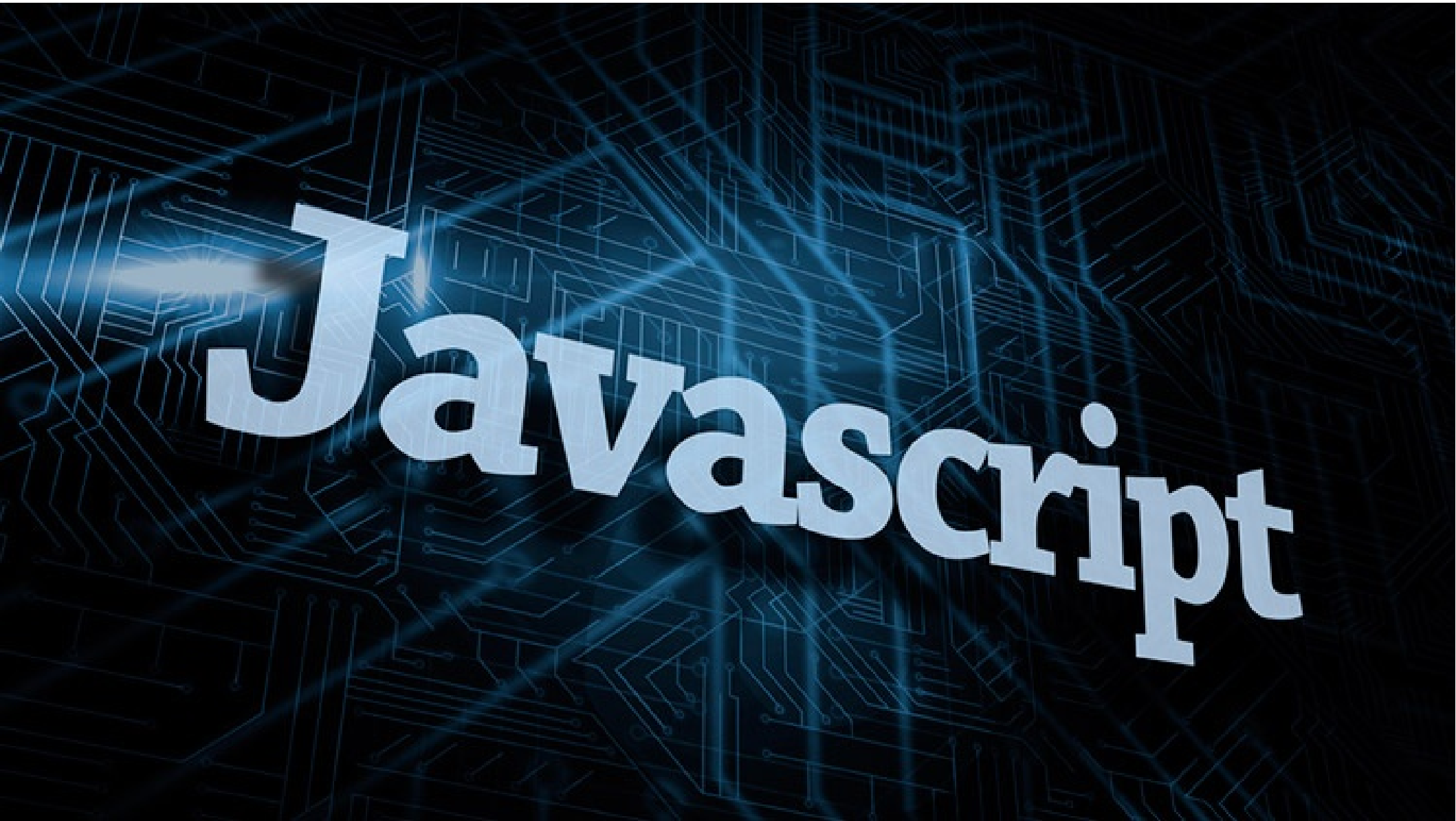
CURSO DESAROLLO WEB CON MEAN

WEB FULL STACK DEVELOPER

Germán Caballero Rodríguez
germanux@gmail.com

Introducción a JavaScript



INDICE

Qué es Javascript

- JavaScript (abreviado comúnmente JS) es un lenguaje de programación interpretado, dialecto del estándar ECMAScript.
- Se define como orientado a objetos,3 basado en prototipos, imperativo, débilmente tipado y dinámico.
-

Las necesidades de las aplicaciones web

- Las necesidades de las aplicaciones web modernas y el HTML5 ha provocado que el uso de Javascript que encontramos hoy haya llegado a unos niveles de complejidad y prestaciones tan grandes como otros lenguajes de primer nivel.

Un poco de historia

- Desde el 2012, todos los navegadores modernos soportan completamente ECMAScript 5.1, una versión de javascript.
- Los navegadores más antiguos soportan por lo menos ECMAScript 3.
- La sexta edición se liberó en julio del 2015
- JavaScript se diseñó con una sintaxis similar a C, aunque adopta nombres y convenciones del lenguaje de programación Java.
- Sin embargo, Java y JavaScript tienen semánticas y propósitos diferentes.

Un poco de historia

En el contexto de un sitio web, con Javascript puedes hacer todo tipo de acciones e interacción. Antes se utilizaba para validar formularios, mostrar cajas de diálogo y poco más. Hoy es el motor de las aplicaciones más conocidas en el ámbito de Internet:

- Google
- Facebook
- Twitter
- Outlook

Absolutamente todas las aplicaciones que disfrutas en tu día a día en la Web tienen su núcleo realizado en toneladas de Javascript.

La Web 2.0 se basa en el uso de Javascript para implementar aplicaciones enriquecidas que son capaces de realizar todo tipo de efectos, interfaces de usuario y comunicación asíncrona con el servidor por medio de Ajax.

Empleos de JavaScript

- Interfaces de usuario
- páginas web dinámicas
- validación del lado del cliente
- Animación
- *frameworks* para desarrollo de aplicaciones que requieren actualizar información en tiempo real
- Servidores web
- Internet de las cosas
- Y mucho más...

Empleos de JavaScript

- Con Javascript podemos crear efectos especiales en las páginas y definir interactividades con el usuario
- El navegador del cliente es el encargado de interpretar las instrucciones Javascript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, con que cuenta este lenguaje es el propio navegador y todos los elementos que hay dentro de una página

Empleos de JavaScript

- Ahora, gracias a las API Javascript del HTML5, que están disponibles en los navegadores actuales de ordenadores y dispositivos, podemos acceder a todo tipo de recursos adicionales, como la cámara, espacio para almacenamiento de datos, creación de gráficos basados en vectores y mapas de bits, flujos de datos con servidores, etc.

Empleos de JavaScript

- Javascript es un lenguaje con muchas posibilidades, permite la programación de pequeños scripts, pero también de programas más grandes, orientados a objetos, con funciones, estructuras de datos complejas, etc.
- Además, Javascript pone a disposición del programador todos los elementos que forman la página web, para que éste pueda acceder a ellos y modificarlos dinámicamente.
- Javascript es un lenguaje con muchas posibilidades, permite la programación de pequeños scripts, pero también de programas más grandes, orientados a objetos, con funciones, estructuras de datos complejas, etc.

Características

Imperativo y estructurado

- JavaScript es compatible con gran parte de la estructura de programación de C (por ejemplo, sentencias if, bucles for, sentencias switch, etc.).
- Con una salvedad, en parte: en C, el ámbito de las variables alcanza al bloque en el cual fueron definidas; sin embargo JavaScript no es compatible con esto, puesto que el ámbito de las variables es el de la función en la cual fueron declaradas.

Características

Imperativo y estructurado

- Esto cambia con la versión de ECMAScript 2015, ya que añade compatibilidad con block scoping por medio de la palabra clave let.
- Como en C, JavaScript hace distinción entre expresiones y sentencias.
- Una diferencia sintáctica con respecto a C es la inserción automática de punto y coma, es decir, en JavaScript los puntos y coma que finalizan una sentencia pueden ser omitidos.

Características

Dinámico

- Es de tipado dinámico
- Como en la mayoría de lenguajes de scripting, el tipo está asociado al valor, no a la variable.
- Por ejemplo, una variable x en un momento dado puede estar ligada a un número y más adelante, religada a una cadena.

Características

Objetual

- JavaScript está formado casi en su totalidad por objetos.
- Los objetos en JavaScript son arrays asociativos, mejorados con la inclusión de prototipos
- Los nombres de las propiedades de los objetos son claves de tipo cadena: obj.x = 10 y obj['x'] = 10 son equivalentes, siendo la notación con punto azúcar sintáctico.

Características

Objetual

- Las propiedades y sus valores pueden ser creados, cambiados o eliminados en tiempo de ejecución.
- La mayoría de propiedades de un objeto (y aquellas que son incluidas por la cadena de la herencia prototípica) pueden ser enumeradas a por medio de la instrucción de bucle for... in.
- JavaScript tiene un pequeño número de objetos predefinidos como son Function y Date.

Características

Evaluación en tiempo de ejecución

- JavaScript incluye la función eval que permite evaluar expresiones como expresadas como cadenas en tiempo de ejecución.
- Por ello se recomienda que eval sea utilizado con precaución y que se opte por utilizar la función JSON.parse() en la medida de lo posible, pues resulta mucho más segura.

Características

Funcional

- A las funciones se les suele llamar ciudadanos de primera clase; son objetos en sí mismos.
- Como tal, poseen propiedades y métodos, como .call() y .bind().
- Una función anidada es una función definida dentro de otra.
- Esta es creada cada vez que la función externa es invocada.

Características

Prototípico

- JavaScript usa prototipos en vez de clases para el uso de herencia.
- Es posible llegar a emular muchas de las características que proporcionan las clases en lenguajes orientados a objetos tradicionales por medio de prototipos en JavaScript.

Características

Prototípico

- Funciones como constructores de objetos:
  - Las funciones también se comportan como constructores.
  - Prefijar una llamada a la función con la palabra clave new crear una nueva instancia de un prototipo, que heredan propiedades y métodos del constructor (incluidas las propiedades del prototipo de Object).
  - ECMAScript 5 ofrece el método Object.create, permitiendo la creación explícita de una instancia sin tener que heredar automáticamente del prototipo de Object

Características

Prototípico

- Funciones como constructores de objetos:
  - La propiedad prototype del constructor determina el objeto usado para el prototipo interno de los nuevos objetos creados.
  - Se pueden añadir nuevos métodos modificando el prototipo del objeto usado como constructor.

Características

Prototípico

- Funciones como constructores de objetos:
  - Constructores predefinidos en JavaScript, como Array u Object, también tienen prototipos que pueden ser modificados.
  - Aunque esto sea posible se considera una mala práctica modificar el prototipo de Object ya que la mayoría de los objetos en Javascript heredan los métodos y propiedades del objeto prototype, objetos los cuales pueden esperar que estos no hayan sido modificados.

Características

Expresiones regulares

- JavaScript también es compatible con expresiones regulares de una manera similar a Perl, que proporcionan una sintaxis concisa y poderosa para la manipulación de texto que es más sofisticado que las funciones incorporadas a los objetos de tipo string.

Características

Funciones como métodos

- A diferencia de muchos lenguajes orientados a objetos, no hay distinción entre la definición de función y la definición de método.
- Más bien, la distinción se produce durante la llamada a la función; una función puede ser llamada como un método.
- Cuando una función es llamada como un método de un objeto, la palabra clave this, que es una variable local a la función, representa al objeto que invocó dicha función.

Características

Funciones variádicas

- Un número indefinido de parámetros pueden ser pasados a la función.
- La función puede acceder a ellos a través de los parámetros o también a través del objeto local **arguments**.
- Las funciones variádicas también pueden ser creadas usando el método .apply().

MVC en Javascript

- Siguiendo con la secuencia lógica de tu aprendizaje de Javascript, llegarás en este punto al MVC.
- Son las siglas de Modelo, Vista y Controlador y se trata de un paradigma de programación que se usa en lenguajes donde se tiene que trabajar con interfaces gráficas, como es el caso de la Web.
- Propone la separación del código de las aplicaciones por responsabilidades.

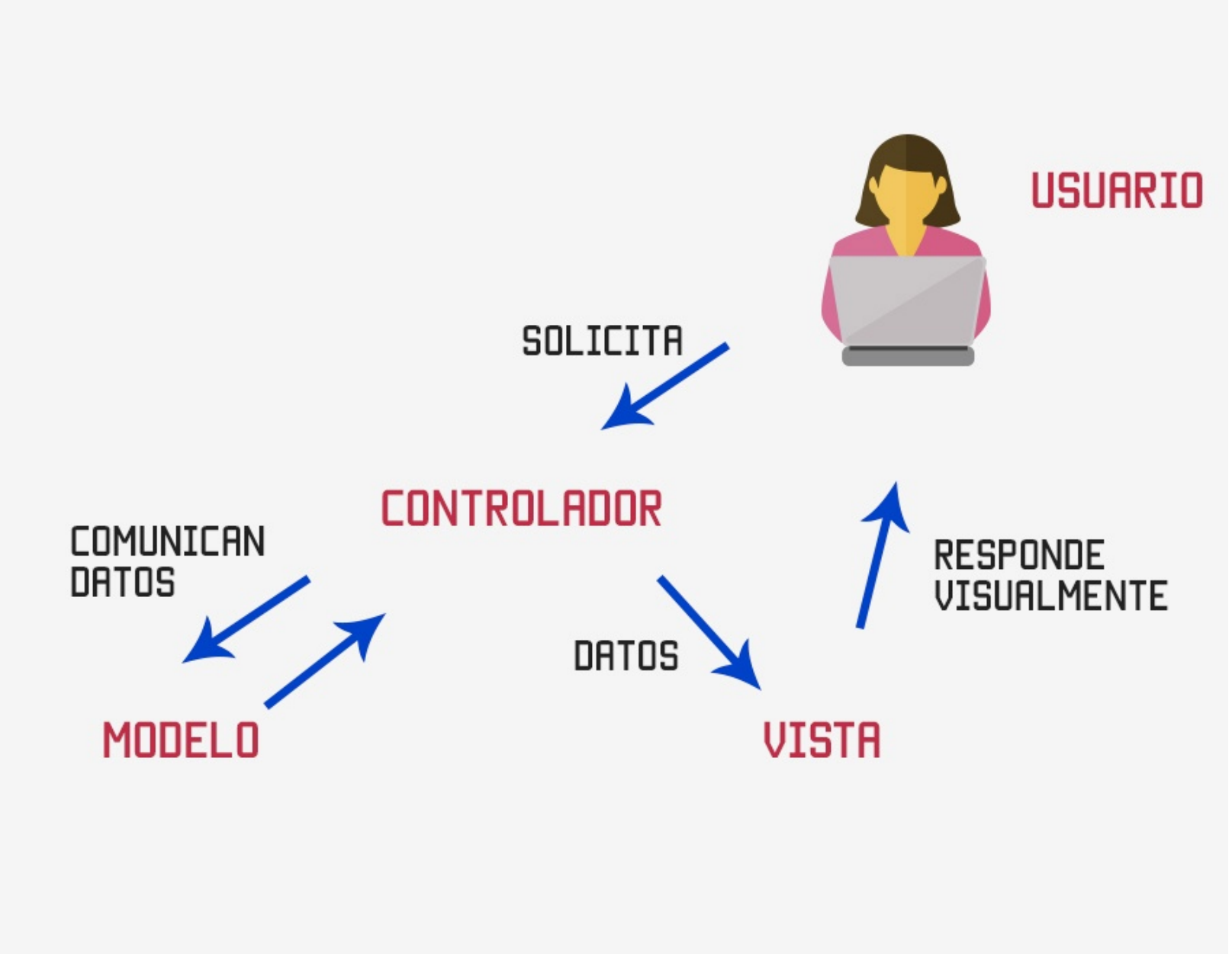
MVC en Javascript

- Los modelos se encargan de trabajar con los datos de la aplicación, las vistas con la presentación y los controladores hacen de conexión entre vistas y modelos. MVC no es algo específico de Javascript, sino que lo encontramos en lenguajes del lado del servidor como PHP o incluso en lenguajes de propósito general como es Java.
-

MVC en Javascript

- Trabajar con paradigmas como MVC es fundamental en el mundo de las aplicaciones web, porque nos permite organizar mejor nuestro código, facilitando el mantenimiento de las aplicaciones.
- Existen diversas librerías para realizar MVC en Javascript, entre las más populares están BackboneJS, EmberJS, AngularJS, KnockoutJS, NodeJS, etc.

MVC en Javascript



MVC en Javascript

HOW MODEL VIEW CONTROLLER DESIGN WORKS

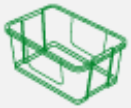
Upwork™

The MVC approach allows for clear separation of a web application's user interface (UI) code ("View") from its data and business logic ("Model") with scripts that send requests and updates between the two ("Controller"). Separating objects into these three bins allows for clean code that's easy to maintain.

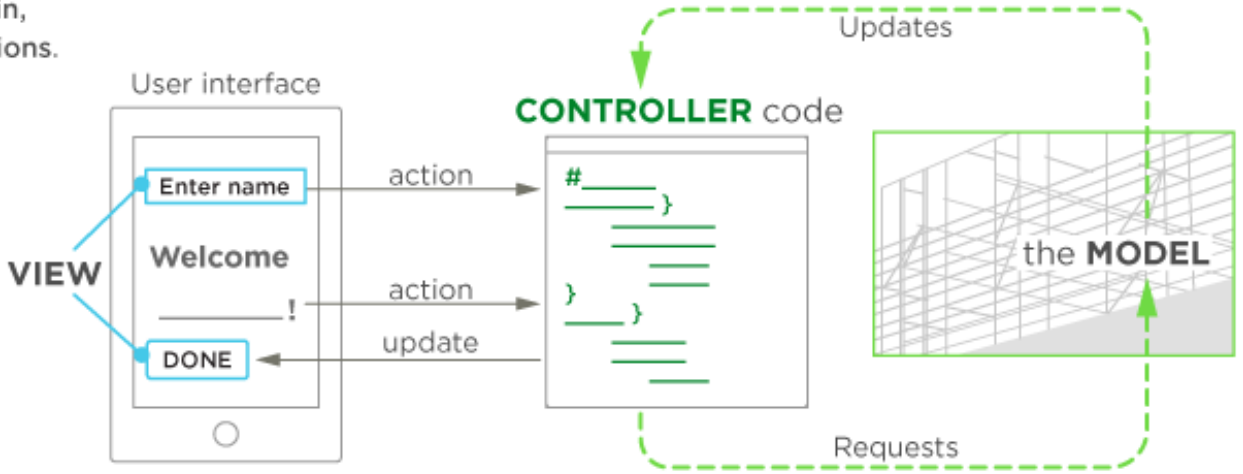
Each object should be written for **one bin, and one bin only—no overlapping functions.**

MODEL
data and server-side applications

VIEW
format and layout of data (UI)



CONTROLLER
communicates between the two



MVC en Javascript