artTemplate

```
性能卓越，执行速度快
支持运行时调试，可精确定位异常模板所在语句错误位置
支持 Express
支持 include 语句，可导入定义的其它模块
模板语法简洁
兼容所有浏览器
```

查看官网 (https://github.com/aui/artTemplate)

```
https://github.com/aui/artTemplate
```

下载 (https://raw.githubusercontent.com/aui/artTemplate/master/dist/template.js)

```
https://raw.githubusercontent.com/aui/artTemplate/master/dist/template.js
```

Basis

```
<script id="test" type="text/html">
        <h1>{{ title }}</h1>
        <ul>
        {{ each list as value i }}
                <li>索引 {{ i + 1 }} ：{{ value }}</li>
        {{ /each }}
        </ul>
</script>
```

```
var data = {
    title: '标签',
    list: ['文艺', '博客', '摄影', '电影', '民谣', '旅行', '吉他']
};
var html = template('test', data);
// 将执行后的 html 添加到 dom 中
```

方法

```
template(id, data);

template 通过 id 去获取模版代码，然后与 data 数据结合，返回编译后的 html 代码
```

template.compile

用于获取一个模版对象

```
var tpl = template.compile('test');
```
等同于
```
var tpl = template('test');
```

假如一个数据结构更新很频繁，当数据更新很频繁时就可以事先将模版准备好，当拿到数据时直接做编译

```
var tpl = template.compile('test');

var html = tpl({
        title: '标签',
        list: ['文艺', '博客', '摄影', '电影', '民谣', '旅行', '吉他']
});


var html = tpl({
        title: '标签',
        list: ['java', 'c#', 'javascript', 'html', 'css']
});


var html = tpl({
        title: '标签',
        list: ['文艺', '博客', '摄影', '电影', 'c++', 'php', '.net']
});

...
```

template.helper

```
辅助方法，可以扩展常用的公共方法

语法 : {{ data | funname : '第二参数' }}

具体调用为 : funname(data,'第二参数');
```

添加一个时间格式化辅助方法

```
template.helper('dateFormat', function (date, format) {
    date || (date = new Date(date));
    format || (format = 'yyyy-MM-dd hh:mm:ss');
    var map = {
        "M": date.getMonth() + 1, //月
        "d": date.getDate(), //日
        "h": date.getHours(), //小时
        "m": date.getMinutes(), //分
        "s": date.getSeconds(), //秒
        "q": Math.floor((date.getMonth() + 3) / 3), //季度
        "S": date.getMilliseconds() //毫秒
    };
    format = format.replace(/([yMdhmsqS])+/g, function(all, t){
        var v = map[t];
        if(v !== undefined){
            if(all.length > 1){
                v = '0' + v;
                v = v.substr(v.length-2);
            }
            return v;
        }
        else if(t === 'y'){
            return (date.getFullYear() + '').substr(4 - all.length);
        }
        return all;
    });
    return format;
});
```

定义模版

```
<script id="test" type="text/html">
        {{time | dateFormat:'yyyy-MM-dd hh:mm:ss'}}
</script>
```

调用模版

```
var tpl = template.compile('test');
var html = tpl({
        time : new Date()
});
```

不转义 html

当参数中有 html 字符串时，会默认转义 html，如果希望不转义时需要单独处理

定义模版

```
<script id="test" type="text/html">
    <p>不转义：{{#value}}</p>
    <p>默认转义：  {{value}}</p>
</script>
```

调用模版

```
var data = {
        value: '<span style="color:#F00">hello world!</span>'
};
var html = template('test', data);

// html 输出如下
//<p>不转义：<span style="color:#F00">hello world!</span></p>
//<p>默认转义：  &#60;span style=&#34;color:#F00&#34;&#62;hello world!&#60;/span&#62;</p>
```

include

导入其它模版，很实用的一个功能

定义模版

```
<script id="test" type="text/html">
        <h1>
                {{ title }}
        </h1>
        {{ include 'list' }}
</script>
<script id="list" type="text/html">
        <ul>
            {{ each list as value i }}
                <li>索引 {{ i + 1 }} ：{{ value }}</li>
            {{ /each }}
        </ul>
</script>
```

调用模版

```
var data = {
        title: '嵌入子模板',
        list: ['文艺', '博客', '摄影', '电影', '民谣', '旅行', '吉他']
};
var html = template('test', data);
```

template.config

具体配置模版编译参数

字段	类型	默认值	说明
openTag	String	{{	逻辑语法开始标签
closeTag	String	}}	逻辑语法结束标签

escape	Boolean	true	是否编码输出 HTML 字符
escape	Boolean	true	是否开启缓存
escape	Boolean	false	是否压缩 HTML 多余空白字符

语法

表达式

使用 {{ 与 }} 符号包括起来的语句为表达式，一句简单的 js 逻辑代码，可以做简单的预算

```
{{ content }}

{{ 1 + 1 }}

{{ content ? 1 : 2 }}
```

条件逻辑

if 、else 、else if

```
{{if admin}}
    <p> {{ admin }} </p>
{{else if code > 0}}
    <p>{{ code }}</p>
{{else}}
    <p> auto </p>
{{/if}}
```

遍循环

遍历数组或则对象
语法 ： `data as item key`

```
{{each list as value index}}
    {{ index + 1}} - {{value.user}}
{{/each}}
```