





PORTADA

RESPUESTAS

TUTORIALES

CURSOS



Home

Tutoriales

Desarrollo

JavaScript

Primeros pasos con Knockout.js

Primeros pasos con Knockout.js



En este tutorial estaremos dando los primeros pasos con Knockout.js, un framework que sigue el paradigma de desarrollo Modelo Vista VistaModelo una variación de MVC y nos ofrece la posibilidad de trabajar de forma muy ordenada, con lo cual le podemos dejar al lado cliente la lógica para nuestras páginas e interfaces de usuario.

Escrito por [Solvetic](#) feb 25 2015 04:19

⚡ TE INTERESA!

[Soluciones EXPRESS en YouTube a un PL...](#)

El traslado de la parte más dura de la aplicación al lado cliente está siendo cada vez más común, relegando así a los lenguajes de servidor a ser únicamente los que manejen la persistencia a través de servicios **REST**.

Parte de la responsabilidad que todo esto ocurra recae en las nuevas formas que se tienen de trabajar con **JavaScript** y la generación de interfaces de usuarios dinámicas y responsivas, gracias a los diferentes *frameworks* existentes.

Este framework tiene muchos puntos fuertes como lo son su bajo peso y su documentación, además que tiene una gran cantidad de tiempo en el mercado por lo que ha estado madurando de forma consistente para poder ser aplicado en entornos de producción.

Requisitos

1- Como la mayoría de las librerías y frameworks de **JavaScript** los requisitos no son tan exigentes, en este caso vamos a necesitar un pequeño servidor web para poder procesar de mejor forma nuestras peticiones, puede ser **Wamp** en *Windows* o una instancia de **Apache** en *Linux* o *Mac*.

2- Necesitamos un editor de textos enriquecido al estilo de *Sublime Text* o *NotePad ++* para poder escribir el código necesario para cumplir los ejemplos y ejercicios que estaremos mostrando.

3- Por último necesitamos acceso a Internet para poder hacer la descarga de **Knockout.js** aunque sea una vez.

Instalación

El proceso de instalación es bastante sencillo y no depende de una plataforma en específico, el proceso será el mismo ya sea que estemos en *Windows*, *Linux* o *MAC*.

Información

Publicado

feb 25 2015 04:19

X



↗ AMPLIAR

Últimos Tutoriales JavaScript

- Cómo actualizar NPM con PowerShell en Windows 10
- Como arrastrar y capturar imágenes con JQuery UI
- Cómo hacer un Spinner o icono de carga con JQuery
- Cómo tener una página web traducida en varios idiomas

VER MÁS DE JAVASCRIPT

Es importante que guardemos el archivo en una carpeta pertinente a nuestro proyecto para poder incluirlo. También se puede instalar mediante **Bower** o inclusive utilizarlo desde alguna de las *CDNs* de servicios para no tener que utilizar nuestro servidor como repositorio de la librería.

Una vez que ya tenemos el archivo simplemente hay que incluirlo en un archivo **HTML** de la siguiente forma:

```
<!DOCTYPE html><html lang="en"><head> <meta charset="UTF-8"> <title>Instalando Knockout.js</title>
```

Como vemos únicamente tuvimos que incluir el archivo descargado a través de una etiqueta script. En caso que queramos trabajar con la versión de la **CDN** solo debemos cambiar la ruta local por la ruta en Internet del archivo, esto es muy similar al trabajo con librerías como **jQuery** o frameworks como **AngularJS**.

Modelo Vista VistaModelo

El título de esta sección puede parecer un error, ¿Cómo es posible hablar de Modelo Vista VistaModelo? ¿No es igual? Resulta que **Knockout** tiene un patrón de diseño o paradigma de desarrollo donde tiene una inspiracion de **MVC**, donde ciertamente el *Modelo* y la *Vista* se comportan como lo que conocemos del enfoque “clásico” sin embargo tenemos una capa nueva llamada **VistaModelo** que es la encargada de unir y dar la funcionalidad a nuestras vistas.

Data Binding

Data Binding es la característica principal de nuestro framework que nos permite indicarle al documento *HTML* cuales son los elementos a los cuales debe dar acceso a **Knockout.js**, a pesar que suena como algo complejo, podemos resumirlo como algo más sencillo, es la simple aplicación de propiedades a los elementos del árbol *DOM* donde queremos que tenga actuación *Knockout.js*.

Veamos el siguiente código de ejemplo, digamos que queremos imprimir un nombre proveniente de nuestra **VistaModelo** en una etiqueta dentro del *body*, para ello solo debemos hacer lo siguiente:

```
/title> </head><body> <h1>Hola <span data-bind="text: nombre"></span></h1> <script src="js/knockout.js"></script>
```

Vemos que dentro de nuestra etiqueta **h1** hay una etiqueta **span** y esta última posee una propiedad que no es nativa de *HTML* que es la propiedad **data-bind**, esto es lo que hace que podamos conectar **Knockout.js** con nuestro *Árbol DOM* en este caso simplemente hemos indicado que se debe tomar la propiedad o variable nombre de nuestra **VistaModelo**.

Hola Mundo!



Con esto ya hemos realizado nuestra primera aplicación utilizando **Knockout.js**.

Un poco más de VistaModelo

Ya a estas alturas debemos habernos dado cuenta que la filosofía **VistaModelo** es una parte importante del trabajo en **Knockout.js**, es por ello vamos a seguir ahondando un poco más en el tema.

Creación de VistaModelo

Existen dos formas de crear una VistaModelo, la forma básica y la orientada a objetos, cual utilicemos ya depende de nuestra forma de desarrollar, aunque la recomendada es la **orientada a objetos** ya que aunque es más compleja en un inicio nos permite lograr un mejor orden en nuestra aplicación.

Para crear una **VistaModelo** básica solo debemos hacer algo como lo siguiente:

```
var miVistaModelo = { nombre: 'Pedro Perez'};
```

Cómo vemos no es nada complejo simplemente definimos una variable y le colocamos un contenido estilo **JSON**.

Ahora la forma **orientada a objetos** nos permite crear métodos de acceso y manipulación de las propiedades que deseemos colocar en nuestra **VistaModelo**, para ello podemos hacer algo como lo siguiente:

```
var miVistaModelo00 = function() { var self = this; self.nombre = 'Jonathan'; self.apellido =
```

Si queremos en este caso podemos acceder directamente al nombre o al apellido de forma individual, o utilizar el método *getNombre()* que nos da el nombre y el apellido unidos o concatenados.

Aquí quizás muchos se pregunten si no era más fácil hacer un solo *String* con el nombre y el apellido, la respuesta es muy simple, mientras más opciones tengamos, más sencilla será la aplicación en nuestra vista, imaginemos que solo hubiésemos querido el nombre, con la forma básica hubiésemos tenido que procesar la cadena para poder separarlos, en cambio orientado a objetos solo accedemos a la propiedad que nos

forma una **VistaModelo** puede servir para muchos fines, ya que no nos limitaremos a lo que establezcamos, si no que podemos ir cambiando los datos según la situación.

Vamos a rescribir nuestra forma *orientada a objetos*, y utilizaremos la siguiente forma:

```
var miVistaModelo00 = function(nombre, apellido) { var self = this; self.nombre = nombre; sel
```

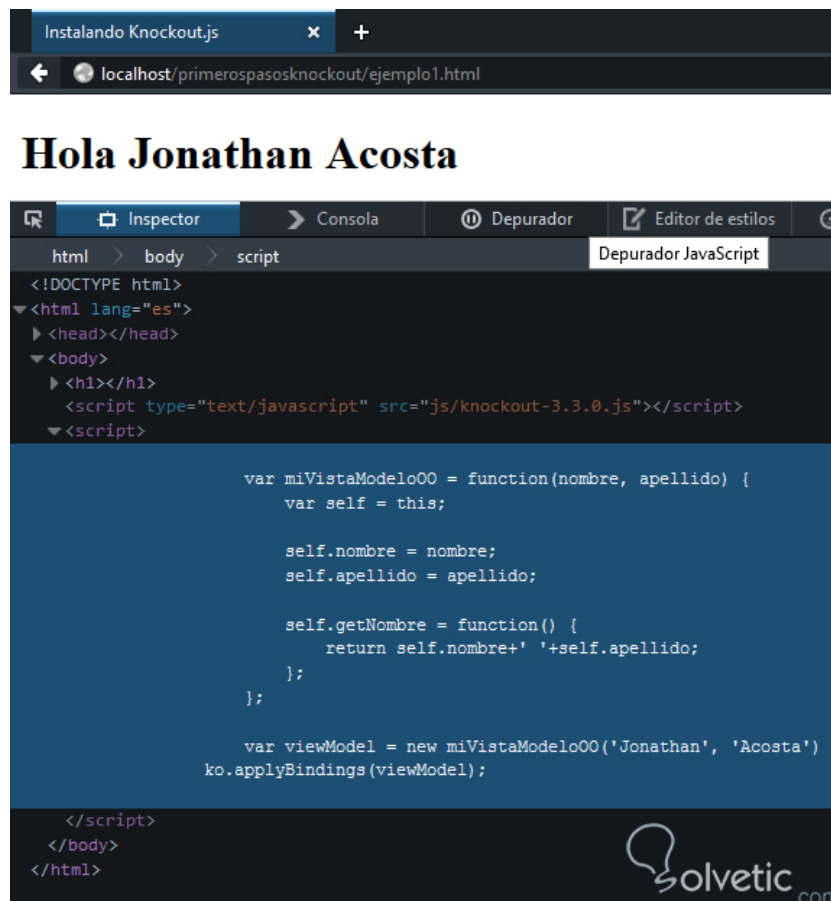
Cómo pudimos apreciar fue muy fácil implementar el uso de parámetros, ahora vamos a implementar esto en nuestro primer ejemplo y así ver como todo va encajando en su lugar, para ello utilizaremos el siguiente código HTML:

```
<!DOCTYPE html><html lang="es"><head> <meta charset="UTF-8"> <title>Instalando Knockout.js</t
```

Aquí lo primero que notamos es que el **data-bind** ya no es **text** como en el ejemplo inicial si no que utilizamos **html**, esto nos permite enviar etiquetas *html* y así poder dar más estilo a nuestra respuesta de la **VistaModelo**.

Ya en la sección donde definimos nuestra **VistaModelo** vemos que antes de aplicar el *binding* de **Knockout.js**, hacemos una instancia de nuestra VistaModelo **orientada a objetos**, es en este punto donde le podemos dar los valores a nuestros parámetros o propiedades, como vemos mantuvimos nuestra separación de nombre y apellido pero al final podemos utilizar el método que los concatena.

Cuando ejecutamos nuestro ejemplo nos da como resultado lo siguiente en el navegador:



[Hacer Pregunta](#)[Responder](#)[Identificarse](#)[Crear Cuenta](#)

Con esto finalizamos nuestro tutorial, hemos dado de forma exitosa nuestros primeros pasos con **Knockout.js**, con ello podremos empezar a pensar en todas las posibilidades que nos ofrece este framework de desarrollo de aplicaciones con la lógica del lado del cliente.

¿Te ayudó este Tutorial?

[Si](#)[No](#)

Sin comentarios, sé el primero!

No esperes más y entra en Solvetic

Deja tus comentarios y aprovecha las ventajas de la cuenta de usuario ¡Únete!

Crear Cuenta

Regístrate GRATIS para tener tu cuenta Solvetic

[Registrar una Cuenta](#)

Identificarse

¿Tienes ya cuenta? Identificate aquí

[Identificarme en mi Cuenta](#)

Solvetic es el punto de encuentro de profesionales y amantes de la tecnología. Una comunidad que te sorprenderá y ayudará en más de una ocasión

[Pregunta](#)[Responde](#)[Ranking](#)

SECCIONES

[Portada](#)[Respuestas](#)[Tutoriales](#)[Cursos](#)

OTROS ENLACES

[Sitemap](#)[Explorador](#)[Calendario](#)

BUSCADOR

[Política Cookies](#)[Aviso Legal](#)