



APRENDERAPROGRAMAR.COM

PROGRAMACIÓN WEB-
APP JAVASCRIPT.
LIBRERÍAS,
FRAMEWORKS. JQUERY,
ANGULARJS. VENTAJAS.
DIFERENCIAS. (CU01194E)

Sección: Cursos

Categoría: Tutorial básico del programador web: JavaScript desde cero

Fecha revisión: 2029

Resumen: Entrega nº94 del Tutorial básico “JavaScript desde cero”.

Autor: César Krall

LIBRERÍAS Y FRAMEWORKS JAVASCRIPT

Un programador que no conoce las bases de un lenguaje tiene difícil crear buenas aplicaciones usando ese lenguaje. Pero conocer las bases no significa que todo deba ser programado por nosotros. Cuando se trabaja a nivel profesional, entra en juego el concepto de productividad. El uso de librerías y frameworks es una forma de tener una buena productividad como programadores y de eso vamos a hablar a continuación.



PRODUCTIVIDAD COMO PROGRAMADORES

Una frase muy utilizada en los ambientes de programación es: “no trates de inventar la rueda”. Esta frase alude a que existen numerosas funcionalidades que son usadas con frecuencia por los programadores, están bien estudiadas y resueltas (posiblemente por expertos con un mayor nivel de conocimiento que nosotros). Dichas soluciones están disponibles a través de internet y por tanto es más inteligente y más productivo utilizar algo que funciona bien y está ya desarrollado que tratar de crearlo nosotros desde cero.

Veamos un ejemplo de una situación que se nos puede presentar como programadores: supongamos que desarrollamos una aplicación web para una asociación que tiene un salón de actos que puede reservarse por días. Queremos incluir un calendario en nuestra página web. Crear un calendario con un buen diseño y una buena funcionalidad (que muestre al abrirse la fecha actual, que permita avanzar o retroceder meses o años, etc.) puede requerir bastantes horas de trabajo, posiblemente días. Ahora bien, ¿es este un problema frecuente dentro de los trabajos de programación? Obviamente sí, muchas aplicaciones web trabajan con calendarios para permitir seleccionar fechas: por ejemplo aplicaciones para realizar reservas o compras en hoteles, restaurantes, etc. o para pedir cita con el médico o con una esteticista.

Esta tarea por ser muy común está bien estudiada y resuelta por la comunidad de programadores, por tanto no tiene sentido que tratemos de inventar la rueda construyéndonos nosotros mismos nuestro propio calendario (a no ser que tratemos de hacer algo innovador y exclusivo). Para resolver esta tarea lo más lógico será recurrir a código disponible a través de librerías JavaScript.

¿QUÉ ES UNA LIBRERÍA JAVASCRIPT? ¿QUÉ ES UN FRAMEWORK JAVASCRIPT? ¿QUÉ ES UNA API?

Entre los recursos que facilitan el trabajo a los programadores tenemos librerías, frameworks y apis. Estos términos muchas veces se entremezclan entre sí, ya que no existe un límite o definición clara para cada uno de ellos. Nosotros vamos a dar aquí unas definiciones, pero ten en cuenta que son sólo algo orientativo.

Librería: conjunto de código que contiene funciones que permiten realizar tareas que nos facilitan la programación. Por ejemplo, si tenemos que calcular el factorial de un número y tenemos una librería

que nos ofrece dicha función, simplemente tendremos que llamar `calcularFactorial(n)` para obtener el factorial de dicho número, sin necesidad de escribir el código que resuelve dicha tarea (ya que ha sido escrito por otra persona). Una librería muy conocida JavaScript es JQuery.

Framework: un framework (“marco de trabajo”) puede verse como una gran librería o conjunto de librerías donde además de facilitárse nos funciones para su uso, se suele disponer de una sintaxis o metalenguaje específico del framework y una forma de organización del código específica para dicho framework. Por tanto para usar un framework no basta con conocer el nombre de una función que queramos usar: hay que conocer qué sintaxis utiliza el framework, qué obligaciones impone (a la hora de organizar el código, archivos) y su lógica o filosofía de trabajo. El uso de frameworks se ha extendido debido a la gran complejidad de las aplicaciones que se desarrollan hoy en día: los frameworks facilitan su organización y mantenimiento. Un framework muy conocido JavaScript es AngularJS.

API: api (application programming interface o interface de programación para aplicaciones) suele aludir a la especificación de librerías amplias con funciones y métodos (signaturas) utilizadas en torno a una tecnología, usualmente por toda la comunidad de programadores, aunque no siempre. El código de implementación (es decir, el detalle de la programación) suele estar oculto o no disponible, pero sin embargo la especificación suele estar muy bien documentada. Una API puede estar asociado al lenguaje en sí, o bien haber sido creada por una empresa o casa comercial independiente. Son ejemplos de apis muy conocidas JavaScript la api Canvas y la api de Google Maps.

Además de estos términos es frecuente que en torno a la programación surjan otros que vamos a citar brevemente:

IDE: un IDE (integrated development environment ó entorno de desarrollo integrado) es una aplicación que facilita la programación en un lenguaje. Un IDE suele incluir un editor, herramientas de generación de código automática, de corrección sintáctica, de ejecución de código, depuración, etc. Son IDEs que se pueden usar en JavaScript Eclipse, Aptana, Codelobster, etc.

SDK: son las siglas de software development kit ó kit de desarrollo de software. Es un paquete de software que puede incluir varias cosas útiles para la programación en un determinado lenguaje, desde compilador, depurador, ejemplos, organizadores de ficheros, etc.

Toolkit: término que puede englobar librerías e incluso frameworks. Suele usarse para aludir a un conjunto de herramientas de desarrollo que facilita una empresa, o una comunidad de software libre. Estas herramientas pueden incluir cosas diversas, por ejemplo formatos prediseñados de botones, tablas, cajas de diálogo, etc. Un ejemplo de toolkit sería GWT (Google Web Toolkit).

Ten en cuenta que las definiciones anteriores son imprecisas en el sentido de que estos términos no tienen unos límites claros y distintas empresas o comunidades de programadores los utilizan de distinta manera.

Es imposible conocer todas las herramientas de tipo librería, framework, api, ide, sdk, toolkit, etc. para un lenguaje porque hay cientos. Una empresa (o una persona) elige una o varias de estas herramientas y desarrolla con ellas ignorando a las demás.

¿CUÁLES SON LAS MEJORES LIBRERÍAS Y FRAMEWORKS JAVASCRIPT?

Es difícil decir cuáles son los mejores frameworks o librerías JavaScript por varios motivos:

- a) Posiblemente dependa de cuáles sean nuestros objetivos en una aplicación concreta. Por ejemplo habrá librerías o frameworks que manipulen muy bien el DOM pero ofrezcan pocas funciones matemáticas, mientras que otros quizás ofrezcan muchas funciones matemáticas pero no manipulen bien el DOM.
- b) Las librerías y frameworks (y el resto de elementos en torno al software) están evolucionando continuamente. Por ello una librería que sea la más usada en 2089 puede dejar de serlo en 2090 debido al cambio y auge de unas frente a otras. ¿Cuál será la librería más usada el próximo año? No es fácil saberlo.
- c) Algunos proyectos de librerías y frameworks se abandonan por parte de sus creadores, quedando como “reliquias fósiles”.
- d) No todos los programadores tienen los mismos gustos o ideas: a unos les puede gustar algo que a otros no le gusta.

Lo que sí es cierto es que aunque sea difícil decir cuáles son las mejores, sí hay una forma indirecta que nos indica por lo menos la popularidad o apoyo que tienen librerías y frameworks entre los programadores. Esta forma es simplemente medir el número de descargas o de páginas web que usan una determinada librería o framework.

De acuerdo con esto, podemos citar las siguientes librerías y frameworks JavaScript como muy populares (ten en cuenta que el grado de popularidad va cambiando con el tiempo):

Librería	Logo	Descripción
Jquery		Librería o biblioteca JavaScript creada por John Resig en 2006 y que se ha convertido en una de las más usadas. Se complementa con jQuery UI, una biblioteca que facilita distintos componentes (menús desplegables, calendarios, etc.)
AngularJS		Framework JavaScript desarrollado por Google.
Backbone		Framework JavaScript creado por Jeremy Ashkenas.
Ember		Framework JavaScript creado por Yehuda Katz.

Existen decenas de librerías y frameworks que no hemos citado (como Foundation, React, Three, Meteor, Underscore...) . Existen librerías o frameworks que tienen fines muy específicos, por ejemplo si necesitamos crear gráficos (de barras, de líneas, circulares, etc.) sobre datos nos resultará de interés una librería especializada en la creación de gráficos como ChartJS, D3.js, ó Google Charts.

En resumen: el mundo de librerías y frameworks javascript es muy amplio, y no podemos pretender conocer ni saber usar todos ellos. Tendremos que conocerlos y utilizarlos a medida que los vayamos necesitando para proyectos concretos.

VENTAJAS E INCONVENIENTES DEL USO DE FRAMEWORKS JAVASCRIPT

El uso de frameworks JavaScript tiene ventajas e inconvenientes. Como ventajas podemos citar las siguientes:

- a) Elevan la productividad de los programadores: se pueden hacer cosas complejas en poco tiempo.
- b) Elevan la claridad y mantenibilidad del código. Al partir de un código bien pensado y estructurado, los proyectos resultan más fáciles de mantener.
- c) Reutilizan código creado por grandes programadores, posiblemente un código más efectivo y conciso del que nosotros seríamos capaces de crear.
- d) Código gratuito y abierto, en general con buena documentación para los programadores.
- e) Permiten el uso de funciones sin conocer el detalle del código (abstracción).

Y como desventajas:

- a) Los frameworks evolucionan e incluso en algunos casos dejan de ser mantenidos y dejan de ser usados. Esto genera un problema para el mantenimiento o continuidad de sitios web.
- b) Pueden generar sobrecarga. Si se utiliza un framework cuyo contenido y tiempo de carga resultan pesados para el navegador y por el contrario no se le saca partido, estaremos generando una sobrecarga innecesaria.
- c) Problemas de compatibilidad. Algunos frameworks son incompatibles entre sí y usar dos de ellos al mismo tiempo puede generarnos problemas. También pueden existir problemas de compatibilidad con algunos navegadores.
- d) Aprender a usar un framework es algo restringido y que únicamente nos sirve para manejar dicho framework, mientras que aprender un lenguaje de programación es algo universal que tendrá múltiples aprovechamientos y aplicaciones.
- e) Si los desarrolladores cambian y no manejan el framework que se venía empleando en un sitio web, puede existir dificultad para continuar el desarrollo (o mantenerlo si ya se había concluido) debido a que tendrán que adaptarse a las particularidades del framework que se estuviera empleando, rediseñar el sitio o como peor opción, hacer un collage de diferentes frameworks o estilos.

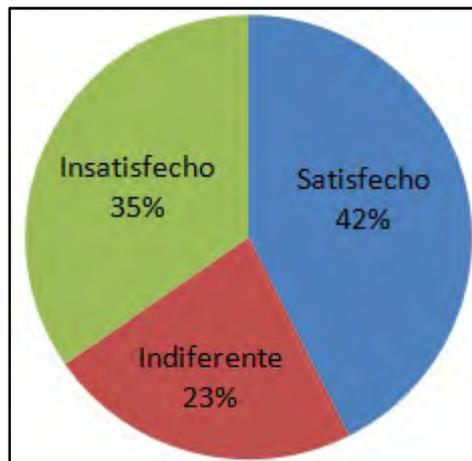
Si te preguntas si usar una librería o framework puede ser o no conveniente, la respuesta es que posiblemente dependa de lo que quieras hacer. Pero ten en cuenta que las grandes multinacionales como Google, FaceBook, Microsoft, etc. y las grandes empresas usan frameworks porque en general se acepta que las ventajas que aportan son mucho mayores que sus inconvenientes.

EJERCICIO OPCIONAL

Este ejercicio no forma parte del curso, por eso se califica como opcional. El ejercicio consiste en ver un breve ejemplo de uso de una librería JavaScript para creación de un gráfico.

Considera que como resultado de realizar una encuesta a los clientes de un hotel se han obtenido estos resultados: 35 % insatisfechos, 42 % satisfechos y 23 % indiferentes.

Se pide utilizar una librería o framework JavaScript especializada en la creación de gráficos para crear un gráfico circular que refleje dichos resultados. El resultado a obtener debe ser similar a este:



Para comprobar si tus respuestas y código son correctos puedes consultar en los foros aprenderaprogramar.com.

Próxima entrega: CU01195E

Acceso al curso completo en aprenderaprogramar.com -- > Cursos, o en la dirección siguiente:

http://aprenderaprogramar.com/index.php?option=com_content&view=category&id=78&Itemid=206