

¿Qué es Vanilla Js?

Vanilla Js (JavaScript) es lo que se conoce *como JavaScript puro* o sus fundamentos. Es un código que se ejecuta del lado del cliente sin ayuda de ningún framework o bibliotecas para su desarrollo. Es importante conocer los fundamentos de JS para poder aprender más rápido el uso de nuevos frameworks.

¿Buscar 3 FrameWorks de Javascript que se usen en la actualidad?

FrameWorks de Fron-ted

* React.js

Se puede utilizar como base en el desarrollo de aplicaciones móviles o de una sola página.

Características clave: DOM Virtual – es multiplataforma (IOS y Android) – React Flux que permite una actualización de la vista del cliente muy efectiva.

*Vue.js

Se puede utilizar para crear interfaces de usuario y aplicaciones de una sola página. También llamado JavaScript progresivo.

Características clave: API de composición - Componente de archivo único (SFC) - Transición/Animación - Propiedades calculadas

* Ember.js

Sirve para un desarrollo web sea más productivo y agradable.

Características clave: sus Componentes - Enrutamiento - Servicio (se puede crear un servicio de terceros en nuestra aplicación Ember que esté disponible durante todo el ciclo de vida de la aplicación) - Ember Data Model.

*Netx.js

Sirve para construir interfaces de usuario y sitios web estáticos.

Características clave: Páginas (se puede crear paginas con el enrutamiento de archivos) - Optimización de imagen - Compatibilidad con TypeScript - Navegadores compatibles modernos - Obtención de datos - Diseño

*AngularJS

Fue diseñado para hacer que el desarrollo web sea más eficiente y fácil de usar. Angular.js se basa en la arquitectura Model-View-Controller (MVC), lo que facilita el desarrollo de aplicaciones web dinámicas.

Características clave: Enlace de datos - Directivas - Inyección de dependencia (DI) - Plantillas

FrameWorks de back-end

***Node.js**

Se utiliza para desarrollar aplicaciones del lado del servidor.

Características clave: Controlado por **eventos/Asíncrono** - Subproceso único - Sin almacenamiento en búfer - Muy rápido

*** Backbone.js**

Proporciona arquitectura y sincronización Model-View-Presenter (MVP) para aplicaciones web y se basa en el patrón de diseño Model View Controller (MVC).

Características clave: Funciones de JavaScript - Extensiones - Bibliotecas principales (jQuery y Underscore.js) - Modelos y vista – Colecciones

*** Meteor.js**

Sirve para desarrollar aplicaciones web.

Características clave: Desarrollo multiplataforma - Base Unicode – Paquetes - Meteor Galaxy (plataforma en la nube para implementar meteor apps).

***Express.js**

Diseñado para crear aplicaciones web y API.

Características clave: Ofrece solicitudes de enrutamiento de clientes simples – Funciona como un middleware