Vue.js Tutorial

Sumario

Introducción a Vue.js

- Página oficial: **https://vuejs.org/**
- Vue (pronunciado /vju?/, como view) es un framework progresivo para construir interfaces de usuario. A diferencia de otros frameworks monolíticos, Vue está diseñado desde cero para ser utilizado incrementalmente.
- La librería central está enfocada solo en la capa de visualización, y es fácil de utilizar e integrar con otras librerías o proyectos existentes.
- Por otro lado, Vue también es perfectamente capaz de impulsar sofisticadas Single-Page Applications (SPA) cuando se utiliza en combinación con herramientas modernas y librerías de apoyo.
- La forma más fácil de probar Vue.js es usando un **ejemplo básico**:

```
<!DOCTYPE html>
<html>

<head>
    <title>Mi primera aplicación Vue</title>
    <script src="https://unpkg.com/vue"></script>
</head>

<body>
    <div id="app">
        {{ message }}
    </div>

    <script>
        var app = new Vue({
```

```
            el: '#app',
            data: {
                message: 'Hola Vue.js!'
            }
        })
    </script>
</body>

</html>
```

- En el ejemplo básico hemos cargado Vue.js utilizando la librería remota, aunque también la podríamos tener alojada en nuestro propio servidor.
- Tenemos un div con el id app.
- Instanciamos la aplicación Vue, con **new Vue()** y en esa instanciación creamos un objeto con las propiedades de la aplicación.
- En el momento de la instanciación los datos y el DOM están vinculados, y todo es **reactivo**, es decir cualquier cambio en los datos se verán reflejados en el navegador en cualquier componente dónde se estén utilizando.
    - ♦ Le indicamos el elemento (el) que va a contener la aplicación, en este caso #app.
    - ♦ Creamos una propiedad data que contendrá todas las variables de la aplicación.
- En el div #app, empleamos el **operador mustache {{ }}** para vincular datos en el div. Esta sintaxis básica se puede utilizar para vincular cualquier texto dentro de HTML (title, description, etc...)
- **La sintaxis {{ }} sin embargo no se puede utilizar en los atributos HTML tales como: href, id, src, etc.. Vue para ello proporciona el atributo nativo v-bind (conocido como directiva Vue).**

Fundamentos de Vue.js

Trabajando con plantillas

- En este ejemplo tenemos 2 propiedades nombre y edad.
- Se mostrará solamente el nombre, sin los apellidos.
- En la plantilla evaluamos la propiedad edad y según su valor se muestra Viejo o Joven.

```
<!DOCTYPE html>
<html lang="es">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Trabajanco con plantillas en Vue.js</title>
    <script src="https://unpkg.com/vue"></script>
</head>

<body>
    <div id="app">
        <h1>{{ nombre.split(' ')[0]}}</h1>
        <h1>{{ edad > 60 ? 'Viejo' : 'Joven' }}</h1>

    </div>

    <script>
        var app = new Vue({
            el: '#app',
            data: {
                edad: 47,
                nombre: 'Rafa Veiga'
            }
        })
    </script>
</body>

</html>
```

Métodos

- Los métodos nos permitirán imprimir datos basados en ciertas reglas o lógica.
- Para ello son muy útiles las funciones.
- Lo usaremos en Vue.js con las propiedad **methods** de la instancia de Vue.js.

- La **propiedad methods** es un objeto cuyas **claves** son el **nombre de la función** y el **valor** es el **contenido de la función en si misma**.
- En el ejemplo tenemos el método **obtenerNombreCompleto** que podemos utilizarla directamente en {{ }}.

```
<!DOCTYPE html>
<html lang="es">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Trabajanco con métodos en Vue.js</title>
    <script src="https://unpkg.com/vue"></script>
</head>

<body>
    <div id="app">
        <h1>{{ obtenerNombreCompleto(nombre,apellidos) }}</h1>


    </div>

    <script>
        var app = new Vue({
            el: '#app',
            data: {
                nombre: 'Rafa',
                apellidos: 'Veiga'
            },
            methods: {
                obtenerNombreCompleto: function (primer, segundo) {
                    // Podríamos acceder a las propiedades nombre y apellidos con
                    // this.nombre o this.apellidos
                    // si no quisiéramos pasarlas como parámetros.
                    return primer + ' ' + segundo;

                    // También se podría devolver un objeto con una nuevapropiedad
                    /*
                    return {
                        nuevapropiedad: primer + ' ' + segundo
                    }

                    // Atención no se pueden usar funciones arrow en los métodos,
                    // ya que estas funciones están asociadas al contexto principal y no al contexto de Vue.js
                    */
                }
            }
        });
    </script>
</body>

</html>
```

Introducción a directivas

Expresiones y directivas

Expresiones con directivas

Introducción a eventos

Paso de argumentos a eventos

Modificadores de directivas de eventos

Claves modificadoras

Cola de actualizaciones asíncronas

Entendiendo el DOM virtual

Añadiendo watchers de forma dinámica

Acceso al DOM con refs

Montaje de plantillas dinámicamente

Uso de plantillas inline

Destrucción de una instancia Vue.js

Más información sobre el ciclo de vida y hooks de una instancia Vue.js

Configuración de webpack con Vue.js

Introducción

Instalación de Chrome developer tools

Introducción al CLI de Vue.js

Creación de un proyecto Vue.js desde CLI

Explicación de la estructura del proyecto Vue.js

Componentes en un único fichero

Compilación para producción del proyecto Vue.js

Componentes

Introducción a componentes

Por qué la propiedad data debe ser una función

Componentes globales

Creación de un componente

Organizando componentes

Estilos globales

Transfiriendo datos a los componentes

Validación de datos recibidos

Trabajando con eventos

Comunicación a través de un evento bus

Slots

Slots con nombre

Componentes dinámicos

Cómo mantener componentes dinámicos activos

Ciclo de vida de los componentes dinámicos

Ejemplo de aplicación mail con Vue.js

Filtros mixins

Introducción a los mixins

Utilización de mixins

Cómo se mezclan los mixins

Mixins globales

Filtros globales

Formularios con Vue.js

Input text y textarea

Checkbox

Radio

Select

Modificadores

Cómo funciona la directiva v-model

Añadiendo valores por defecto

Envío de formularios

Animaciones y transiciones con Vue.js

Introducción a animaciones y transiciones

Explicación de transiciones en un único elemento

Transiciones con clases CSS

Implementando nuestra primera transición

Especificando nombres a las transiciones

Especificando clases de transiciones

Implementación de una animación CSS específica

Mezclando transiciones y animaciones

Transiciones entre elementos

Modos de transición

Transición de elementos en la carga inicial de la página

Transiciones con hooks Javascript

Ignorando clases CSS

Transiciones entre componentes dinámicos

Transiciones entre múltiples elementos

Transiciones entre elementos móviles

Enrutamiento en SPA (Single Page Applications)

Introducción a SPA (Single Page Applications)

Instalación de vue-router

Activación del router

Registro de rutas

Renderización de componentes enrutados

Modificación del modo de enrutamiento

Ruta catch-all

Moviendo rutas a un fichero

Añadiendo enlaces de navegación

Estilo del enlace activo en la navegación

Rutas dinámicas matching y linking

Rutas con nombre

Recuperando parámetros en las rutas

Uso de propiedades en las rutas

Reaccionando a cambios en los parámetros

Navegación programática

Navegando en el historial del navegador

Redireccionando

Alias

Rutas anidadas

Parámetros query

Fragmentos hash

Controlando el comportamiento del scroll

Vistas con nombre

Transición de rutas y demás

Conexión a servidores empleando HTTP

Introducción al uso de HTTP en Vue

Ajustes de vue-resource

Obteniendo datos usando peticiones GET

Plantillas URI

Peticiones POST

Uso de recursos

Recursos específicos y acciones

Configuración global

Configuración con componentes

Interceptores

Vuex

Introducción a Vuex

Por qué necesitamos Vuex

Instalación de Vuex

...