

TEXT CLASS REVIEW

TEMAS A TRATAR EN EL CUE:

- Introducción.
- ¿Qué es un SFC? ¿Estructura de un SFC?.
- Reactividad en vue.js.
- One way data binding.
- Two way data binding.
- Directiva v-if.
- Directiva v-for.
- Instalando VUE, y configuración con webpack. Instalación VUE/CLI.

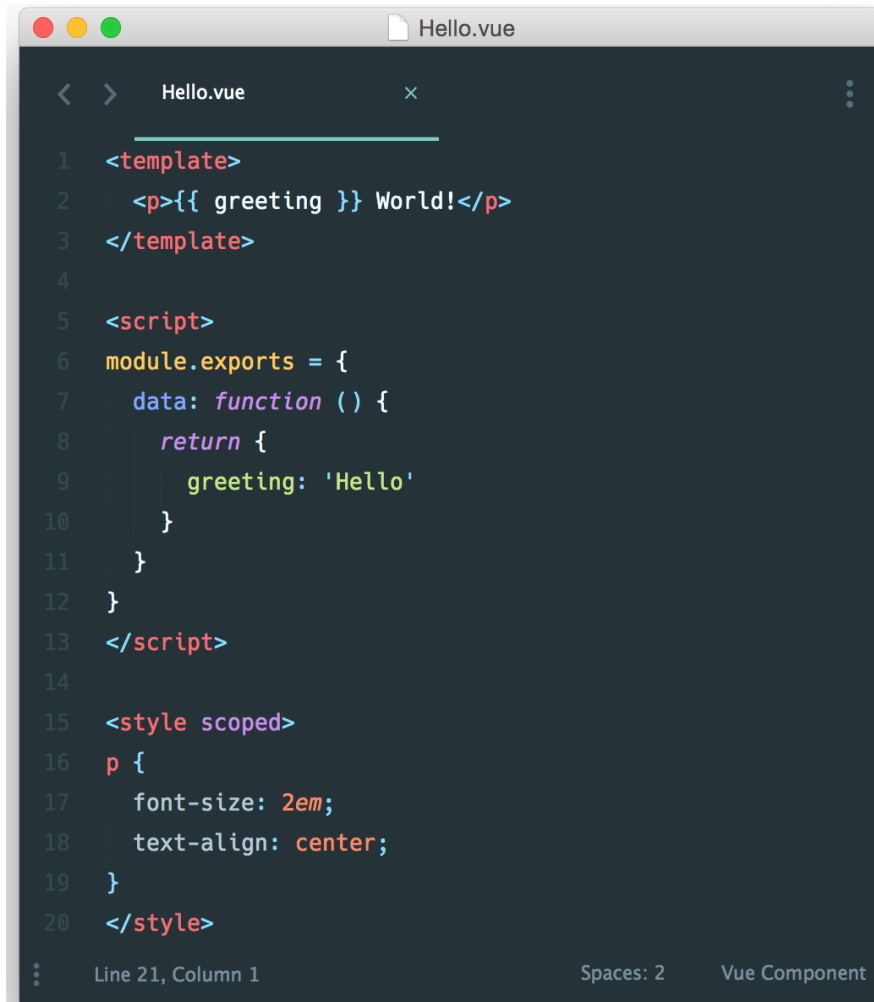
INTRODUCCIÓN:

Como lo revisamos en el CUE1, se puede adjuntar Vue a nuestro proyecto web, solo con exportar su librería a través de CDN (Content Delivery Network). Esta forma funciona excelente cuando necesitamos crear una pequeña pieza de código, que utilice las propiedades de Vue, pero ¿qué pasa cuando nuestro proyecto comienza a crecer?, o cuando notamos que ya se debe ocupar en casi todo nuestro sitio. En este caso, la solución sería instalar Vue y comenzar a utilizar los SFC (Single file Components), para poder tener nuestros componentes ordenados y mantenibles.

¿QUÉ ES UN SFC?

Un Single File Component, es un archivo con extensión .vue, el cual nos permite ir creando componentes de manera ordenada, y con una estructura definida. Lo que busca Vue con los SFC, es poder soportar grandes aplicaciones web, integrando pequeñas partes que luego serán utilizadas en conjunto.

¿ESTRUCTURA DE UN SFC?

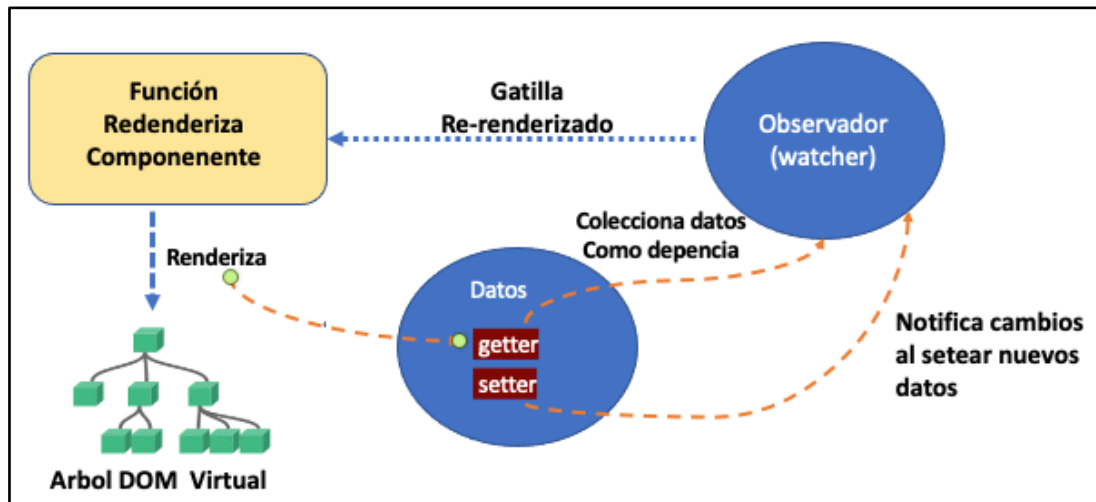


```
1 <template>
2   <p>{{ greeting }} World!</p>
3 </template>
4
5 <script>
6   module.exports = {
7     data: function () {
8       return {
9         greeting: 'Hello'
10      }
11    }
12  }
13 </script>
14
15 <style scoped>
16   p {
17     font-size: 2em;
18     text-align: center;
19   }
20 </style>
```

Line 21, Column 1 Spaces: 2 Vue Component

1. Template: en esta sección va todo el código HTML, el cual puede ser enlazado con datos. También podemos utilizar directivas para generar diferentes comportamientos.
2. Script: encargada de poder manejar los datos y la lógica de nuestros componentes; en esta sección va: el objeto data, methods, entre otros.
3. Style: en esta sección se nos permite crear estilos CSS (o SASS), para darle presentación a nuestro componente.

REACTIVIDAD EN VUE.JS



Cuando declaramos un dato dentro del objeto data, a éste, internamente, se le asigna un getter y un setter, los cuales están enlazados a un observador(watcher¹), que está atento a cualquier cambio que ocurra dentro de él. Si dicho cambio se produce, vue vuelve a renderizar el dato, y es lo que hace que se mantenga reactivo.

ONE WAY DATA BINDING

El enlace de datos en una dirección, se refiere a cuando nosotros tomamos un dato desde el objeto data, y lo renderizamos. En este caso, estamos enlazando en una dirección, desde data hacia template. Para hacer one way data binding, existen dos formas: a través de renderizar datos con *mustache* (`{{}}`), o a través de la directiva `v-bind`.

TWO WAY DATA BINDING

El enlace de datos en 2 direcciones, es utilizado solo por inputs, textarea o select option. En este caso, los cambios pueden ser realizados desde el dato hacia el template, o viceversa.

Este enlace de dos direcciones se hace con la directiva `v-model`.

¹ Watcher: es un patrón de programación, el cual observa los cambios de estados que tenga un sujeto u objeto, para notificar sus suscripciones.

DIRECTIVA V-IF

Esta nos sirve para, en base a una condición, poder renderizar o no, partes de nuestro template. Por ejemplo, si un usuario está logueado, mostrarle cierto contenido específico, y esto es posible hacerlo a través de hacer condiciones dentro de nuestro template.

DIRECTIVA V-FOR

Nos permite generar ciclos iterativos para desplegar datos dentro del template. Imaginemos que tenemos un array con datos, pero no sabes con exactitud el largo de este array, o puede ser variable, por lo que no podemos generar una estructura estática. Lo que se debe hacer, es iterar sobre esos datos, para poder desplegarlos en la vista del usuario.