**vue中的methods属性**

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：

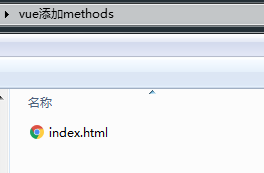
一个不知名的前端开发，正在为能走向更高更远的地方而努力。

**一.前言**

上一篇《初始vue第一篇》我们创建了一个vue实例，并将它挂载到了div#box这个DOM节点上，同时在html使用双花括号插值法引用显示了data中的数据。这篇主要是学习总结给vue中添加方法。

**二.给vue添加methods属性**

**创建项目目录**



**基本语法**

var vm = new Vue({

methods:{

//在此处定义方法

方法名：function(){

}

}

});

或者

var vm = new Vue({

});

//在构造函数外部定义方法，不需要使用vm.$methods，直接使用vm.方法名 = function(){}

vm.方法名 = function(){

}

**示例**

       在这个示例之前需要补充一个知识点：**如在methods方法中访问data的数据** 。

前一篇文章我们介绍了在Vue构造函数外部使用vm.$data.属性名去访问data中的数据。而在methods方法中，我们可以直接使用this.属性名去访问data中的数据，其中this表示的就是vue实例对象。

<!-- index.html -->

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>给vue添加methods属性</title>

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

</head>

<body>

<div id="box">

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

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

</div>

<script type="text/javascript">

var vm = new Vue({

el: '#box',

data:{

name: 'todo',

age: 20,

},

methods:{

add: function(){

//在methods内部访问data中的数据：this.属性名

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

return this;

}

}

});

// 手动调用add方法

var obj = vm.add();

//返回的obj等于vm实例

console.log(obj == vm); //true

</script>

</body>

</html>

**三.总结**

1.使用**methods**属性给vue定义方法

2.在方法中使用**this.属性名**就可以直接访问data中的数据

3.在构造函数外部可以使用**vm.方法名**定义或者调用方法

**要加油鸭**