**初识Vue（一）**

写在前面的话：

文章是个人学习过程中的总结，为方便以后回头在学习。

文章中会参考官方文档和其他的一些文章，示例均为亲自编写和实践，若有写的不对的地方欢迎大家指出。

作者简介：

一个不知名的前端开发，正在为能走向更高更远的地方而努力。

**一.前言**

本篇作为vue入门的一篇总结，目地在使用vue一些简单的特性去打开vue的学习之门。

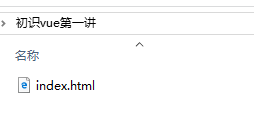
**二.如何去搭建vue开发环境**

搭建vue的开发环境一般有两种方式，一种是使用vue-cli脚手架，这个需要具备Node的一些基础知识；第二种方式是在我们编写的html页面中引入vue.js文件，就可以使用vue了。后面我们一直会使用这样的方式去介绍vue中的一些知识点和特性。

注意：后面引入的vue文件都是基于vue.2x版本的，大多数的知识点和特性的介绍都是参考官方文档。官方文档地址：[https://vuejs.bootcss.com/v2/guide/](https://vuejs.bootcss.com/v2/guide/" \t "_blank)

**三.开始使用vue**

**1.创建项目目录**



**2.在html中引入vue.js**

**在index.html引入vue.js**

在html中直接引入vue.js也有两种方式，一种是直接将文件下载到本地，一种是引用cdn上面的文件。

此处我们就直接按照官方文档使用cnd的方式引入vue.js



<!-- index.html -->

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>初识vue入门第一课</title>

<!-- 按照官方文档，引入开发环境的vue.js，准确的版本为 v2.6.10 -->

<script type="text/javascript" src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>

</head>

<body>

</body>

</html>

**2.创建第一个vue实例**

创建vue实例需要通过构造函数创建的。

var vm = new Vue({

});

使用构造函数创建vue实例时，我们可以传入一个包含多个键值对的字典对象，完整的键值对配置可以参考[官方文档](https://vuejs.bootcss.com/v2/api/" \l "%E9%80%89%E9%A1%B9-%E6%95%B0%E6%8D%AE" \t "_blank)。后面的部分我们只简介最基础的配置项。

**3.DOM-el配置项**

**el配置项的数据类型**

el配置项的值可以是两种类型，一种是字符串，一种是HTMLElement。当是字符串时，它表示一个css选择器；当是HTMLElement时，它表示文档中的某个元素。

**基本语法**

var vm = new Vue({

el:'id选择器' | HTMLElement,

});

或者

var vm = new Vue({

});

vm.$el = 'id选择器' | HTMLElement ;

**作用**

它主要的作用是将vue实例挂载到文档中的DOM节点上，挂载完成后的DOM节点就是一个vue容器，在这个容器里我们就可以使用vue的一些特性；vue实例创建之后，我们可以使用vm.$el去访问挂载的这个DOM节点。

**示例**

下面我们写一个示例体验一下。

<!-- index.html -->

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>初识vue入门第一课</title>

<!-- 按照官方文档，引入开发环境的vue.js，准确的版本为 v2.6.10 -->

<script type="text/javascript" src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>

</head>

<body>

<div id="box">

</div>

<script type="text/javascript">

var vm = new Vue({

el: '#box'

});

//在构造函数外部访问vue挂载点dom节点

var vueEle = vm.$el;

console.log(vueEle);

</script>

</body>

</html>

打印结果：

image.png

**4.数据-data配置项**

**data配置项的数据类型**

data的数据类型同样是两种：javascript原生的Object类型或者一个返回原生对象的函数。函数类型一般用于在定义组件的时候，因为组件是会被复用的，这样每一个复用组件的实例都会拥有这个data数据的副本，而不会互相影响。

**基本语法**

var vm = new Vue({

data: {

属性名: 属性值

},

});

或者

var vm = new Vue({

});

vm.$data.属性名 = 属性值;

**作用**

它主要是给vue示例提供数据支撑。同时vue实例创建之后，可以使用vm.$data访问这个数据。

**示例**

<!-- index.html -->

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>初识vue入门第一课</title>

<!-- 按照官方文档，引入开发环境的vue.js，准确的版本为 v2.6.10 -->

<script type="text/javascript" src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>

</head>

<body>

<div id="box">

</div>

<script type="text/javascript">

var vm = new Vue({

el: '#box',

data:{

name: 'todo',

age: 20,

}

});

var name = vm.$data.name;

console.log(name); // 'todo'

var age = vm.$data.age;

console.log(age); // 20

</script>

</body>

</html>

**5.vue模板语法渲染数据-文本插值**

上面我们介绍了如何在vue实例中定义数据，那么如何将vue实例中的数据展示到我们的页面中呢，这就需要使用vue的模板渲染语法，[模板渲染语法](https://cn.vuejs.org/v2/guide/syntax.html" \t "_blank)包含很多种形式，这里我们只讲双花括号形式的文本插值。

双花括号的文本插值法，就是将vue中的数据以文本的形式解析到html中。在html中直接访问vue实例里data对象中的属性即可获取到对应的值。而且当data对象中的属性值发生变化，模板中插入的值也会自动发生变化。

**基本语法**

{{ 变量名 }}

**示例**

<!-- index.html -->

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>初识vue入门第一课</title>

<!-- 按照官方文档，引入开发环境的vue.js，准确的版本为 v2.6.10 -->

<script type="text/javascript" src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>

</head>

<body>

<div id="box">

<!-- 使用双花括号将vue中的数据以文本的形式展示 -->

<h1>{{name}}</h1>

<h1>{{age}}</h1>

</div>

<script type="text/javascript">

var vm = new Vue({

el: '#box',

data:{

name: 'todo',

age: 20,

}

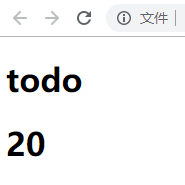
});

</script>

</body>

</html>

界面展示结果：



**四.总结回顾**

至此我们vue入门第一课内容基本已经结束，在此总结一下前面讲过的内容

* 使用构造函数的方式可以创建一个vue示例
* el属性将vue挂载到DOM节点上
* data属性给vue示例提供数据
* 双花括号可以访问实例中data对象的属性

**要加油鸭**