**¿Cuál es la mejor definición de la variable \_REQUEST?**

\_REQUEST array que contiene las variables pasadas a través de cualquier mecanismo de entrada.

\_REQUEST array que contiene las variables pasadas por el método REQUEST.

\_REQUEST array que contiene las variables pasadas a traves del del metodo \_GET.

\_REQUEST array que restringe los valores del conjunto de datos del formulario a caracteres ASCII

Explicacion:

1. \_REQUEST array que contiene las variables pasadas a través de cualquier mecanismo de entrada.
2. \_GET array que contiene las variables pasadas por el método GET. Su uso es análogo al array HTTP\_GET\_VARS de versiones anteriores a las 4.2.0 de PHP (aun disponible pero obsoleto).
3. \_POST array que contiene las variables pasadas a través del método \_POST. Su uso análogo al array HTTP\_POST\_VARS de versiones anteriores a las 4.2.0 de PHP (aun disponible pero obsoleto).
4. El método get restringe los valores del conjunto de datos del formulario a caracteres ASCII. Solo el metodo post (con enctype=”multipart/form-data”) cubre el conjunto de caracteres [ISO10646] completo.

**¿Qué nos permite la función sesión\_register(nombre[,nombre])?**

**sesión\_register(nombre[,nombre]):** Devuelve true cuando la variable cuyo nombre se ha pasado como argumento a la llamada está registrada dentro de la sesión actual.

**sesión\_register(nombre[,nombre]):** Evita que la variable que ha pasado como argumento a la llamada sea salvada como parte de la sesión actual.

**sesión\_register(nombre[,nombre]):** Es un fichero de configuración global de PHP.

**sesión\_register(nombre[,nombre]):** Registra una nueva variable global para la sesión actual.

**Explicacion:**

1. **sesion\_is\_registered(nombre):** Devuelve true cuando la variable cuyo nombre se ha pasado como argumento a la llamada está registrada dentro de la sesión actual.
2. **sesion\_unregister(nombre):** Evita que la variable que ha pasado como argumento a la llamada sea salvada como parte de la sesión actual, es decir, la elimina de la sesión, pero sigue siendo una variable del script donde se ejecuta (para esto tendríamos que llamar a la función unset()).
3. La directiva register\_globals se usa como alternativa para acceder a las variables de sesión sin utilizar la variable global $\_SESSION.
4. **sesion\_register(nombre[,nombre]):** Devuelve true cuando la variable cuyo nombre se ha pasado como argumento a la llamada está registrada dentro de la sesión actual. En caso de que no exista una sesión, la crea realizando una llamada explicita a la función sesion\_star(), devuelve true cuando todas las variables pasadas como argumentos han sido creadas sin ningún problema.

**mysql\_fetch\_array permite:**

Devuelve el valor numérico del mensaje de error de la última operación MySQL

Obtener la información de una columna de un resultado y la devuelve como un objeto

Recupera una fila de resultados como un array asociativo, un array numérico o con ambos tipos de índices.

Obtener información del servidor MySQL.

Obtiene el nombre de la tabla de un campo

**Explicacion:**

1. mysql\_errno, devuelve el valor numérico del mensaje de error de la última operación MySQL.
2. mysql\_fetch\_field, obtiene la información de una columna de un resultado y la devuelve como un objeto.
3. **mysql\_fetch\_array permite,** Recupera una fila de resultados como un array asociativo, un array numérico o con ambos tipos de índices.
4. mysql\_get\_server\_info, obtiene información del servidor MySQL.
5. mysql\_tablename, obtiene el nombre de la tabla de un campo.

**¿Cuáles son los niveles de privilegios del comando GRANT?**

Global, Base de datos, Tabla, Columna

Usuario, grupos, otros

Propietario, Grupos, otros

Usuario, privilegiado

**Explicación:**

1. GRANT, cuatro niveles de privilegios: Global, Base de datos, Tabla, Columna.

Se utiliza para crear usuarios y concederle previlegios. La sintaxis general del comando GRANT es la siguiente:

GRANT privilegios (columnas)  
ON element TO nombre\_usuario IDENTIFIED BY 'contraseña'  
(whith grant option);

1. Usuario, grupos, otros (Niveles de privilegio en LINUX).
2. Propietario, Grupos, otros (Niveles de privilegio en MAC).
3. Usuario, privilegiado (Niveles de privilegio en CISCO).

**Una ventaja de usar JavaScript?**

JavaScript tiene mucho que ver con el lenguaje de programación Java.

JavaScript es una excelente solución para poner en práctica la validación de datos de un formulario en el lado del cliente.

 JavaScript es que este tiende a introducir una cantidad enorme de fragmentos de código en nuestros sitios web.

JavaScript se ejecuta dentro del navegador de los clientes, se puede utilizar para cambiar el aspecto de la pantalla en el dispositivo de los usuarios después que la página ha sido enviada por el servidor

Explicacion:

1. JavaScript no tiene nada que ver con el lenguaje de programación Java. (Definición)
2. JavaScript es una excelente solución para poner en práctica la validación de datos de un formulario en el lado del cliente. JavaScript puede desplegar en pantalla un mensaje popup para hacerle saber al usuario acerca de la omisión. (Ventaja)
3. JavaScript es que este tiende a introducir una cantidad enorme de fragmentos de código en nuestros sitios web.(Desventaja)
4. JavaScript se ejecuta dentro del navegador de los clientes, se puede utilizar para cambiar el aspecto de la pantalla en el dispositivo de los usuarios después que la página ha sido enviada por el servidor.(Desventaja)

**Cuando queremos almacenar una lista de valores en Javascript lo más adecuado es usar...**

1. Una instancia del objeto array.
2. Una variable cuyo nombre termine en un número o índice.
3. Una constante tipo lista.
4. Una variable cualquiera.

# **Tipos de variables**

Aunque todas las variables de JavaScript se crean de la misma forma (mediante la palabra reservada var), la forma en la que se les asigna un valor depende del tipo de valor que se quiere almacenar (números, textos, etc.)

**Arrays**

Un array es una colección de variables, que pueden ser todas del mismo tipo o cada una de un tipo diferente. Su utilidad se comprende mejor con un ejemplo sencillo: Si una aplicación necesita manejar los días de la semana, se podrían crear siete variables de tipo texto:

var dia1 = "Lunes";

var dia2 = "Martes";

……….

var dia7 = "Domingo";

**¿Qué es DOM?**

 DOM es una de las innovaciones que más ha influido en el desarrollo de las páginas web dinámicas y de las aplicaciones web más complejas.

DOM permite a los programadores web acceder y manipular las páginas XHTML como si fueran documentos XML. De hecho, DOM se diseñó originalmente para manipular de forma sencilla los documentos XML.

DOM se ha convertido en una utilidad disponible para la mayoría de lenguajes de programación (Java, PHP, JavaScript) y cuyas únicas diferencias se encuentran en la forma de implementarlo.

A través del DOM se puede acceder, por medio de Javascript, a cualquiera de estos elementos, es decir a sus correspondientes objetos para alterar sus propiedades o invocar a sus métodos.

Es importante destacar ahora que, dado que los niveles del DOM cambian de versión a versión del navegador y que las especificaciones se han entendido de manera distinta por las distintas organizaciones creadoras de los navegadores, se ha producido un marco donde trabajar con los objetos de la página difiere de un navegador a otro.

**Donde se escribe un JavaScript?**

Se incorpora entre las etiquetas <head></head>.

En el cuerpo de la página entre las etiquetas <BODY></BODY>, a esto se lo llama código incrustado.

Se puede introducir varios scripts en una misma página con etiquetas <script> diferentes.

Cuando existen varias funciones probadas y que pueden servir a más de un programa.

Todas las anteriores.

La programación de JavaScript se realiza en un documento HTML.

1. Se incorpora entre las etiquetas <head></head>.
2. En el cuerpo de la página entre las etiquetas <BODY></BODY>, a esto se lo llama código incrustado.
3. Se puede introducir varios scripts en una misma página con etiquetas <script> diferentes.

O se puede realizar un archivo externo:

1. Cuando existen varias funciones probadas y que pueden servir a más de un programa.
2. El archivo externo normalmente tiene la extensión .JavaScript y se lo incluye de la siguiente manera, suponiendo que el archivo con las funciones se llame externo.

<script src=”eterno.js”></script>

**¿Qué es un objeto en JavaScript?**

Es un método llamado en el momento de la creación de instancias.

Un objeto es una entidad, algo que existe, que tiene propiedades que lo definen y diferencian además, tiene métodos para gestionar sus propiedades y, en general, para manejar el objeto.

Diferentes objetos que podrían definir el mismo método o propiedad.

Este método atraviesa todas las propiedades enumerables de un objeto y su cadena de prototipo

Explicación:

**Constructor:** Es un método llamado en el momento de la creación de instancias.

Un objeto es una entidad, algo que existe, que tiene propiedades que lo definen y diferencian además, tiene métodos para gestionar sus propiedades y, en general, para manejar el objeto.

**Polimorfismo:** Diferentes Clases podrían definir el mismo método o propiedad.

**bucles [for...in](https://developer.mozilla.org/en-US/docs/JavaScript/Reference/Statements/for...in" \o "en-US/docs/JavaScript/Reference/Statements/for...in)**Este método atraviesa todas las propiedades enumerables de un objeto y su cadena de prototipo

**¿Cuáles son los tipos de declaraciones en JavaScript?**

**Var, let ,const**

**Hosting, string, array**

**Int, double, float**

**Number, symbol, null**

**Ninguna de las anteriores**

**Explicacion:**

## Declaraciones

Hay tres tipos de declaraciones en JavaScript.

[**var**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Sentencias/var)**:** Declara una variable, inicializa opcionalmente un valor.

**[let](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Sentencias/let" \o "The documentation about this has not yet been written; please consider contributing!):** Declara una variable local en un bloque de ámbito, inicializa opcionalmente un valor.

**[const](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Sentencias/const" \o "Declaración de una constante de solo lectura.):** Declaración de una constante de solo lectura.

**¿Qué realiza el efecto $.fn.slideToggle?**

**1. Muestra el elemento seleccionado.**

**2. Oculta el elemento seleccionado.**

**3. Da forma animada, cambia de opacidad del elemento seleccionado al 100%.**

**4. Muestra u oculta el elemento seleccionado con un movimiento de deslizamiento vertical.**

**5. Oculta el elemento seleccionado con un movimiento de deslizamiento vertical.**

**Explicacion:**

**1. $.fn.show** Muestra el elemento seleccionado.

**2. $.fn.hide** Oculta el elemento seleccionado.

**3. $.fn.fadeIn** De forma animada, cambia la opacidad del elemento seleccionado al 100 %.

**4. $.fn.slideUp** Oculta el elemento seleccionado con un movimiento de deslizamiento vertical.

**5. $.fn.slideToggle** Muestra u oculta el elemento seleccionado con un movimiento de deslizamiento vertical, dependiendo si actualmente el elemento está visible o no.

**¿Cuál es la mejor definición para la licencia MIT?**

1. Instituto de Tecnología de Massachusetts (MIT en inglés).
2. Es la licencia creada en principio por la Apache Software Foundation para publicar los paquetes del proyecto Apache, sin embargo en la actualidad es usada para muchos otros proyectos.
3. Es una licencia muy similar a la Mozilla Public License ya que busca un equilibrio entre licencias permisivas y fuertes, teniendo como objetivo principal hacerla atractiva para los ambientes de negocios.
4. Creada por la Mozilla Foundation para publicar la mayoría de sus aplicaciones y utilizada por muchos otros programas en la actualidad.
5. Es una licencia muy permisiva que admite el uso comercial, la redistribuición, la modificación, etc. La única condición es que proporcionemos una copia de la licencia con el software que hemos distribuido.

**Explicación.**

1. Acrónimo MIT(Instituto de Tecnología de Massachusetts).
2. **Apache 2.0**: Es la licencia creada en principio por la Apache Software Foundation para publicar los paquetes del proyecto Apache, sin embargo en la actualidad es usada para muchos otros proyectos. Se considera una licencia permisiva porque no requiere que los trabajos derivados sean publicados bajo la misma licencia y tampoco exige la liberación del código fuente.
3. **Eclipse Public License (EPL) 1.0**: Es una licencia muy similar a la Mozilla Public License ya que busca un equilibrio entre licencias permisivas y fuertes, teniendo como objetivo principal hacerla atractiva para los ambientes de negocios. No requiere ningún seguimiento en los cambios y sólo exige la publicación del código fuente cuando las modificaciones se consideran un trabajo derivado y no una extensión o un módulo separado.**Los trabajos derivados deben ser publicados siempre bajo la licencia EPL**. Esta licencia es usada por la Eclipse Foundation y nace como reemplazo a la Common Public License.
4. **Mozilla Public License (MPL) 2.0**: Creada por la Mozilla Foundation para publicar la mayoría de sus aplicaciones y utilizada por muchos otros programas en la actualidad. Es una licencia que busca un balance entre las licencias permisivas (como la BSD) y las copyleft fuertes (como la GPL).
5. **MIT (**Massachusetts Institute of Technology) .Es una licencia muy permisiva que admite el uso comercial, la redistribución, la modificación, etc. La única condición es que proporcionemos una copia de la licencia con el software que hemos distribuido.

**¿Qué realiza en siguiente código JavaScript?**

**<html>**

**<head>**

**<title></title>**

**</head>**

**<body id="cuerpo">**

**<h1 id="encabezado" align="center"></h1>**

**<div id="message" align="center">**

**<input type="text" id="txtA" value="xyz">**

**<input type="text" id="txtB" value="abc">**

**</div>**

**<script type="text/javascript" src="js/jquery.js"></script>**

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

**$(document).ready(function()**

**{**

**$("#txtA").focus(function(){**

**$(this).animate({opacity: 0.3 }, 400).css('background-color', '#4B8A08')**

**})**

**$("#txtA").blur(function(){**

**$(this).animate({opacity: 0.3 }, 400).css('background-color', '#DF0101')**

**});**

**$("#txtB").focus(function(){**

**$(this).animate({opacity: 0.3 }, 400).css('background-color', '#4B8A08')**

**})**

**$("#txtB").blur(function(){**

**$(this).animate({opacity: 0.3 }, 400).css('background-color', '#DF0101')**

**});**

**});**

**</script>**

**</body>**

**</html>**

1. **Nada**
2. **Genera error en la línea 30.**
3. **Oculta los textos que se muestran en pantalla, cuando se da clic sobre el botón Ocultar.**
4. **Muestra en un principio el texto sin ningún efecto, después al dar clic sobre el primer texto va a cambiar el fondo a color verde y al dar clic en el otro texto va a cambiar a color rojo, además también se muestra un efecto de transparencia en cada uno de los textos.**
5. **Colorea las filas pares de color negro y las impares de color gris.**

**¿Cuáles son las ventajas de usar Bootstrap?**

1. **El usar Bootstrap en el desarrollo web no presta ventajas, mejor es un dolor de cabeza.**
2. **Ofrece un paquete de elementos web personalizables**.

**Utiliza componentes vitales para los desarrolladores.**

**Se integra con librerías JavaScript**

1. Es necesario adaptarse a su forma de trabajo, si bien su curva de aprendizaje es liviana, deberás comprender y familiarizarte con su estructura y nomenclatura.

Debes adaptar tu diseño a un grid de 12 columnas, que se modifican según el dispositivo.

Aquí empiezan los problemas, Bootstrap por defecto te trae anchos, márgenes y altos de línea, y realizar cambios específicos es por decir, un poco tedioso.

1. Es complicado, cambiar de versión si has realizado modificaciones profundas sobre el core.

Si necesitas añadir componentes que no existen, debes hacerlos tú mismo en CSS y cuidar de que mantenga coherencia con tu diseño y cuidando el responsive.

Un problema que he visto a menudo en muchos Themes, es que el responsive funciona bien, pero a nivel de diseño en ocasiones se le escapan ciertos detalles que le quitan estética al sitio. Si eres un perfeccionista del diseño, ten por seguro que Bootstrap te va a dar algunos problemas.