# 模板引擎

//art-template模板引擎

//安装

npm install art-template

//安装目录  node\_modules中

//template-web.js浏览器使用

在浏览器中使用模板引擎要引入template-web.js文件

 <!-- 引入文件 -->

    <script src="node\_modules/art-template/lib/template-web.js"></script>

    <script type="text/template" id="tpl">

        hello {{name}}

        爱好{{each hobbies}} {{$value}} {{/each}}

    </script>

    <script>

        //template('script中的id',{对象})

        var ret=template('tpl',{

            name:'chen',

            hobbies:[

                '唱歌',

                '跳舞',

                '敲代码'

            ]

        })

        console.log(ret);

    </script>

输出：hello chen 爱好 唱歌 跳舞 敲代码

# 在Node中使用模板引擎

//加载引擎

//art-template是下载的包名

var template=require('art-template');

var fs=require('fs');

fs.readFile('./08tpl.html','utf8',function(err,data){

    if(err){

        return console.log('读取文件失败');

    }

    // template.render('模板字符串',{替换对象})

    var ret=template.render(data,{

        name:'chen',

        hubbies:[

            '唱歌',

            '敲代码'

        ]

    })

    console.log(ret);//模板引擎资只会解析{{}},其他的都是字符串

})

08tpl.html

<body>

    hello {{name}}

    爱好：{{each hubbies}} {{$value}} {{/each}}

</body>

# each和foreach的区别

## 在模板引擎中each （只可以在模板字符串中使用）：

{{each 数组}}

<li>{{$value}}</li>

{{/each}}

## 在JS中 each和foreach都可以（由于foreach是EcmaScript 5中的，所有低版本的浏览器不支持）：

* 在j Query遍历中 each必须加载jquery ：

$.each( 数组 ，function)

$(‘div’).each(function) 一般用于遍历jQuery选择器选择到的伪 数 组实例对象

each在jQuery的原型链中

* foreach循环是EcmaScript中的一个数组遍历函数，是JavaScript原生支持的遍历方法 可以遍历任何被遍历的成员(数组)，不需要加载

# 总结：

1. jQuery的each不是专门用来遍历jQuery元素的
2. 方便的遍历jQuery元素
3. 可以在不兼容foreach的低版本浏览器中使用jQuery的each方法

伪数组转换为数组，就可以使用foreach在不兼容的浏览器中

[].slice.call($('div')) //$('div')实例对象

### 伪数组转换为数组

Var a ={

0:’df’,

1:’cv’,

2:’gh’,

Length:3

} //这是对象，不是数组，是一个伪数组

[].slice.call(a); //转换为数组