
<p id="resultado"></p>
<script>
var numero=8;
function comprueba() {
    var entrada=document.getElementById("entrada").value;
    if (entrada>numero)
    {
        document.getElementById("resultado").innerHTML = "Lo siento,
el número era menor...";
    }
    else if(entrada==numero)
    {
        document.getElementById("resultado").innerHTML =
"¡Acertaste!";
    }
    else {
        document.getElementById("resultado").innerHTML = "Lo sentimos,
el número era mayor...";
    }
}
</script>
</body>
</html>
```

Observe que:

- El conjunto de instrucciones a ejecutar si una condición es verdadera se encierra entre los caracteres '{' y '}'.
- Podemos utilizar *else if* y *else* para encadenar otras comparaciones.

### Ejercicio:

Modifique el script para que el número elegido por el sitio web sea aleatorio. Busque en la web cómo generar un número aleatorio en JavaScript en un rango determinado.

Podemos considerar múltiples condiciones haciendo uso de la sentencia selectiva *switch*.

### javascript16.html

```
<html>
<head>
  <title>Javascript 16</title>
  <meta charset="UTF-8">
</head>
<body>
<p id="demo"></p>
<script>
  var text;
  var fecha = new Date();
  switch (fecha.getDay()) {
    case 0:
      text = "Hoy es domingo";
      break;
    case 1:
      text = "Hoy es lunes";
      break;
    case 2:
      text = "Hoy es martes";
      break;
    case 3:
      text = "Hoy es miércoles";
      break;
    case 4:
      text = "Hoy es jueves";
      break;
    case 5:
      text = "Hoy es viernes";
      break;
    case 6:
      text = "Hoy es sábado";
      break;
    default:
      text = "Deseando que llegue el finde";
  }
  document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

Observe que:

- La palabra reservada *case* sirve para evaluar distintas condiciones.
- Con *break* conseguimos que no se sigan evaluando otras condiciones y se termine el bloque *switch*.
- El comportamiento por defecto si no encaja en ninguna de las condiciones es el determinado bajo *default*.

**Ejercicio:**

Realice una página web que nos permita introducir un número de DNI y automáticamente calcule su letra.

Tenga en cuenta lo siguiente:

- El DNI suministrado por el usuario debe tener 8 dígitos, en caso contrario mostraremos un mensaje de error.
- La letra del DNI se calcula dividiendo el número entre 23 y obteniendo su resto. Dicha operación matemática en JavaScript se realiza con el operador módulo %. Las letras se obtienen de un array donde el índice se corresponde con el resultado del resto anterior:

RESTO	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
LETRA	T	R	W	A	G	M	Y	F	P	D	X	B	N	J	Z	S	Q	V	H	L	C	K	E

**Ejercicio:**

Realice una página web que permita calcular el número de habitaciones necesarias para un conjunto de personas. Tenga en cuenta que las personas pueden ser adultos o niños y que disponemos de este tipo de habitaciones:

- Dobles: un adulto o dos(habitaciones dobles).
- Dobles con supletoria: Dos adultos y un niño.
- Triples: Tres adultos.
- Familiares: Dos adultos y dos niños.

El sitio web debe mostrar un mensaje indicando el número de habitaciones dobles, dobles con supletoria, triples o familiares necesarias para tales circunstancias.