Vue.js 基础入门

张镇球

课程安排

1. 初步了解 Vue.js 框架
2. 介绍 Vue.js 开发环境的搭建和脚手架工具的使用
3. Vue.js 具体的指令和项目实践

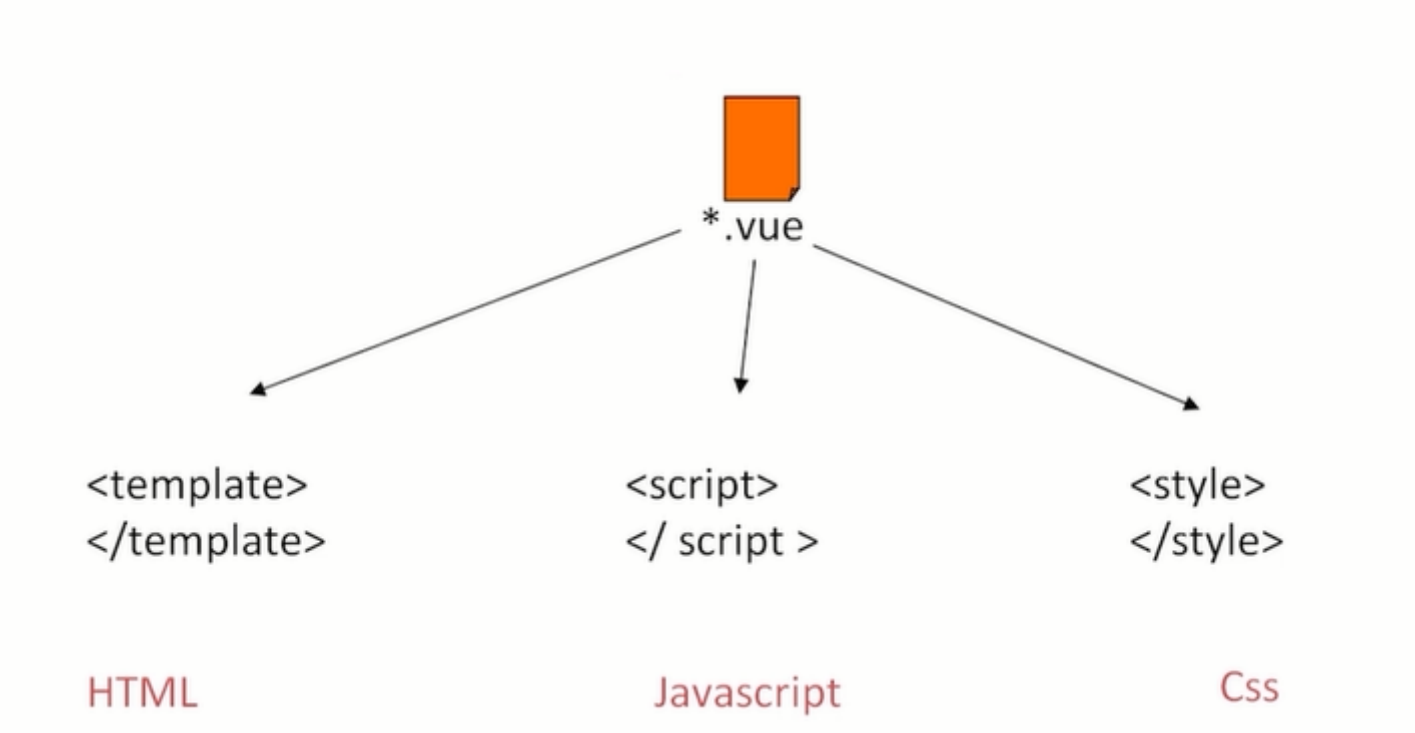
认识 Vue.js

- Vue.js 作者是尤雨溪
- Vue.js 是轻量级的MVVM框架，它吸收了 React 和 Angular 的优点，它强调 React 组件化的概念（可以轻松实现数据的展现和分离），引入了 Angular 的灵活指令和页面操作的方法。
- Vue.js 的目标是通过尽可能简单的 API 实现响应的数据绑定和组合的视图组件。

准备知识

1. 前端开发基础 HTML, CSS, JS
2. 前端 模块化 基础
3. 对 ES6 有初步的了解

Vue.js 每个页面都是一个组件



环境搭建和脚手架

Node.js 版本要求4.x以上

```
# 最新稳定版
$ npm install vue

# 全局安装 vue-cli
$ npm install --global vue-cli
# 创建一个基于 webpack 模板的新项目
$ vue init webpack my-project
# 安装依赖，走你
$ cd my-project
$ npm install
$ npm run dev
```

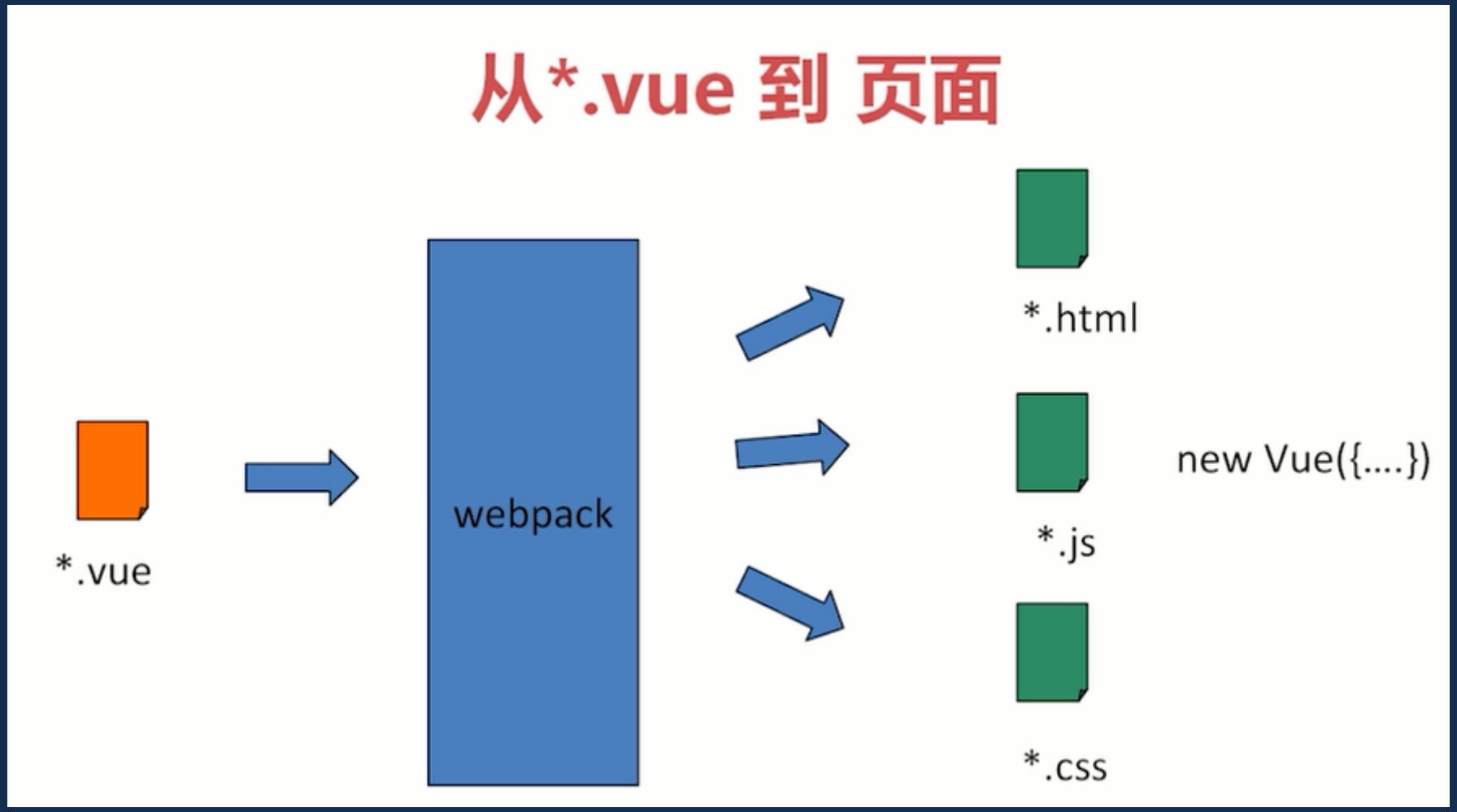
环境搭建和脚手架

```
F:\Day 2-Vue.js>vue init webpack my-project

  This will install Vue 2.x version of the template.

  For Vue 1.x use: vue init webpack#1.0 my-project

? Project name my-project
? Project description this is my first vue project
? Author zhangzq
? Vue build standalone
? Install vue-router? No
? Use ESLint to lint your code? Yes
? Pick an ESLint preset Standard
? Setup unit tests with Karma + Mocha? No
? Setup e2e tests with Nightwatch? No
```



演示简单的例子！

Vue 组件的重要选项

- 数据对象 data
- 方法对象 methods
- 监听对象 watch

Vue 实例的数据对象 data

- Vue的所有数据都是放在data对象里面的；
- 对象必须是纯粹的对象(含有零个或多个的key/value对)
- 理论上data 只能是数据 - 不推荐观察拥有状态行为的对象。

Vue 方法对象 methods

- Vue 的所有方法(函数)都是放在methods对象里面
- 注意：不应该使用箭头函数来定义 method 函数 (例如 plus: () => this.a++)。理由是箭头函数绑定了父级作用域的上下文，所以 this 将不会按照期望指向 Vue 实例，this.a 将是 undefined。

Vue 监听对象 watch

- Vue 的数据监听的方法 watch，就是当数据改变的时候我需要干什么？
- 是一个对象，键是需要观察的表达式，值是对应回调函数。值也可以是方法名，或者包含选项的对象。

Vue 模板指令

模板指令是 HTML 和 Vue 对象的粘合剂，模板指令是写在 HTML 里的

```
- 数据渲染：v-text ，v-html，{{}}
- 判断模块显示隐藏：v-if，  v-show
- 渲染循环列表：v-for
- 事件绑定：v-on
- 属性绑定：v-bind
- 双向绑定：v-model
```

Vue 数据渲染

数据渲染包括三个 v-text，v-html，{{}}

1. v-text 是更新元素的内容部分；
2. v-html 可以更新元素的innerHTML，保留的html结构
3. {{}} 是类似v-text对HTML结构做了处理，形成文本字符串输出；

Vue 判断模块显示隐藏

判断模块显示隐藏：v-if, v-show

- v-if,v-else, v-else-if 不渲染 DOM，根据表达式值 的真假条件渲染元素
- v-show 有渲染 DOM，根据表达式之真假值，切换元素的 display CSS 属性

Vue 渲染循环列表：v-for

- 我们用 v-for 指令根据一组数组的选项列表进行渲染。
- v-for 指令需要以 item in items 形式的特殊语法， items 是源数据数组并且 item 是数组元素迭代的别名。

Vue 事件绑定：v-on

指定的方法写在methods对象里，v-on内容比较多，大家可以在官网上查看

```
<!-- 方法处理器 -->
<button v-on:click="doSomething"></button>

<!-- 缩写 -->
<button @click="doSomething"></button>
```

```
methods: {
    doSomething: function () {
      this.a ++
    }
  }
```

Vue 属性绑定：v-bind

注意有布尔值和字符串两种用法，这里的值都是在data里面取的；

```
<!-- 绑定一个属性 -->
<img v-bind:src="imageSrc">
<!-- 缩写 -->
<img :src="imageSrc">

<!-- class 绑定 -->
<div :class="{ red: isRed }"></div>
<div :class="[classA, classB]"></div>
<div :class="[classA, { classB: isB, classC: isC }]">
```

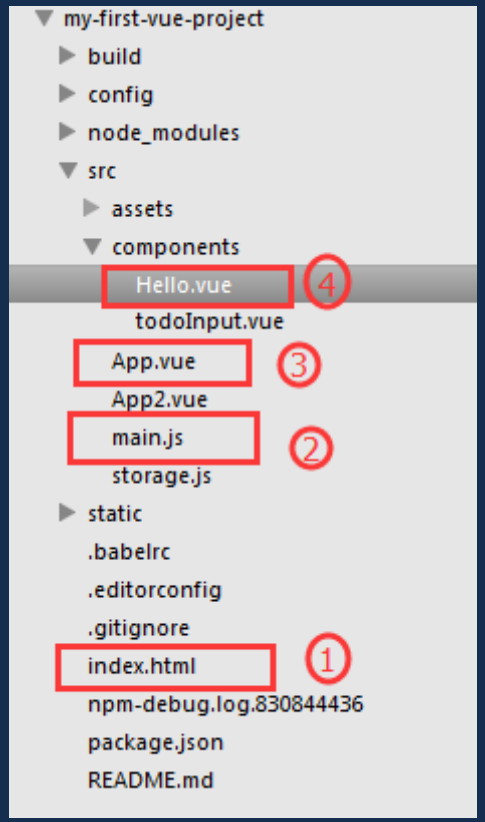
Vue 双向绑定：v-model

可以通过v-model在表单控件或者组件上创建双向绑定，可以实现随控件变化而变化。通常用在

```
<input>, <select>, <textarea>
```

```
<input v-model="msg" />
<h2>{{msg}}</h2>
```

实战，制作一个todolist



工程结构说明

实现组件间调用 components

组件（Component）是 Vue.js 最强大的功能之一。

- 组件可以扩展 HTML 元素，封装可重用的代码
- 在较高层面上，组件是自定义元素， Vue.js 的编译器为它添加特殊功能

```
HTML部分的使用，自定义新增的标签
<hello></hello>

应入组件
import Hello from './components/Hello'

export default {
  name: 'app',
  components: {  // 注册
    Hello
  }
}
```

Vue.js 组件间传值

1. 父亲组件给子组件传值props
2. 子组件给父组件传值'自定义事件v-on'

父亲组件给子组件传值props

- props 可以是数组或对象，用于接收来自父组件的数据；选项是注册在子组件中，然后父亲组件的属性作为Key
- props 是父组件用来传递数据的一个自定义属性。

```
属性写在父亲组件
<hello1 msgA="这个是父亲组件传过来的内容"></hello1>

props 选项注册在子组件下
props: ['msgA']
```

子组件给父组件传值'自定义事件v-on'

```
父组件绑定一个监听子组件消息的自定义事件“child-tell-me”
<hello2 v-on:child-tell-me="saySomething"></hello2>

父亲组件的事件处理，将子组件的内容传递回来
methods: {
    saySomething: function (msg) {
      this.childTellMe = msg
    }
  }
```

```
子组件触发，向父亲组件传值
<button v-on:click="mySay">子组件说话</button>

通过 emit 触发，将子组件的数据传给父亲组件
methods: {
    mySay: function () {
      this.$emit('child-tell-me', this.msg)
    }
  }
```

状态管理 store

所有 store 中 state 的改变，都放置在 store 自身的 action 中去管理。这种集中式状态管理能够被更容易地理解哪种类型的 mutation 将会发生，以及它们是如何被触发。

姚军来补充

作业

请大家完成一个 todolist

谢谢！