



真传X

IT前沿技术在线大学

www.zhenchuanx.com

模板引擎和模块化

模板引擎

```
<div id="container">  
| 我们正处于刀耕火种的石器时代  
</div>
```

DOM API

```
var newTxt = '石器时代需要自己撸工具，摩擦摩擦，似魔鬼的步伐...';  
var container = document.getElementById('container');  
var title = document.createElement('H1');  
var txt = document.createTextNode(newTxt);  
title.appendChild(txt);  
container.replaceChild(desc, container.childNodes[0]);
```

innerHTML

```
var newTxt = '石器时代需要自己撸工具，摩擦摩擦，似魔鬼的步伐...';  
var template = '<H1>' + newTxt + '</H1>';  
var container = document.getElementById('container');  
container.innerHTML=template;
```

有什么问题？

- Dom api 操作繁琐
- innerHTML 可读性和可维护性查
- innerHTML易产生XSS 安全漏洞

怎么解决

模板类库

JavaScript Micro-Templating

```

// Simple JavaScript Templating
// John Resig - https://johnresig.com/ - MIT Licensed
(function(){
    var cache = {};

    this.tmpl = function tmpl(str, data){
        // Figure out if we're getting a template, or if we need to
        // load the template - and be sure to cache the result.
        var fn = !/\W/.test(str) ?
            cache[str] = cache[str] ||
                tmpl(document.getElementById(str).innerHTML) :

            // Generate a reusable function that will serve as a template
            // generator (and which will be cached).
            // 等价于var fn = function(obj) {}
            new Function("obj",
                "var p=[],print=function(){p.push.apply(p,arguments);};" +

                // Introduce the data as local variables using with(){}
                "with(obj){p.push('" +

                // Convert the template into pure JavaScript
                str
                    .replace(/\r\t\n/g, " ")
                    .split("<%").join("\t")
                    .replace(/((^|>)[^\\t]*)'/g, "$1\r")
                    .replace(/\t=(.*?)>/g, "', $1, '")
                    .split("\t").join("'");"
                    .split(">").join("p.push('")
                    .split("\r").join("\\'")
                + "');}return p.join('');");

        // Provide some basic currying to the user
        return data ? fn( data ) : fn;
    };
})();

```

```

<script type="text/html" id="item_tmpl">
  <div id="<%=id%>" class="<%= (i % 2 == 1 ? " even" : "") %>">
    <div class="grid_1 alpha right">
      
    </div>
    <div class="grid_6 omega contents">
      <p><b><a href="/<%=from_user%>"><%=from_user%></a>:</b> <%=text%></p>
    </div>
  </div>
</script>
<script>
  var dataObject = {
    id: 1,
    i: 1,
    profile_image_url: 'https://www.google.com.hk/images/branding/googlelogo/2x/g
    from_user: 'Bob',
    text: 'hello world'
  }
  var results = document.getElementById("container");
  results.innerHTML = tmpl("item_tmpl", dataObject);
</script>

```



这些模板引擎又可以细分为2类:

功能 VS 性能

功能：handlebars强大的Blocks

```
<div class="entry">
  <h1>{{title}}</h1>
  <div class="body">
    {{#bold}}{{body}}{{/bold}}
  </div>
</div>
```

```
Handlebars.registerHelper('bold', function(options) {
  return new Handlebars.SafeString(
    '<div class="mybold">'
    + options.fn(this)
    + '</div>');
});
```



<http://handlebarsjs.com/>

如何做到性能好？

1. 体积小
2. 支持线下预编译，成为静态JS文件

性能：体积小

<https://github.com/olado/doT/blob/master/doT.min.js>

<https://github.com/jashkenas/underscore/blob/master/underscore.js#L1525>

性能：预编译

```
var render = (function () {
  var cache =
    "var $out = [];\n
    with ($data) {\n
      $out.push('<h3>');\n
      if (typeof content === 'string') {\n
        $out.push(content);\n
      }\n
      $out.push('</h3>');\n
    }\n
    return $out.join('');"

  return function (data) {
    var fn = new Function('$data', cache);
    return fn(data);
  }
})();
```

```
var render = function ($data) {
  var content = $data.content, $out = '';
  $out += '<h3>';

  if (typeof content === 'string') {
    $out += content;
  }

  $out += '</h3>';
  return $out;
};
```

性能：预编译

1. `str.push('test') ; str.join("");`
2. `str+=""; str+="";`

哪种快？

性能：预编译

数组 `push` 方法拼接字符串会比 `+=` 快，要知道这仅仅是 IE6-8 的浏览器下。

实测表明现代浏览器使用 `+=` 会比数组 `push` 方法快，而在 v8 引擎中，使用 `+=` 方式比数组拼接快 4.7 倍。

更加强大的模板

--- MVVM和react.js

MVVM: DOM template

```
<div id="app-3">
  <p v-if="seen">现在你看到我了</p>
  <ol>
    <li v-for="todo in todos">
      {{ todo.text }}
    </li>
  </ol>
</div>
```

配合双向绑定，DOM template具有强大灵活的表征能力

React.js: JSX

```
export default class Online extends Component {
  constructor() {
    super();
    Report.init(this);
  }
  wording() { ...
  }

  downWording() { ...
  }

  openApp() { ...
  }

  render() {
    return (
      <section id="bar-container">
        <div className="download">
          <i className="icon-logo"></i>
          { this.props.appStatus !== -1 && this.props.anchorName &&
            <span>{this.wording()}</span>
          }
          { this.props.appStatus !== -1 && this.props.anchorName &&
            <div className="download-btn" onClick={this.openApp.bind(this)}>
              {this.downWording()}
            </div>
          }
        </div>
      </section>
    )
  }
}
```


React.js: JSX

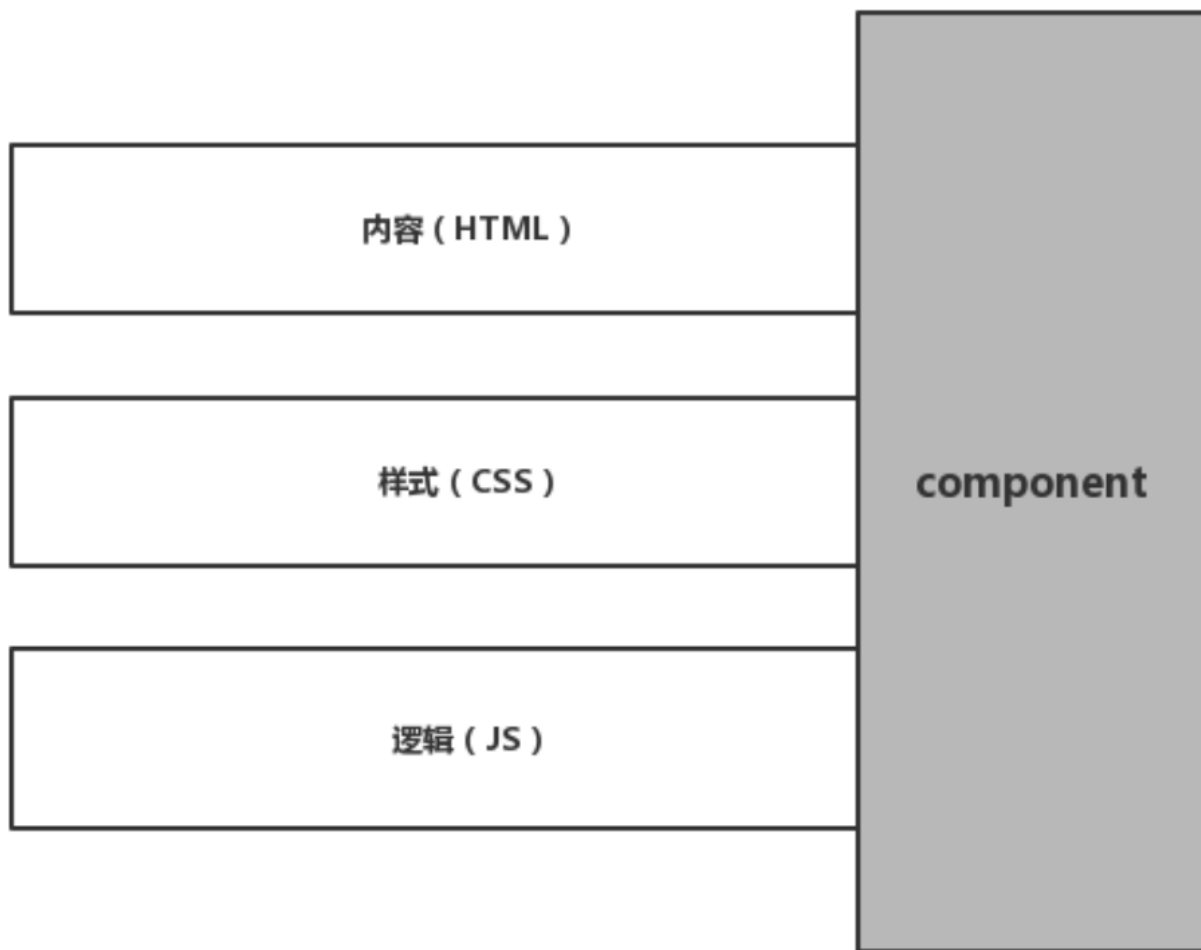
- 1.all in JS（逻辑+内容+样式（可选））
- 2.利于组件化（<https://ant.design/components/button-cn/>）

组件化（web compoent）

内容 (HTML)

样式 (CSS)

逻辑 (JS)



```
<hello-element name="Bob"></hello-element>
<script>
  class HelloElement extends HTMLElement {
    static get observedAttributes() {
      return ['name'];
    }

    attributeChangedCallback(attribute, oldValue, newValue) {
      if(attribute === 'name') {
        this.textContent = `Hello ${newValue}`
      }
    }
  }

  customElements.define('hello-element', HelloElement);
</script>
```

模块化

```
// lib.js  
  
var test=1234;  
  
// main.js  
console.log(test)
```

```
//lib.js  
(function (){  
    var test 1234;  
})
```

```
//main.js => undefined , 无法引用  
console.log(test)
```


Commonjs - nodejs 规范

amd - requirejs 规范

cmd - seajs 规范

Commonjs - nodejs 规范

node

amd - requirejs 规范

browser

cmd - seajs 规范

```
amd
    require(["a.js", "b.js"] , function (){
    })
```

```
cmd
    require(function (){
        var a = require("a.js");
        var b= require("b.js")
    })
```

异步加载，性能差！

---- 线下打包

- require.js

r.js

grunt-contrib-requirejs

- seajs

spm

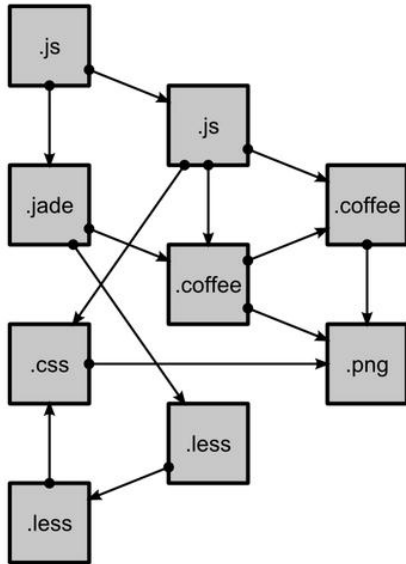
(seajs 模块系统)

grunt-cmd-transport

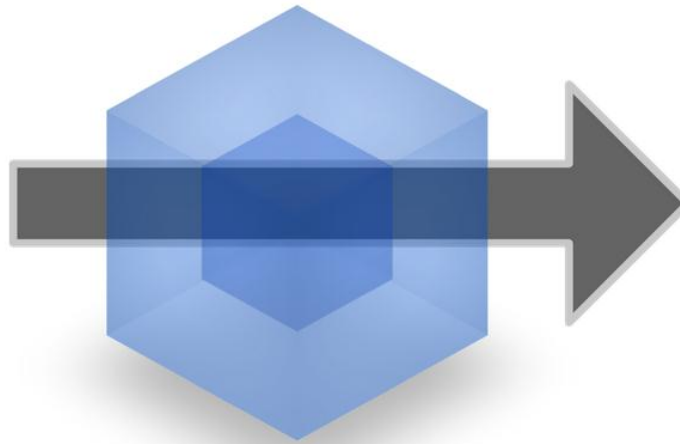
(提取依赖)

grunt-cmd-concat

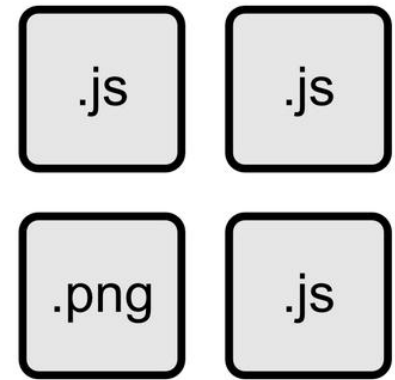
(合并)



modules
with dependencies



webpack
MODULE BUNDLER



static
assets

实战

作业

写一个简单的模板类库
写一个webpack server 模式
写一个 webpack 插件
用webpack 改造自己项目



IT前沿技术在线大学