

01. ¿Qué es Javascript?
- Un lenguaje de programación web en el lado del cliente.**
 - Un framework para el desarrollo de aplicaciones web.
 - Un derivado del lenguaje Java para programar en el lado del servidor.
02. ¿Qué elemento se encarga de procesar el código Javascript de una página?
- Un servidor de aplicaciones compatible con el estándar JEE.
 - La máquina virtual Java instalada en el cliente.
 - Un navegador web, como por ejemplo Mozilla FireFox o Google Chrome.**
03. Selecciona todas las respuestas que permitirían hacer uso de Javascript en una determinada página (seleccionar también las consideradas como obsoletas).
- <script> código </script>.**
 - <script language="JAVASCRIPT"> código </script>.**
 - <script type="text/javascript"> código </script>**
 - <style type="javascript"> código </style>.
 - <%-- código --% >.
04. Para importar código Javascript desde un fichero externo a la página actualmente en desarrollo, ¿Qué opción u opciones nos permitiría hacerlo?
- No es posible importar código externo Javascript ya que sólo puede utilizarse el incrustado en la página.
 - <script type="text/javascript" import="ficheroExternoJavaScript.js"/>
 - <script type="text/javascript" src="ficheroExternoJavascript.js"></script>**
05. ¿Qué es DHTML?
- Un software depurador de código Javascript.
 - Un API de programación Javascript para manejar con facilidad el DOM de la página.**
 - Dynamic HyperText Markup Language.**
 - Un framework de maquetación para establecer los estilos de la página.
06. ¿Hay algún error al ejecutar el siguiente código (suponer página Html completa)?

```
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>MASTER JAVA</title>

  <script>
    var capaMensajes = document.getElementById("mensajes");
    capaMensajes.innerHTML = "HOLA MUNDO !!!";
  </script>

</head>

<body>
  <div align="center">
    <div id="mensajes"></div>
  </div>
</body>
```

- Sí, el control gráfico referenciado no existirá para el bloque Javascript produciéndose un error, lo que hará que no se muestre nada en pantalla.**
 - No, la variable del bloque Javascript se cargará con el control gráfico referenciado, se modificará el contenido HTML del mismo y en pantalla se mostrará HOLA MUNDO !!!.
07. Elige la secuencia que sigue la ejecución de una página en el navegador web, correspondiendo BODY al cuerpo de la misma, JAVASCRIPT al bloque de código Javascript definido en su cabecera y ONLOAD al evento de carga del citado cuerpo de dicha página.
- BODY → JAVASCRIPT → ONLOAD.
 - JAVASCRIPT → ONLOAD → BODY.
 - JAVASCRIPT → BODY → ONLOAD.**

08. ¿Cuáles de las siguientes afirmaciones son correctas?

- a. **"Javascript" es un nombre comercial para una tecnología que poco o nada tiene que ver con Java.**
- b. Al igual que HTML, no hace ninguna distinción entre mayúsculas y minúsculas en su sintaxis.
- c. **Posibilita la modificación de los estilos y el contenido de la página en tiempo de ejecución.**
- d. **No dispone de un mecanismo eficaz y propio para la depuración del código desarrollado.**
- e. **Hace uso de variables y funciones, además de un amplio conjunto de eventos.**
- f. **No permite al programador declarar los tipos de las variables ya que los infiere desde el valor empleado.**

09. ¿Qué indicará por pantalla la ejecución del siguiente código (suponer página Html completa)?

```
<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>MASTER JAVA</title>

<script type="text/javascript">

    var valor = "ABC";

    alert(valor + " ... " +
        (isNaN(valor) ? "NO ES UN VALOR NUMERICO" : "ES UN VALOR NUMERICO"));

</script>

</head>
```

- a. Se produce un error, no mostrándose nada en pantalla.
- b. **ABC ... NO ES UN VALOR NUMERICO.**
- c. ABC ... ES UN VALOR NUMERICO.

10. Si se desean generar números enteros aleatorios en el rango [0 .. 10], ¿qué expresión utilizarías?

- a. parseFloat(Math.random() * 9);
- b. parseInt(Math.random() * 9);
- c. parseInt(Math.random() * 10);
- d. **parseInt(Math.random() * 11);**

11. Selecciona qué sentencias, bloques y bucles de control permite utilizar la tecnología Javascript.

- a. **If - else if - else.**
- b. **switch-case.**
- c. **for, for-in.**
- d. **while, do-while.**

12. ¿Qué diferencia existe entre el atributo *name* y el atributo *id* al definir un control gráfico Html en la página?

- a. Ambos son equivalentes (funcionan igual) y establecen un identificador único en el DOM.
- b. El atributo *name* es único mientras que *id* puede repetirse para diversos controles gráficos.
- c. **Sólo puede asociarse un *id* por control gráfico, mientras que *name* puede repetirse para constituir grupos de elementos.**

13. Dado el siguiente código ¿Qué obtendremos sobre la variable *obtenido* (suponer página Html completa)?

```
<script type="text/javascript">

    var cesta = "LECHE#HUEVOS#AZUCAR";
    var obtenido = cesta.split("#");

    alert("OBTENIDO = " + obtenido);

</script>
```

- a. El código genera un error, no mostrándose nada en pantalla.
- b. **Un array con los valores "LECHE", "HUEVOS" y "AZUCAR" en dicho orden.**
- c. Un array con los valores "AZUCAR", "HUEVOS" y "LECHE" en dicho orden.
- d. Una cadena de texto cuyo valor es "LECHE,HUEVOS,AZUCAR".

14. Siendo la representación ASCII para el caracter A, el número 65, ¿qué observaremos en pantalla una vez ejecutado el siguiente código Javascript (suponer página Html completa)?

```
<script type="text/javascript">

    var datos = [65,66,67];
    var cadena = "";

    for (var i in datos) {
        cadena += String.fromCharCode(datos[i]);
    }

    alert(cadena);

</script>
```

- a. 656667
- b. ABC
- c. CBA
- d. abc
- e. El código no compilará, no mostrándose nada en pantalla.

15. Indica el valor mostrado en la ventana de alert tras ejecutar el siguiente código.

```
<script type="text/javascript">

    var cadena = "HOLA MUNDO";

    alert(cadena.lastIndexOf("O"));

</script>
```

- a. -1
- b. 0
- c. 1
- d. 9

16. ¿Qué debería implementar un escuchador *onsubmit* ligado a un formulario?
- a. **Una cláusula return para confirmar o denegar el envío a partir del booleano devuelto por la función llamada.**
 - b. Nada, salvo la propia lógica de negocio de la función, no siendo obligatoria una cláusula return.
 - c. El escuchador del evento debe asociarse al botón submit y no al formulario.
17. Si sobre un determinado control gráfico Html, cuyo *id* es *usuario*, se desea cambiar mediante Javascript su estilo de color de fondo a amarillo, ¿qué opción de las presentadas a continuación tendríamos que utilizar?
- a. `document.getElementById("usuario").focus.backgroundColor = "yellow".`
 - b. `document.getElementById("usuario").backgroundColor = "yellow".`
 - c. **`document.getElementById("usuario").style.backgroundColor = "yellow".`**
 - d. `document.getElementById("usuario").style.background-color : "yellow".`
18. ¿Es posible deshabilitar un elemento (como es el caso de un botón) mediante el uso de Javascript?
- a. No, todos los elementos del *body* están siempre activos ("enable") al cargarse la página, por lo que no es posible realizar dicha operación.
 - b. **Sí, utilizando DHTML y la propiedad *disabled* del elemento `document.getElementById("identificador").disabled = true.`**
 - c. Sí, pero sólo puede realizarse mediante CSS y antes de que la página se cargue en memoria (DOM), no pudiéndose más tarde modificar su estado.
19. Se necesita cambiar en tiempo de ejecución el tipo de un control *input*, que originalmente se definió como texto, a contraseña o *password*. ¿Es posible realizar tal operación con Javascript?
- a. **Sí, localizando el elemento y modificando el valor de su atributo *type* para establecerlo en "password". Así, podemos hacer uso de `document.getElementById("identificador").type = "password".`**
 - b. No, es imposible modificar el tipo de un elemento una vez se ha cargado este en memoria (DOM).
 - c. Se puede realizar la modificación pero sólo a través de CSS y en el momento de cargarse la página.

20. Indica cuáles de los siguientes escuchadores de eventos pueden aplicarse a un elemento de tipo caja de texto (<input type="text">), invocándose así la correspondiente función Javascript de control o escucha.
- onclick.**
 - onsubmit.
 - onchange.**
 - onload.
 - onfocus.**
 - onblur.**
21. Selecciona cuales de los siguientes métodos están disponibles a través de DHTML.
- getElementById("id").**
 - getElementsByName("name").**
 - getElementsByTagName("tagName").**
 - getElementsByClassName("className").**
22. ¿Qué se busca con el siguiente código tras hacer 'click' sobre un conjunto de controles de tipo "button" que pertenecen a un mismo formulario (elige todas las respuestas correctas)?

```
<script type="text/javascript">
    f_enviar(operacion) {
        var url = operacion + "Servlet";           // parámetro de valor "Entrar" o "Registrar".
        var elementoFormulario = document.getElementById("formulario"); // queda: "EntrarServlet" o "RegistrarServlet".
        elementoFormulario.action = url;
        elementoFormulario.submit();
    }
</script>
```

- El código es erróneo ya que sólo un botón de tipo *submit* es capaz de lanzar el formulario y nunca un *button*.
 - Mediante la pulsación de los botones se llamará a la función que modificará el atributo *action* del formulario en función del valor del parámetro recibido, para a continuación lanzar la petición.**
 - Constituir una "botonera" y que el usuario pueda solicitar varias operaciones desde un mismo formulario.**
23. ¿Cómo puede obtenerse una referencia al control que generó un determinado evento en Javascript?
- Es posible utilizar la clausula *this* como parámetro de entrada a la función de escucha o respuesta al evento, ya que representa al elemento que lanzó o generó dicho evento.**
 - Se puede hacer uso del parámetro *event* al llamar a la función de escucha y a partir de él, invocar a su atributo *.target*, obteniéndose así una referencia al control que generó el evento en cuestión.**
 - Consultando desde la función de escucha el atributo *origin* que es propio de todo control gráfico Html.
24. ¿De qué formas podemos definir y posteriormente crear objetos en Javascript?
- Mediante parejas clave-valor, admitiéndose como valores tanto elementos simples como funciones.**
 - A través de lo que se conoce como una "función de creación".**
 - Utilizando el operador *new* para instanciar los objetos previamente definidos en el código.**
 - No se permite crear objetos por parte del programador, sino simplemente utilizar los propios del sistema.
25. ¿Qué resulta de la ejecución del siguiente código?

```
<script type="text/javascript">
    function AvionCombate(modelo) {
        this.modelo = modelo;
        this.atacar = function(armamento) {
            return "EL AVION DE COMBATE " + modelo +
                " ATACA UTILIZANDO " + armamento;
        };
    }

    var avionCombate = new AvionCombate("F14-TOMCAT");
    alert(avionCombate.atacar("MISILES HELLFIRE"));
</script>
```

- EL AVION DE COMBATE F14-TOMCAT ATACA UTILIZANDO MISILES HELLFIRE.**
- Se produce un error no mostrándose nada en pantalla.
- EL AVION DE COMBATE null ATACA UTILIZANDO null.