

# Ejercicios y Cuestiones



## Objetivos.-

- ☞ Dominar los conceptos fundamentales de la programación script cliente.
- ☞ Identificar y utilizar adecuadamente las estructuras básicas de programación de Javascript/DOM.
- ☞ Saber elaborar materiales, manuales y documentación en el marco de la programación en Javascript/DOM.
- ☞ Dado un problema en script cliente javascript/DOM:
  - ❖ Identificar y definir las clases y/o objetos propios del problema.
  - ❖ Identificar y organizar los eventos que permiten implementar la solución.
  - ❖ Elaborar un XHTML junto con los script cliente javascript/DOM que resuelvan el problema planteado.
  - ❖ Saber manipular la estructura de Modelo de Objetos del Documento.



## alert()

Crea una caja de diálogo con un icono de peligro amarillo, un botón 'Aceptar' y un texto definido por el parámetro enviado a la función.

### Ejemplo de alerta

```
000 <script>
001 alert("Ha ocurrido un error");
002 </script> </CODE< code>
```

Las alertas, nos serán útiles para transmitir información al usuario tal como errores ocurridos en la navegación, problemas en el rellenado de un formulario...

## confirm()

Crea una caja de confirmación con un icono de interrogación, botones Aceptar y Cancelar y un texto definido por el parámetro enviado a la función.

Devuelve 1 cuando el usuario abandona el diálogo pulsando Aceptar y 0 si lo hace pulsando Cancelar o el aspa de cerrar.

### Ejemplo de confirm()

```
000 <script>
001 if(confirm('¿Seguro que ha leído las condiciones del contra
002 to?'))
    this.form.submit();
</script>
```

Será útil para recibir información del usuario en tiempo de ejecución (al pulsar un botón, al pasar el mouse por un lugar...)

## prompt()

Muestra un diálogo de campo de formulario con botones Aceptar y Cancelar, un texto definido por el primer parámetro enviado a la función y un input de texto con valor predeterminado definido por el segundo parámetro.

La función devuelve el valor insertado en el campo de formulario si el usuario pulsa en Aceptar o null si pulsa Cancelar o el aspa de cerrar.

### Ejemplo de prompt()

```
000 <script>
001 nombre = prompt('Introduce tu nombre', '[ nombre del usuario
002 ]');
</script>
```

Prompt nos será útil sobre todo para recoger datos del usuario para utilizar en nuestro script en tiempo de ejecución.



## Introducción a la Programación Javascript

1. Elabora una lista de 5 WebReferencias de tutoriales Javascript.
2. Tratamiento de variables en Javascript.
3. Elabora una tabla con los distintos operadores que podemos utilizar en Javascript.
4. Describe las reglas de construcción de identificadores, pon ejemplos de variables bien y mal definidas.

5. Indica el tipo de datos que se expresa a continuación

- ❖ 12, 5.7, 's', "dd", 'juan antonio'
- ❖ var a1=new Array();
- ❖ var a2=[1,"2",'juan antoni']
- ❖ a2[0]
- ❖ a2[2].chartAt(0)

6. Supuesto que tenemos una única página XHTML muestra con alert(); o con el método write del objeto documento –document.write(), el resultado de las siguientes expresiones aritméticas, alfanúmericas, relacionales y lógicas. Por otra parte establece el orden de prioridad y por tanto de ejecución de los operadores utilizados.

- ❖ 27 % 4 +15 / 4
- ❖ 37 \ 4 + 4 + (-7)
- ❖ 9 + 2 / 3 \* 25 \* 3
- ❖ (7 \* 3 – 4) – 45
- ❖ (27 % 4) +15 / 4 <= 4 + (-7)
- ❖ 9 + (2 / 3) > 3 \* (25 \*-3)
- ❖ (7 \* 3 % 45) – 45 != 2 / 3
- ❖ 'a' < 'z'
- ❖ "Juan" == "Juan"
- ❖ 25<=7 && !(7>=2) || false
- ❖ !("Juan\_" < "pepe" && (37 < 4 + (-7))) || ('3' < 'z')
- ❖ 24 < 5 && 10 >=10 || 10 = 5 || false
- ❖ (! (6 / 3 > 3) || 7 < 7) && ( 3 > 9 / 2 || 2 + 3 <= 7 / 2) && true
- ❖ 6 – 3 \* 4 / 4 + 5 \* 2
- ❖ (2 \* 3 + (6 / 3 – 1 ))
- ❖ ((a % b) / c) \* (d + e). Define las variable e inicialízalas
- ❖ ((a / b) \* c) – d
- ❖ (a + (b \* (c + d ))) – (( e + f ) / g \* h)

7. Define e inicializa una variable para cada una de las siguientes situaciones

- La edad de un niño.
- Las medidas de un edificio.
- El nombre de una persona.
- La letra de su apartamento.
- Una variable que indique si alguien tiene hijos o no.



## Estructuras de Control – Programación Secuencial

8. Describe las diferentes estructuras que permiten controlar el flujo de ejecución .
9. Resuelve los siguientes problemas. Utiliza por ejemplo: alert(), prompt() , confirm() y document.write() y otros métodos o propiedades. Indicar que si la lectura se realiza con prompt() lo que recibe la variable es un texto por lo que hay que realizar una conversión a valor decimal, si así se precisa por el problema a resolver, para lo que podemos utilizar el método parseInt() [http://www.w3schools.com/jsref/jsref\\_obj\\_global.asp](http://www.w3schools.com/jsref/jsref_obj_global.asp)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Programa de Javascript</title>
</head>
<body>
  <p> Descripci&oacuten del problema</p>
  <script language="javascript">
    // Sentencias que resuelve la cuestión planteada

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

### JavaScript Objects

[JS Array](#)  
[JS Boolean](#)  
[JS Date](#)  
[JS Math](#)  
[JS Number](#)  
[JS String](#)  
[JS RegExp](#)

- ▣ Leer un número y mostrar si dicho número es o no es par.
- ▣ Leer un número y mostrarlo en octal y hexadecimal.
- ▣ Leer 2 números y mostrar el producto de ellos.
- ▣ Leer 2 números y determinar el mayor de ellos.
- ▣ Leer 3 números y mostrar el mayor de ellos.
- ▣ Leer un número y mostrar su tabla de multiplicar.
- ▣ Leer una secuencia de 30 números y mostrar la suma y el producto de ellos.
- ▣ Leer una secuencia de números, hasta que se introduce un número negativo y mostrar la suma de dichos números.
- ▣ Leer dos números y realizar el producto median sumas.
- ▣ Lee una secuencia de números y determina cual es el mayor de ellos. Véase [JS Number](#)
- ▣ Calcular la media de una secuencia de números proporcionado por el usuario.
- ▣ Leer una secuencia se números y mostrar cuáles de ellos es el mayor y el menor. Véase [JS Number](#)
- ▣ Leer un número y determinar su factorial.
- ▣ Leer un número y determinar si es o no es primo.
- ▣ Leer una secuencia de 30 números y mostrar la suma de los primos.
- ▣ Leer una secuencia de 30 números y mostrar la suma de su factorial.
- ▣ Leer una secuencia de n números almacenarlos en un vector y mostrar la posición donde se encuentra el mayor valor leído.



- ▣ Dado dos vectores A y B de 15 elementos cada uno, obtener un vector C donde la posición i se almacene la suma de  $A[i]+B[i]$ .
- ▣ Dado una secuencia de número leídos y almacenados en un vector A mostrar dichos números en orden. Véase [JS Array](#)
- ▣ Dado una secuencia de número leídos y almacenados en un vector A y un número leído determinar si dicho número se encuentra o no en el vector.
- ▣ Dado dos matrices A y B de enteros obtener la suma.
- ▣ Dado una matriz determinar la posición (i,j) del mayor.
- ▣ Dado una matriz determinar la posición (i,j) del mayor y menor.
- ▣ Dado un vector de secuencias de caracteres mostrar la longitud de cada una de ellas y
- ▣ Dado una secuencia determina si es palíndromo.
- ▣ Dado un DNI proporcionado por el usuario determinar si es correcto o no.
- ▣ Dado un E-mail determinar si cumple el formato.

### Ejemplo de recorrido de un vector de enteros:

```
<script>

var vector=new Array();
var aux=prompt("Introduce el número de elementos del vector","n°
elementos");
var valor;

/* Recorrido de un vector de tamaño inicialmente 0, al cual,
se incorporar tantos enteros como el valor decimal de la variable aux */

for(i=0;i<parseInt(aux);i++)
{
    valor=prompt("Introduce un valor en el vector: ","introduce valor");
    vector[i]=parseInt(valor);
}

</script>
```

### Ejemplo de recorrido de una matriz de enteros:

```
<script>

var matriz=new Array();
var filas=prompt("Introduce el número de filas de la matriz","n°
elementos");
var columnas=prompt("Introduce el número de columnas de la matriz","n°
elementos");
var valor;

for(f=0;f<parseInt(filas);f++)
{
    matriz[f]=new Array();
    for(c=0;c<parseInt(columnas);c++)
    {
        valor=prompt("Introduce m["+f+"]["+c+"]: ","introduce valor");
        matriz[f][c]=parseInt(valor);
    }
}

</script>
```



</script>

## Funciones – Programación Modular

10. Definición, concepto y estructura de una **function()**
11. Resuelve problemas planteados en el anterior punto pero cuya resolución se realizará mediante el concepto de programación modular (**function()**).

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Programa de Javascript</title>
  <script language="javascript">
    function nombre_funcion(var1,var2,..)
    {
      return valor_retorno;
    }
  </script>
</head>
<body>
  <p>Descripci&oacuten del problema</p>
  <script language="javascript">
    // Sentencias adicionales para la resolución del problema
    alert(nombre_funcion(parametro1,parametro2,..));
    document.write(alert(nombre_funcion(parametro1,parametro2,..)));
  </script>
</body>
</html>
```

- Función que calcule el factorial de un valor.
- Función que determine si un valor es o no es primo.
- Función que determine si una secuencia determina es palíndromo.
- Función que determine si un DNI es correcto o no.
- Función que determine si un E-mail si cumple el formato.
- Función que dado un vector de temperaturas devuelva un vector con aquellas temperaturas < 0º grados.
- Función que dado un vector y un entero determine si se encuentra o no.
- Función que dado un vector de palabras y una palabra determine si se encuentra o no.
- Función que dado un vector de palabras las ordene lexicográficamente.
- Función que dado un entero devuelva todos sus divisores.

### JavaScript Objects

[JS Array](#)  
[JS Boolean](#)  
[JS Date](#)  
[JS Math](#)  
[JS Number](#)  
[JS String](#)  
[JS RegExp](#)  
[JS Global](#)



### Ejemplo de una función y un objeto Array():

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Documento sin título</title>
<script>
function limpiar_matriz (m)
{
for (f=0;f<m.length;f++)
{
    for (c=0;c<m[f].length;c++)
    {
        matriz[f][c]=0;
    }
}
}
</script>
</head>
<body>
<script>
var matriz=new Array();
var filas=prompt("Introduce el número de filas de la matriz","nº
elementos");
var columnas=prompt("Introduce el número de columnas de la matriz","nº
elementos");
var valor;
for (f=0;f<parseInt(filas);f++)
{
    matriz[f]=new Array();
    for (c=0;c<parseInt(columnas);c++)
    {
        valor=prompt("Introduce m["+f+"]["+c+"]: ","introduce valor");
        matriz[f][c]=parseInt(valor);
    }
}
//Se envia como parámetro una matriz que se verá modificada en sus valores
limpiar_matriz(matriz);
alert(matriz[0][0]);
</script>
</body>
</html>
```

12. Analiza los métodos (funciones) que me proporciona cada una de los siguientes objetos:

#### JavaScript Objects

[JS Array](#)

[JS Boolean](#)

[JS Date](#)

[JS Math](#)

[JS Number](#)

[JS String](#)

[JS RegExp](#)



## Eventos

13. Define el concepto de evento.
14. Elabora un resumen de los eventos y pon ejemplos.

## HTML - DOM

15. Árbol de nodos. Programación DOM
16. Métodos de **Acceso directo a los nodos**.
17. Describe *La interfaz Node*.
18. Pon un ejemplo de creación, eliminación y sustitución de nodos del XHTML.
19. Pon un ejemplo de acceso directo a los atributos y propiedades CSS de los nodos del XHTML
20. Uso de método .innerHTML

Ejemplo:

```
<script type="text/javascript">
function changeText() {
    document.getElementById('boldStuff').innerHTML = 'Fred Flinstone';
}
</script>
<p>Welcome to the site <b id='boldStuff'>dude</b> </p>
<input type='button' onclick='changeText()' value='Change Text' />
```

21. A qué etiqueta representa esta expresión. Pon un ejemplo:

```
document.getElementById('navigation').getElementsByTagName('a')[3];
document.getElementsByTagName('div')[2].getElementsByTagName('p')[0];
document.getElementById('nav').childNodes[3].firstChild;
document.getElementById('nav').parentNode.previousSibling.firstChild.childNodes[2];
```

22. Elabora un XHTML para cada una de las siguientes cuestiones:

- Su puesto un <div> con border definido. Define un mecanismo de forma que el usuario pueda mover 10px a la izquierda, arriba, abajo o la derecha.
- Dado un XHTML con un formulario en el que hay un <select> de los días del mes y otro <select> con los meses. Suponemos que la carga de sus valores se realiza en cuando se ejecuta el evento onload().
- Realiza el ejercicio anterior utilizando: document.write();
- Tras 5 segundos aparecerá un <div> con imagen de fondo en una position:absolute. setTimeout(), setInterval()
- Elabora un <div> en el que aparezca un reloj.
- Implementa el cambio de idioma del contenido de una capa. Utiliza <span lang="">
- Dado un XHTML en el que tenemos un <form> de registro de un usuario definir las comprobaciones que se consideren oportunas respecto a los valores incorporados por el usuario.  

```
<form name="reg" action="prueba.php" onsubmit="return valida(this);">
<form name="reg" action="prueba.php" id="reg" onsubmit="return valida();">
```
- Cuando el ratón pase por encima de una imagen que sea sustituida por otra y que al salir el curso del área la imagen anterior se recupere.
- Cuando se onload() la página se creará un <div> el cual se mueva en vertical a intervalos de 2 segundos. setTimeout(), setInterval()





- ❑ Página web con un <form> para realizar un registro de un usuario. Se le añadirá un enlace que permita antes de realizar el <input type="submit"../> verificar la validez de los datos introducidos. Aquel <input> valor incorrecto se mostrará un <p> explicativo.
- ❑ Un XHTML con un <div><p>../></div> que se mueva de manera pendular (arriba /abajo o derecha/izquierda)
- ❑ Un xhtml de forma que cuando se ejecute el evento onload() del <body> un imagen cambie su src en base a un valor aleatorio. Utiliza la función `random` y un vector de nombres de ficheros.
- ❑ Un formulario con un <input>, de tal manera que cada vez que pulse una tecla verifique si es un valor dígito y si no lo es se eliminará del value. Utiliza el evento `onkeypress="comprobar(this)"` la función `charCodeAt()` del JS String. O utilizar el gestor de eventos 7.2.5. *Restringir los caracteres permitidos en un cuadro de texto.*
- ❑ Dado una capa que se mueva x px mediante la utilización de 4 teclas que representarán las 4 direcciones. 6.3. *Obteniendo información del evento (objeto event)*
- ❑ Opción para que todas los <div> se les defina un border.
- ❑ Dado un <input> y un <select> cada texto proporcionado al <input> se incorpore como <option> del <select>.
- ❑ Elabora un menú de forma que cuando el ratón pase por las opciones principales se haga visible el submenú asociado.
- ❑ Implementa el juego adivina.
 

```
var a=new Number(Math.random()*100);
a.toFixed();
```
- ❑ Cambiar el href de todos los <a> de la página.
- ❑ Eliminar todos los <p> de una determinada capa <div>.
- ❑ Dado un XHTML en el que tenemos un formulario de subida de ficheros del que inicialmente solo se ha habilitado para subir uno. Añade los elementos necesarios que permitan que el usuario pueda añadir tantos como considere oportunos.
- ❑ Insertar en una capa <div> un <img> al principio o al final. Las propiedades y atributos se obtienen del usuario.
- ❑ Eliminar el contenido completo de la página que se está visualizando.
- ❑ Eliminar todos los enlaces que hay en la página.

```
nodohijo.parentNode.removeChild();
nodohijo.parentNode.replaceChild();
nodohijo.parentNode.insertBeforeChild();
```

## Elaboración de la práctica.-

- ✚ La fecha límite para la entrega de la práctica será el día examen de la UT.
- ✚ El documento en formato **.doc** y con nombre **GrupoXX\_javascript.doc**.
- ✚ La documentación (.doc) entregada debe tener las páginas numeradas, e incluir una portada con:
  - ✚ Código de GRUPO.
  - ✚ Nombre completo y DNI de los componentes del grupo.
  - ✚ Dirección de correo electrónico.
  - ✚ Titulación y nombre de la Asignatura.
  - ✚ Curso académico (2010/2011) y convocatoria (febrero/junio/septiembre)
  - ✚ Nombre del profesor.



- ✚ El documento deberá contener un índice en que deberá aparecer entre otros elementos:
  - ✚ .....
  - ✚ Webgrafía.
- ✚ **Cada ejercicio planteado se resolverá de la siguiente forma:**
  - Ejercicios 1.-**
    - Enunciado.-**
    - Recursos Necesarios.**
    - Referencias/Webgrafía.**
    - Solución.**
- ✚ El formato del documento será considerado como un elemento más en la evaluación de la práctica.
- ✚ La resolución de cada apartado deberá ser concisa, clara y auto-explicada. Podrá añadirse los comentarios que el alumno considere necesarios con el fin de plantear adecuadamente la solución.
- ✚ Los alumnos pueden enriquecer la especificación de la práctica a partir del conocimiento que vayan adquiriendo.
- ✚ En los casos, si los hubiera, en que el enunciado del problema sea ambiguo o no sea lo suficientemente completo, el grupo deberá recoger por escrito las suposiciones que se adopten, indicando las razones de su elección y/o decisiones.
- ✚ Utilizar la *WWW* como herramienta de consulta que permita resolver las cuestiones que puedan surgir respecto al dominio del problema a resolver.

