Vue技術點之間的聯繫

Vue.js技術棧開發前端PC端項目：

* Vue.js語法
* Vue-router
* Axios
* Webpack
* Element-UI
* Vuex

Vue.js技術棧開發前端移動端項目：

* Vue.js語法
* Vue-router
* Axios
* Webpack
* Vant-UI
* Vuex

Vue基礎

## 1 Vue概述

### （1）創始

尤雨溪是Vue.js的創建者

2014年2月，Vue.js正是發佈

2015年10月27日，正是發佈1.0.0

2016年4月27日，發佈2.0的預覽版本

官網：https://cn.vuejs.org/v2/guide/

Vue很高效：20kb運行大小，超快虛擬DOM。

### （2）Vue是漸進式JavaScript框架

#### ① 框架和庫的區別

框架以提供基礎性的服務為主。比如Vue有雙向數據綁定的支撐，可以減少實際開發時的代碼量。

庫以提供API為主。比如jQuery。

現在框架和庫的界限不是很明顯了。所以很多時候把兩者理解成一個東東。

Vue很靈活，在一個庫和一套完整框架之間自如伸縮。

#### ② 漸進式

聲明式渲染→組件系統→客戶端路由→集中式狀態管理→項目構建

## 2 Vue基本使用

  <body>

    <div id="app">

      <!-- {{}} 是插值表達式，作用是把數據填充到HTML標籤裡 -->

      <div>{{msg}}</div>

      <!-- 差值表達式支持基本的計算操作。比如下面的標籤裡顯示的數字是3。-->

      <div>{{1+2}}</div>

      <!-- 只要符合JS語法規則的表達式都可以放在這裡。 -->

      <div>{{msg + '------' + 123}}</div>

    </div>

    <script src="https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js"></script>

    <script type="text/javascript">

      new Vue({

        el: "#app", //元素的掛載位置，值可以是CSS選擇器或者DOM元素。掛載是指把數據關聯到頁面的某個標籤裡。

        // data是模型數據，即開發中遇到的業務相關數據，值是一個對象。

        data: {

          msg: "Hello World by Vue",

        },

      });

    </script>

  </body>

Vue不需要直接操作DOM了。

Vue代碼運行原理分析

Vue.js把Vue代碼編譯為原生JS代碼，以便瀏覽器解析。

## 3 Vue模板語法

### （1）前端渲染

Vue是用來做前端渲染的。

前端渲染即把從數據庫請求回來的數據填充到HTML標籤中。

前端渲染的產物是靜態HTML內容，這內容最終被瀏覽器呈現給用戶。

### （2）前端渲染的方式

#### ① 原生js拼接字符串

缺點：不同開發人員的代碼風格差別很大；隨著業務越來越複雜，後期的維護變得逐漸困難起來。

#### ② 使用前端模板引擎，比如art-template

優點：大家都遵循同樣的規則寫代碼，代碼可讀性明顯提高，方便後期維護。

缺點：沒有專門提供事件機制。

#### ③ 使用vue特有的模板語法

模板語法概覽：

* 插值表達式
* 指令
* 事件綁定
* 屬性綁定
* 樣式綁定
* 分支循環結構

### （3）指令

#### ① 什麼是指令？

指令的本質就是自定義屬性。

比如下面的abc=123就是一個自定義屬性。標準屬性例子：a標籤中的href屬性。

<div id="app" abc=123>

指令的格式：以v-開始，後面跟個字符串，比如v-cloak。

#### ② v-cloak指令用法

* 差值表達式存在的問題：閃動（如果網絡不好，頁面會先顯示花括號，再把花括號替換為相關的數據內容，這個過程就叫做閃動，給用戶造成不好的體驗。）
* 如何解決該問題？使用v-cloak指令

    <style type="text/css">

      [v-clock] {

        display: none;

      }

    </style>

      <div v-cloak>{{msg}}</div>

* 解決該問題的原理：先通過樣式隱藏內容，然後在內存中進行值的替換，替換好值之後再顯示最終的結果。

#### ③ 數據綁定指令

1. v-text 填充純文本
   * 沒有閃動問題
   * 比插值表達式更加簡潔
   * 推薦用v-text而非插值表達式
2. v-html 填充HTML片段
   * 存在安全問題
   * 本網站內部數據可以使用，來自第三方的數據不可以用
3. v-pre 填充原始信息
   * 顯示原始信息，跳過編譯過程（分析編譯過程）

範例代碼：

  <body>

    <div id="app">

      <div v-text="msg"></div>

      <div v-html="msg1"></div>

      <div v-pre>{{msg}}</div>

    </div>

    <script src="https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js"></script>

    <script type="text/javascript">

      new Vue({

        el: "#app",

        data: {

          msg: "Hello World by Vue",

          msg1: "<h1>HTML標籤h1</h1>",

        },

      });

    </script>

  </body>

顯示的樣子：



## 基礎案例

Vue常用特性

綜合案例