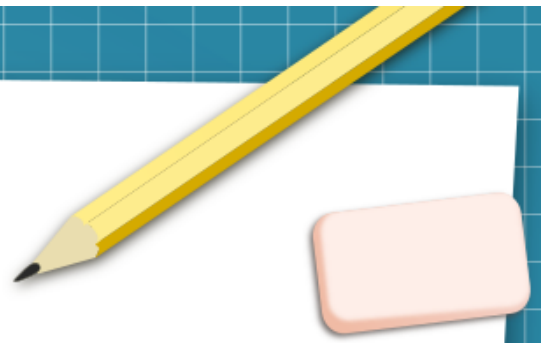
UD2.- UTILIZACIÓN DE OBJETOS PREDEFINIDOS DE JAVASCRIPT.

2º curso – CFGS Desarrollo de Aplicaciones Web

Profesora: María Ojeda García

Objetivos de la unidad



- Conocer las diferentes **alternativas** existentes para la **navegación web** en función de las diferentes tecnologías web que se ejecutan en un cliente.
- Conocer cuáles son los principales **objetos predefinidos** de JavaScript.
- Comprender las **propiedades y métodos** de los principales **objetos** de JavaScript.
- Aprender a **generar código HTML** desde sentencias JavaScript.
- Manipular y gestionar la creación y apariencia de las **ventanas del navegador**, además de la comunicación entre ellas.

2.1. INTRODUCCIÓN

DATOS PRIMITIVOS

```
var cadena = "Ada Lovelace";
var pi = 3.14;
var bool = true; //false
var nodefinida = undefined;
var nula = null;
```

OBJETOS NATIVOS

```
//OBJETOS NATIVOS: no dependen del navegador.
/* - String
   - Number
   - Boolean
   - Data
   - Math
   - RegExp (Expresiones regulares)
   - Array
   - Function
   - Object */
```

OBJETOS DE ALTO NIVEL

```
//OBJETOS DE ALTO NIVEL: dependen del navegador.
/* - Window
   - Screen
   - Navigator
   - Location
   - History
   - Document */
```

2.2. OBJETOS. CONSTRUCTORES INTEGRADOS.

```
//1. String:
var x1 = new String(); //No utilizar
var y1 = "Ada Lovelace"; //Utilizar

//2. Numbre:
var x2 = new Number(); //No utilizar
var y2 = 3.14; //Utilizar

//3. Boolean:
var x3 = new Boolean(); //No utilizar
var y3 = true; //Utilizar

//4. Array:
var x4 = new Array(); //No utilizar
var y4 = []; //Utilizar

//5. RegExp:
var x5 = new RegExp(); //No utilizar
var y5 = /<expresion>/; //Utilizar

//6. Function:
var x6 = new Function(); //No utilizar
var y6 = function (); //Utilizar

//7. Date:
var x7 = new Date();

//8. Math: no se puede declarar con "new" porque es un objeto.
```

2.3. OBJETOS NATIVOS. String.

Instanciación:

Podemos utilizar comillas dobles o simples:

```
var daw = "Desarrollo de aplicaciones web";
var dam = 'Desarrollo de aplicaciones multiplataforma';
var asir = "Administración de 'Sistemas Informáticos' en Red";
var smr = "Sistemas \"Microinformáticos\" y Redes";
var ciclos = new String("");
```

Concatenación:

```
ciclos += "Hay 3 ciclos de Grado Superior: \n";
ciclos += daw + ", " + dam + " y " + asir + "\n";
ciclos += " y 1 ciclo de Grado Medio: \n";
ciclos += smr;
```

Propiedades:

```
//length: devuelve la longitud de una cadena
alert ("La longitud de la cadena ciclos es: "+ciclos.length);
```

2.3. OBJETOS NATIVOS. String.

Métodos:

Método	Descripción
charAt()	Devuelve el carácter especificado por la posición que se indica entre paréntesis.
charCodeAt()	Devuelve el Unicode del carácter especificado por la posición que se indica entre paréntesis.
concat()	Une una o más cadenas y devuelve el resultado de esa unión.
fromCharCode()	Convierte valores Unicode a caracteres.
indexOf()	Devuelve la posición de la primera ocurrencia del carácter buscado en la cadena.
lastIndexOf()	Devuelve la posición de la última ocurrencia del carácter buscado en la cadena.

2.3. OBJETOS NATIVOS. String.

Métodos:

Método	Descripción
match()	Busca una coincidencia entre una expresión regular y una cadena y devuelve las coincidencias o null si no ha encontrado nada.
replace()	Busca una subcadena en la cadena y la reemplaza por la nueva cadena especificada.
search()	Busca una subcadena en la cadena y devuelve la posición dónde se encontró.
slice()	Extrae una parte de la cadena y devuelve una nueva cadena.
split()	Divide una cadena en un array de subcadenas.
substr()	Extrae los caracteres de una cadena, comenzando en una determinada posición y con el número de caracteres indicado.
substring()	Extrae los caracteres de una cadena entre dos índices especificados.
toLowerCase()	Convierte una cadena en minúsculas.
toUpperCase()	Convierte una cadena en mayúsculas.

2.3. OBJETOS NATIVOS. String.

Métodos:

Método	Descripción
includes()	Devuelve true is una cadena contiene una subcadena
localeCompare()	Devuelve un número que indica si la cadena de referencia viene antes, después o es equivalente a la cadena dada
padEnd()	Rellena la cadena actual desde el final, con otra cadena
padStart()	Rellena la cadena actual desde el principio con otra cadena
replaceAll()	Busca una subcadena en la cadena y reemplaza todas subcadenas
repeat()	Devuelve una cadena con una serie de copias de esa cadena
startsWith()	Devuelve true si es una cadena comienza con la subcadena especificada
trimEnd()	Elimina los espacios que hay al final de una cadena
trimStart()	Elimina los espacios que hay al comienzo de una cadena

2.3. OBJETOS NATIVOS. Number.

El objeto Number no se suele instanciar; normalmente asignamos valores numéricos a una variable.

Solo hay un tipo de dato numérico: 34, 34.04, 123e4, etc…

Crear:

```
let num=new Number(23);
let num2=23;
```

Si el parámetro que se pasa al constructor no se puede convertir, devuelve el valor NaN (Not and Numeric).

```
let num=new Number("a23"); //devuelve NaN
```

2.3. OBJETOS NATIVOS. Number.

Propiedades:

Propiedad	Descripción
constructor	Devuelve la función que creó el objeto Number.
MAX_VALUE	Devuelve el número más alto disponible en JavaScript.
MIN_VALUE	Devuelve el número más pequeño disponible en JavaScript.
NEGATIVE_INFINITY	Representa a infinito negativo (se devuelve en caso de overflow).
POSITIVE_INFINITY	Representa a infinito positivo (se devuelve en caso de overflow).
prototype	Permite añadir nuestras propias propiedades y métodos a un objeto.

2.3. OBJETOS NATIVOS. Number.

Métodos:

Método	Descripción
toExponential(x)	Convierte un número a su notación exponencial.
toFixed(x)	Formatea un número con x digitos decimales después del punto decimal.
toPrecision(x)	Formatea un número a la longitud x.
toString()	Convierte un objeto Number en una cadena. •Si se pone 2 como parámetro se mostrará el número en binario. •Si se pone 8 como parámetro se mostrará el número en octal. •Si se pone 16 como parámetro se mostrará el número en hexadecimal.
valueOf()	Devuelve el valor primitivo de un objeto Number.

2.3. OBJETOS NATIVOS. Math.

El objeto Math permite realizar operaciones matemáticas. Math no es un constructor, por tanto, no permite instanciar un objeto.

```
let numPi = Math.PI; //devuelve el número PI

let raizC= Math.sqrt(16); //devuelve la raíz cuadrada de 16
```

2.3. OBJETOS NATIVOS. Math.

Propiedades:

Propiedad	Descripción
E	Devuelve el número Euler (aproximadamente 2.718).
LN2	Devuelve el logaritmo neperiano de 2 (aproximadamente 0.693).
LN10	Devuelve el logaritmo neperiano de 10 (aproximadamente 2.302).
LOG2E	Devuelve el logaritmo base 2 de E (aproximadamente 1.442).
LOG10E	Devuelve el logaritmo base 10 de E (aproximadamente 0.434).
PI	Devuelve el número PI (aproximadamente 3.14159).
SQRT2	Devuelve la raíz cuadrada de 2 (aproximadamente 1.414).

2.3. OBJETOS NATIVOS. Math.

Métodos:

Método	Descripción
abs(x)	Devuelve el valor absoluto de x.
acos(x)	Devuelve el arcocoseno de x, en radianes.
asin(x)	Devuelve el arcoseno de x, en radianes.
atan(x)	Devuelve el arcotangente de x, en radianes con un valor entre -PI/2 y PI/2.
atan2(y,x)	Devuelve el arcotangente del cociente de sus argumentos.
ceil(x)	Devuelve el número x redondeado al alta hacia el siguiente entero.
cos(x)	Devuelve el coseno de x (x está en radianes).
floor(x)	Devuelve el número x redondeado a la baja hacia el anterior entero.
log(x)	Devuelve el logaritmo neperiando (base E) de x.

2.3. OBJETOS NATIVOS. Math.

Métodos:

Método	Descripción
max(x,y,z,...,n)	Devuelve el número más alto de los que se pasan como parámetros.
min(x,y,z,...,n)	Devuelve el número más bajo de los que se pasan como parámetros.
pow(x,y)	Devuelve el resultado de x elevado a y.
random()	Devuelve un número al azar entre 0 y 1.
round(x)	Redondea x al entero más próximo.
sin(x)	Devuelve el seno de x (x está en radianes).
sqrt(x)	Devuelve la raíz cuadrada de x.
tan(x)	Devuelve la tangente de un ángulo.

2.3. OBJETOS NATIVOS. Date.

El objeto Date se utiliza para trabajar con fechas y horas.

- `var d = new Date();`
- `var d = new Date(milisegundos);`
- `var d = new Date(cadena de Fecha);`
- `var d = new Date(año, mes, día, horas, minutos, segundos, milisegundos);`
  `// (el mes comienza en 0, Enero sería 0, Febrero 1, etc.)`

2.3. OBJETOS NATIVOS. Date.

Propiedades:

Propiedad	Descripción
constructor	Devuelve la función que creó el objeto Date.
prototype	Te permitirá añadir propiedades y métodos a un objeto.

2.3. OBJETOS NATIVOS. Date.

Métodos:

Método	Descripción
getDate()	Devuelve el día del mes (de 1-31).
getDay()	Devuelve el día de la semana (de 0-6).
getFullYear()	Devuelve el año (4 dígitos).
getHours()	Devuelve la hora (de 0-23).
getMilliseconds()	Devuelve los milisegundos (de 0-999).
getMinutes()	Devuelve los minutos (de 0-59).
getMonth()	Devuelve el mes (de 0-11).
getSeconds()	Devuelve los segundos (de 0-59).

2.3. OBJETOS NATIVOS. Date.

Métodos:

Método	Descripción
getTime()	Devuelve los milisegundos desde media noche del 1 de Enero de 1970.
getTimezoneOffset()	Devuelve la diferencia de tiempo entre GMT y la hora local, en minutos.
getUTCDate()	Devuelve el día del mes en base a la hora UTC (de 1-31).
getUTCDay()	Devuelve el día de la semana en base a la hora UTC (de 0-6).
getUTCFullYear()	Devuelve el año en base a la hora UTC (4 dígitos).
setDate()	Ajusta el día del mes del objeto (de 1-31).
setFullYear()	Ajusta el año del objeto (4 dígitos).
setHours()	Ajusta la hora del objeto (de 0-23).

2.3. OBJETOS NATIVOS. Boolean.

El objeto Boolean se utiliza para convertir un valor no booleano, en un valor booleano (true o false).

```
let bool=new Boolean(1);
let semaforo =true;
console.log(bool.toString());   //devuelve "true"
console.log(semaforo.valueOf()); //devuelve true
```

2.4. OBJETOS ALTO NIVEL (BOM). Windows.

Características:

- Representa una ventana abierta en un navegador.
- Si una ventana tiene etiquetas de tipo <iframe> el navegador crea un objeto window para el html inicial y uno para cada <iframe>
- Todos los objetos, funciones, variables... son miembros del objeto window: las funciones son sus métodos; las variables globales y el DOM son propiedades.
- No todos los métodos sirven para todos los navegadores: hay que comprobar la compatibilidad.

2.4. OBJETOS ALTO NIVEL (BOM). Windows.

El objeto window tiene una serie de **propiedades**:

- **name**: representa el nombre de la ventana.
- **outerWidth** y **outerHeight**: son el ancho y alto de la ventana incluyendo la barra de herramientas y la de scroll.
- **innerWidth** e **innerHeight**: similar a los anteriores pero sin incluir la barra de herramientas ni la de scroll.
- **pageXOffset** y **pageYOffset**: nos indica dónde se encuentra situado el scroll horizontal y vertical, respectivamente.
- **screenX** y **screenY**: permite conocer la distancia de la ventana desde la izquierda y desde arriba respectivamente.

2.4. OBJETOS ALTO NIVEL (BOM). Windows.

También hay una serie de propiedades con iframes y con otras ventanas que nos van a resultar muy de utilidad para poder conocer el estado de la ventana, quién la creó, etc. Son:

- **frames**: devuelve todos los iframe de la ventana.
- **frameElement**: devuelve el frame en el que está insertada la ventana.
- **length**: devuelve el número de frames que tiene la ventana.
- **closed**: devuelve un booleano que indica si la ventana está cerrada.
- **opener**: devuelve la referencia de la ventana que creó la ventana actual.
- **parent**: devuelve la ventana padre de la actual.
- **self**: devuelve la ventana actual

2.4. OBJETOS ALTO NIVEL (BOM). Windows.

Además, dentro de window se encuentran otros objetos del navegador que veremos más adelante:

- window.document.
- window.navigator.
- window.screen.
- window.history.
- window.location.

2.4. OBJETOS ALTO NIVEL (BOM). Windows.

Eventos de tiempo: nos permitirán ejecutar código asociado a instrucciones de tiempo.
Concretamente hay tres métodos:

- **setTimeout**: en el que indicamos la función que queremos ejecutar y el tiempo que tiene que transcurrir (en milisegundos) antes de que ésta se ejecute.
- **clearTimeout**: si el método anterior lo asignamos a una variable, a éste podemos pasarle esa variable para detener su ejecución.
- **setInterval**: con los mismos parámetros que setTimeout, en este caso repite una función cada vez que transcurre el intervalo de tiempo en milisegundos indicado.

2.4. OBJETOS ALTO NIVEL (BOM). Navigator.

El objeto Navigator contiene información sobre el navegador.

Propiedad	Descripción
`appCodeName`	Cadena que contiene el nombre en código del navegador.
`appName`	Cadena que contiene el nombre del cliente.
`appVersion`	Cadena que contiene información sobre la versión del cliente.
`cookieEnabled`	Determina si las cookies están o no habilitadas en el navegador.
`language`	Devuelve el idioma por defecto del navegador. "es" si es español o "en" si es inglés.
`onLine`	Indica si el navegador está offline u online. O sea que se ha desconectado de Internet usando la opción del navegador.
`platform`	Cadena con la plataforma sobre la que se está ejecutando el programa cliente.
`userAgent`	Cadena que contiene la cabecera completa del agente enviada en una petición HTTP. Contiene la información de las propiedades `appCodeName` y `appVersion`.

Métodos del objeto Navigator

Método	Descripción
`javaEnabled()`	Devuelve true si el cliente permite la utilización de Java, en caso contrario, devuelve false.

2.4. OBJETOS ALTO NIVEL (BOM). Screen.

Representa una referencia al objeto screen (pantalla) asociado a la ventana que está siendo visualizada. Y al igual que los objetos anteriores, no es necesario poner window.screen: si utilizamos screen directamente nos estaremos refiriendo a lo mismo.
Las propiedades más importantes de este objeto:

- **width**: representa el ancho de la pantalla.
- **height**: representa el alto de la pantalla.
- **availWidth**: representa el ancho de la pantalla sin la barra de tareas.
- **availHeight**: representa el alto de la pantalla sin la barra de tareas.
- **colorDepth**: indica la profundidad de color que estamos utilizando en nuestra pantalla.

Desarrollo Web en el Entorno Cliente